**NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS**:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

# A Connectionist Approach to Word Sense Disambiguation

Garrison Weeks Cottrell
Computer Science Department
University of Rochester

May 1985

TR 154

CARNET

Submitted in Partial Fulfillment of the
Requirements for the Degree
Doctor of Philosophy

Supervised by James Frederick Allen
Computer Science Department
University of Rochester
Rochester, New York

Garrison W. Cottrell was born at a very early age on October 24, 1950, in Lake Forest, Illinois. Blissfully unaware that he was surrounded by a sea of Republicans and meat, he managed a happy childhood, rising to the level of Junior Assistant Scoutmaster in Troop 42, North Shore Area Council. He graduated from Lake Forest High School in 1968, and left for Cornell University in the fall. Cornell was a completely different experience than Lake Forest, although he was not "influenced by the Russians" as alleged by his mother. His freshman year the Blacks took over the student union, demanding more Black Studies programs. The student body responded by taking over the ROTC building, Barton Hall, in a group that numbered around 5,000 people. The following three years were spent demonstrating and learning about life, but alas, not much about his majors, sociology and mathematics.

Surprisingly enough, he graduated on time in 1972 with a respectable average. This may not seem like much of a feat, but some of his friends from those years have only recently graduated. Thus, he suddenly found himself in the real world, and the revolution had not taken place. What to do? He re-entered Cornell immediately and received an MAT in math slightly late, in 1975. After several years of school bus driving, ice cream scooping, rough carpentry and auto body work, it was time to reassess and perhaps to leave the mecca of Ithaca, N.Y. In the fall of 1977 he entered graduate school in Computer Science at Syracuse University. After two years there, where he learned that "all AI has accomplished in twenty years is LISP." he transferred to the University of Rochester in the summer of 1979.

Initially encouraged by the amount of hair in the department, he has now become disillusioned: his departure will cleanse the department of its last remnant of the sixties. While at Rochester, he was a University Fellow in 1979-1980, T.A.'ed Numerical Analysis *twice*, and held various other R.A. and T.A. positions. In conjunction with Steve Small and Lokendra Shastri, he wrote the two papers on connectionist parsing, and several papers on

UNIVERSITY LIBRARIES
CARNEGIE-MELLON UNIVERSITY
PITTSBURGH, PENNSYLVANIA 1521

connectionist models on his own. He has also distributed several abstracts on Connectionist Dog iVlodelling, based on his dog and long time friend. Jellybean. Also while at Rochester, he pursued his hobby of mid-sixties slant six automobiles. After deciding the fate of his 1965 Valiant, "Prince", 1966 Dart, "Art", and 1965 Fury, "Hell hath no$^{M}$, he plans to move to San Diego, California to search for less rusty incarnations of his dreams.

**Acknowledgements**

Many people have contributed to this work over the years. I would first like to thank my advisor, James Allen, for unflagging support through many hard times, and for hours of discussions. I would also like to thank my committee: Mike Tanenhaus for discussions on psycholinguistics that seemed like they could go on for hours; Jerry Feldman for many of the insights that form the core of this work; Steve Small for early collaboration when no one else was willing, leading to much of the work in Chapter 4, and (last but not least) Gary Dell for continued optimism, encouragement, and for instilling in me an appreciation of Cognitive Science. It has been an especially rewarding experience to be under the guidance of this rare combination of talent and energy.

I would also like to thank Patrick J. Hayes for putting up with connectionism long enough to help me clarify some of the ideas in Chapter 7. He has been a valued critic throughout the years. Many students also contributed to discussions on this work: Lydia Hrechanyk, Sanjaya Addanki, Lokendra Shastri, and Mark Fanty all had considerable influence on my work. I would also like to thank Mark Brucks, Lokendra Shastri, Steve Kaufman, Roger Meike and Mark Fanty for their work on the simulator and graphics.

I would like to thank the excellent staff at the Computer Science Department, Rose Peet, Peggy Meeker, Suzanne Bell, and the one who really runs the place, Jill Orioli for all of their patience over the years with procrastinators such as myself.

For material support, I would like to thank the National Science Foundation for the portions of their grants MCS-8209971, MCS-8203920, and IST-8208571 that kept me in generic beer and excellent computational surroundings.

For nroff support and help, I would like to thank Lee Moore, Mark Kahrs, and Tom LeBlanc.

Among the moral support contingent, Beth Zimmerman, Tom Gentile, Irene Allen, Amit Bandopadyay, Steve and Dara Kaufman, Steve Kaplan, Rich Pelavin and Diane Litman stand out, not necessarily in any order. Also, Yannis Aloimonos stands out as a constant source of Marlboros. To everyone I've forgotten, sorry, I have proper name anomia.

Thanks to my dog, Jellybean, for spending more hours than he cared to meditating in my office and learning foolish tricks like calling the elevator.

Thanks to Kathy Roth for making finishing worthwhile,

I also would like to thank Officer Burl G. Osborne of the Seneca County Sheriffs Department for allowing me to continue on my way to delivering my thesis to the dean without penalty. *I* now own a radar detector.

Finally, thanks to Lyn Frazier for writing a thesis worth emulating and quoting:

> ... the progression of a thesis resembles a brief excursion into a large city; one simply spends too much time getting where they're going, and then not enough time looking around. The problem, of course, is that afterwards one suspects that if they'd known the city better, they might have been able to begin closer to where they left off.

This thesis is dedicated with love to the memory of Lydia M. Hrechanyk, a fine friend and a fiercely independent soul. Godspeed

## Abstract

A new architecture for representing parsing of natural language is described which conforms to psycholinguistic, neurolinguistic and computational constraints. The parsing model uses a particular spreading activation or neural network scheme called connectionism which entails a massive number of appropriately connected computing units that communicate through weighted levels of excitation and inhibition. Such an architecture adds considerable constraints of its own which serve to explain some constraints at the functional level. The model accounts for psycholinguistic data on the access of word meanings, recent neurolinguistic data on agrammatism, and some of the apparent parsing strategies of normals.

# Table of Contents

# List of Tables

## Figures

# CHAPTER I

# INTRODUCTION

## 1.1. Motivation and Introduction to the Problem

While this is a Computer Science thesis, it is in the area of Artificial Intelligence (AI), and within AI, it ties on the interface to the growing field of Cognitive Science. Cognitive Science is a discipline whose goal is to understand human cognition through combining the insights of the fields of Computer Science, Psychology, Neurophysiology, Linguistics and Philosophy. The work reported here is motivated by constraints and results from the first four fields, and applies these to a model of the human sentence processing mechanism with an emphasis on the problem of lexical ambiguity. Thus there is a great deal of material in this thesis that is not traditionally within the domain of Computer Science.

Lexical ambiguity is a problem that was rarely attacked directly in early work on Natural Language Understanding (NLU) in AI. However, it is perhaps the most important problem facing an NLU system. Given that the goal of NLU is understanding, correctly determining the meanings of the words used is fundamental The problem is not one that appears only in strange sentences devised by linguists. In an informal study, Gentner (1982) found that the 20 most frequent nouns have an average of 73 word senses each; the 20 most frequent verbs have an average of 12.4 senses each. Small (1978) lists 57 senses for the word "take". Not to be outdone, Hirst (1984) reports that "go" has 63 meanings listed in the Merriam Webster Pocket Dictionary. The tack taken here is that it is important to understand how

people resolve the ambiguity problem, since whatever their approach, it appears to work rather well. The model described here resolves ambiguous words in what we[1] believe to be a clean way, and is at the same time neurologically and psychologically plausible.

### 1.1.L Ambiguity Defined

Lexical ambiguity is is of two types, syntactic and semantic. Syntactic lexical ambiguity simply refers to ambiguity of *category,* e.g., Noun vs. Verb. For example, *bark* is both the sound Jellybean makes and the stuff that gives him enough grip to get six feet up the side of a tree in pursuit of a squirrel. This is to be distinguished from *structural* ambiguity, which refers to sentences which have more than one phrase structure tree assignable to them. Winograd's famous example is *put the block in the box on the table,* which can be assigned two structures depending on whether [M]in the box[M] modifies "block" or not. We will not address problems of structural ambiguity in this thesis; however, we do provide a mechanism for this when it is resolvable by semantic information; see Chapter 5.

Semantic ambiguity is of two types. *Polysemy* refers to words whose several meanings are related. For example, the two uses of fell in *Allende's democracy* **fell** *to CIA backed generals* and *John* **fell** *and hurt himself are* similar in meaning, but not literally the same (polysemy, as pointed out by Hirst (1984), often blends into metaphor). *Homonymy* refers to words whose various definitions are unrelated, as in the two uses of ball in *they danced till dawn at the* **ball** versus *this dog can be entertained all day with a* **ball.**

Semantic and syntactic ambiguity are orthogonal, since a single word can have related meanings in different categories (as in can *of fruit* vs. *to* can/rw/7), or unrelated meanings in different categories (as in / saw *the carpenter's* saw),

[1] **I use "we" in this thesis because I find the use of "\" somewhat egotistical-sounding, and avoiding the use of the first person leads to a plethora of passive constructions. The reader may think of "we" as referring to me and my dog, Jellybean, who was a major contributor to my thoughts on connection^ models, but due to University regulations, cannot be cited as a co-author.**

or both (/ saw *the carpenter* saw//7g *with the rusty* saw). In this work we will pretend that polysemy is not different from homonymy, treating related meanings as distinct, although the representation of lexical relations will include more links and shorter paths between related than unrelated words.

In order to resolve ambiguity, an NLU system has to take into account many sources of knowledge. For example, often categorial ambiguity can be resolved on syntactic considerations alone, as in / **can** *do it,* where the only possible syntactic class of **can** is Verb. The system described here will handle ambiguities resolvable in this way. Some sentences are *globally ambiguous* in this respect, e.g., in *His* **will** *be done,* the category of **will** is either Verb or Noun, depending on who is speaking and where, i.e., a minister in church, or a mechanic in a garage. We will not consider a mechanism for context of this type in the thesis.

People appear to use semantic sources of information for categorial disambiguation as well, although this sometimes leads them astray, as in *the old* **man** *the boats* (the old people operate the boats). Although there is a syntactic frequency argument as well, one explanation for the "garden path[M] nature of this sentence is that the semantic representation of "old man" overrides the proper syntactic interpretation. The fact that such semantic garden path sentences exist is some evidence that the semantic representation of a word can influence decisions concerning its syntactic representation.

Semantic ambiguities often require global context for their resolution as well. For example, *democracy* can mean "a system where a dictator rules by force" if the speaker is a government official referring to a country with strong economic and military ties to his own, or it can mean "a system where governments are elected by the people" if the speaker is a high school history teacher. However, often all that is required is *local context,* specifically, the context provided by the rest of the sentence. For example, in *bob* threw *the fight,* whether **fight** refers to "propelling" something or "intentionally losing"

something is determined by the presence of **fight**. The system described here will resolve such ambiguities.

### 1.1.2. Religion

A fundamental premise of this thesis is that the time is right for the interdisciplinary development of a computational theory of human language comprehension. The claim is that there are both sufficient constraints based on hard data, and an adequate computational theory that can incorporate these constraints. Throughout the thesis, the model is guided by and related to the psycholinguistic and neurolinguistic literature, and is presented as one explanation of some of that data. In building such a model, it must be possible to form a clear correspondence between elements of the theory and elements of the world that the theory attempts to explain. It is precisely on this count that existing theories have broken down: how do the symbol structures and symbolic inference schemes of computational models relate to the structures and processing strategies that people use for the same tasks? The answer in many cases is that the correspondence is at a *functional* level. The functions performed by the program must be performed by a human in some way in order to accomplish the same task. The claim advanced here is that in order to explain the wealth of psychological data on low-level language processing, the correspondence must be at a level below the functional; that the *mechanisms* involved in carrying out these functions must be considered if we are ever to have real explanatory power.

Considerations of levels of description have led us to reconsider certain of the basic tools and metaphors employed for theory construction in Cognitive Science. The approach taken here is to use a computational paradigm that is similar to the human brain in form and functional capabilities. The computational metaphor being rejected is that of the information processing school (Simon, 1969), which attributes some of the information processing capabilities of a computer to humans. This is not to say that they believe

humans do floating point multiplications. Rather, it is the view that symbols can be passed from one system to another (i.e., copied) and manipulated in a serial fashion. This may be appropriate for describing some of the behavior of a student solving physics problems, but for a model of sentence processing, we claim not. Some of the constraints that were considered in adopting this view follow.

### 1.1.3. Connectionist Models: A Gentle Introduction

The particular paradigm used in this work is the *connectionist* (Feldman & Ballard, 1982) version of neural networks. Formal definitions of connectionist models are given in Chapter 2. The purpose of this section is to motivate their use and to give an intuitive idea of their operation through an example.

#### Motivation

Francis Crick (1979) has pointed out the inherent differences between the conventional sequential computer and the human brain. His comparison (with some additions) is summarized in Table 1.1. We would like to draw our first set of constraints on a cognitive model from Crick's observations and knowledge of human physiology:

(1) The processing units are relatively simple; they should not be more capable than a neuron. This is not too great a constraint. Recent evidence

Table 1.1. Differences between the brain and digital computers

| | computer | brain |
|---|---|---|
| speed | fast | slow |
| order | serial | parallel |
| component reliability | reliable | unreliable |
| faults | fatal | no degradation |
| signals | precise, symbolic | imprecise, terse |
| programming | needs it | does it |

(Levy, 1982) shows that neurons are far from simple linear threshold units, for example. Some computation appears to be going on at the dendrites, outside the cell body. However, the "cycle time" of a neuron--how fast it responds to input--is on the order of 2 msec, or $10^6$ times slower than the fastest computers.

(2) Another strong constraint is that the brain's connections are fixed; very few new pathways are grown in the adult brain. What may change is weights on the connections, thus developing new pathways. However, this process is probably slow. We presume that this accounts for long-term learning, and that short-term associations are handled differently, through systems of dynamic associations (bindings) as outlined in Feldman (1982).

(3) The coinage of the brain is frequency of firing, thus the inputs (and outputs) cannot carry more than a few bits. This is perhaps the greatest departure from the typical information processing paradigm. There are not enough bits in firing frequency to allow symbol passing between individual units.

(4) Whereas some locations in the brain may control activity in others, decisions must be completely distributed: each unit computes its output solely based on its inputs; it cannot "look around" to see what others are doing, and no central controller gives it instructions.

(5) The model must be noise resistant and robust. Faults in individual units should not (ordinarily) degrade overall performance. While we do not address this constraint in our current implementation, we do assume that redundancy accounts for much of the fault tolerance.

(6) The number of processors and connections must be constrained: on the order of $10^{11}$ processors, with $10^3$-$10^4$ connections each.

The first question one might ask is how people can possibly perform multiple tasks (such as walking and talking) at the same time with these constraints. We must, based on the relatively slow speed of neurons, be able to

do a lot of computation in a small number of steps. This is Feldman's time step argument (given in Feldman & Ballard, 1982): mental events (such as accessing the meaning of a word) occur on the order of a few hundred milliseconds. Neurons have a "response time" on the order of a few milliseconds. Then such mental events must only take around a hundred steps of computation. The only way this could be accomplished is through massive parallelism and the high connectedness of the system.

This was the motivation for the design of the connectionist paradigm, which is an attempt at defining a computational abstraction of the information processing capabilities of neurons. Connectionist models and ones in the same spirit have been used for models of visual recognition of origami figures with noisy inputs (Sabbah, 1985), speech production (Dell, 1980), learning control surfaces (Barto, Anderson & Sutton, 1982), and semantic networks (Shastri & Feldman, 1984). This thesis is an attempt at building a cognitive model of a complicated process with the connectionist paradigm. One of the goals of the work is to show how connectionist models are a good basis for cognitive models.

### A Simple Example

One of the best examples of a connectionist style model is the letter and word model of McClelland & Rumelhart (1981; Rumelhart & McClelland, 1982). Part of the model is shown in Figure 1.1. It is composed of a network of simple processing units that communicate by spreading *activation* over weighted links. A negatively-weighted link is called *inhibitory*. The network is divided into three levels. The bottom level units represent input features activated by the visual system encoded as parts of a letter orthography; these are positionally indexed. The orthographic units feed into units representing the letters formed by those features at the next level. These too are indexed by their position in the word. These, in turn, feed into units representing the words of which they are a part at the next level. Units representing different

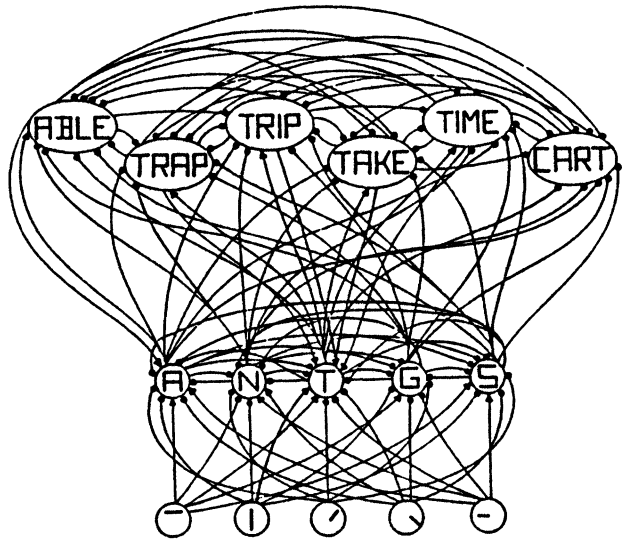



Figure 1.1. A few of the neighbors of the node for the letter "t" in the first position in a word. Links ending in dots are inhibitory, triangles excitatory. From (McClelland & Rumelhart, 1981).

letters in the same position inhibit one another, and features which are incompatible with units at the next level inhibit those units.

This model is used to explain a large body of psychological results which show that it is easier to detect the presence of letters when they are in the context of words than if they are presented alone (the word superiority effect). The effect is explained in this model as a result of feedback to the letter units from the word units they stimulate. This feedback does not occur when the letter is presented alone. It is an explanatory model in a strong sense: the units involved could correspond to neuronal level units, and some of the effects are a direct result of the architecture used. For example, there is a word superiority effect for pronounceable non-words (such as "mave") which is shown in their model to be a result of a "gang effect". There is a "gang" of word units that share many letters with the non-word (its "friends"). These get partially activated by "mave" and provide the necessary feedback for the superiority

effect. Their model predicts a similar effect for non-pronounceable non-words that have many friends. In fact, they found such an effect experimentally. This effect is predicted by the architecture in a way which may not have occurred to those designing symbol-passing models. The system also illustrates an important structuring technique used in designing such networks: division into layers of processing with connections only allowed between adjacent layers and within layers.

A connection machine provides us with a new metaphor for cognitive models to replace the Von Neumann machine. Furthermore, some researchers are developing actual computer hardware to operate with massive parallelism and low degradation of overall behavior in the face of local errors (Hillis, 1981; Fahlman, 1980). We would like to be able to use these machines advantageously. It is not clear that the best idea is to try to convert sequential algorithms to parallel ones; rather, we would like to start with highly parallel models. These machines also make feasible the idea of simulating our models in real time, which is rather difficult on a sequential machine.

## 1.2. Overview of the Model

In this section we present an overview of the model. First, a word of motivation. There appears to be a growing convergence of thought in Linguistics (Bresnan, 1982), Psycholinguistics (Rayner, Carlson & Frazier, 1983), Neurolinguistics (Linebarger, Schwartz & Saffran, 1983) and Artificial Intelligence (Hirst, 1984, Walker, 1978; Winograd, 1983) that the processing of syntactic and semantic representations should (or does, in humans) proceed in parallel. This point of view is adopted here as well.

What started out as a model of lexical ambiguity resolution ended as a model of (single clause) sentence comprehension, simply because in order to show how words are disambiguated, one has to specify the sources of the disambiguating information. The system consists of a three layer, four component network shown in Figure 1.2. The lowest level is the *lexical level*;

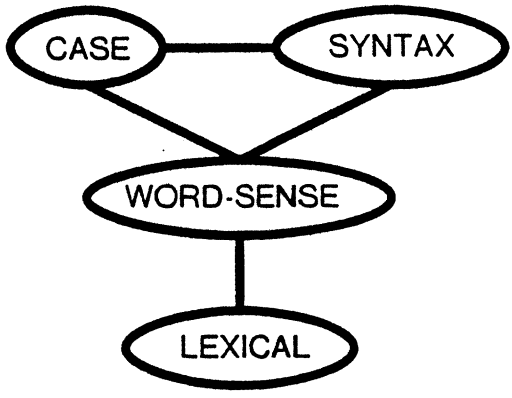



Figure 1.2. Overview of the model.

this is the input level for our model. It consists of a unit for every word in the language, and corresponds to the top level of the McClelland & Rumelhart network shown in Figure 1.1. While morphemic encodings are certainly possible, we have avoided this issue in this model. These units simply represent the spelling of a word; the definitions are represented at the next level. Reading a sentence is simulated by activating units at this level sequentially, with a model-dependent delay between them.

The lexical level units activate nodes at the *word sense* level representing the various definitions of the word. Connections are unidirectional from these units to all their possible senses at the word-sense level, so that a lexical unit excites its definitions and then decays rapidly. The definitions are positionally buffered at the word sense level. They are represented, for the purposes of this thesis, as a unit for the syntactic class (shared by each meaning of that class) and a unit for each meaning, labeled with an "awkward lexeme" (Wilks, 1976), e.g., INTENTIONALLY-LOSE for the "threw the fight" meaning of *threw*. It is at this level that disambiguation is accomplished. The various definitions of a word occupying the same buffer position compete through a

mechanism described in Chapter 3. In order to decide on a meaning or syntactic class, the units in the buffer receive feedback from the next level up which reflects how well each meaning or syntactic class fits into the developing semantic and syntactic representations of the sentence.

The syntactic class units in the word sense buffer are connected to the syntax processing network. This network builds a surface structure representation of the sentence based only on the syntactic class of the words and semantic constraints on bindings of constituents to roles (discussed below). Features such as number agreement are not represented in the current system. Hence, disambiguations that can be accomplished using this information (see Milne, 1983) are not handled in this implementation. The representation of the sentence structure is through activation in units representing syntactic constituents and roles in those constituents. If a constituent plays a role in another constituent, it is connected to the role unit through a *binding* unit, which represents the assignment of the constituent to that role. In general, during the processing of a sentence, there is not a unique syntactic representation of the sentence parsed so far, so the binding units that represent alternative bindings of a constituent will compete with one another.

The units representing the meaning of a word in the buffer are connected to the semantic network. This network builds a semantic representation of the sentence based on case relations (Fillmore, 1968). Cases represent the semantic roles required by the verb of the sentence. For example, *break* requires at least an Object to be broken, along with optional cases such as Agent and Instrument. In *John broke the window with a hammer*, *John* fills the Agent case, *the window* fills the Object case, and *a hammer* fills the Instrument case. We posit an "exploded case" representation; that is, we use several hundred case roles that are more specific than Agent, Object, etc., but fall into those classes (see Fahlman, 1979). At present, this is as far as we go towards a semantic representation; since a case representation is limited to a single clause, so is the model. Word meanings that can fill a case role signal this by

activation spreading through a lexicon where this fact is represented. Verb meanings activate their case frames (the set of cases determined by the verb's meaning). A conjunction of activation from a filler and a verb causes the case node to feed back to the filler and verb. Similar to the syntactic representation, binding nodes represent assignments of fillers to case roles; these compete until one case frame "wins." This is part of the semantic disambiguation mechanism; this is discussed in more detail in Chapter 4. The syntactic and semantic representations constrain one another through connections between the binding nodes. Thus bindings that are compatible are mutually supportive; incompatible bindings inhibit one another.

The operation of the model consists of a flow of activation from the lexical items (introduced in sequence) to their definitions in the word sense buffer. The meaning nodes in turn, activate the case nodes, and the syntactic class nodes activate compatible syntactic representations. The representation that fits the input best will then "win." Winning involves the formation of a stable coalition, that is, a group of connected nodes in which the overall excitation exceeds the overall inhibition. Our model can be said to have "worked" if the proper case roles form a coalition with the appropriate meanings for the sentence, and the correct syntactic representation wins. Since many sentences are ambiguous, the network will have to decide on an interpretation based on word sense frequency and relational knowledge expressed at the case level. We presuppose higher levels in the network for making general inferences and for long term memory. We must leave specifying these to future research. However, these levels provide the famous "context" (aside from local context) and we can simulate their effects by pre-loading the network with different biases.

A schematic representation of an example parse of a simple sentence is shown in Figure 1.3. This is a simplification of the actual nodes involved, but represents roughly the stable coalition of units that corresponds to the parse of the sentence. The important thing to note is that the communication between

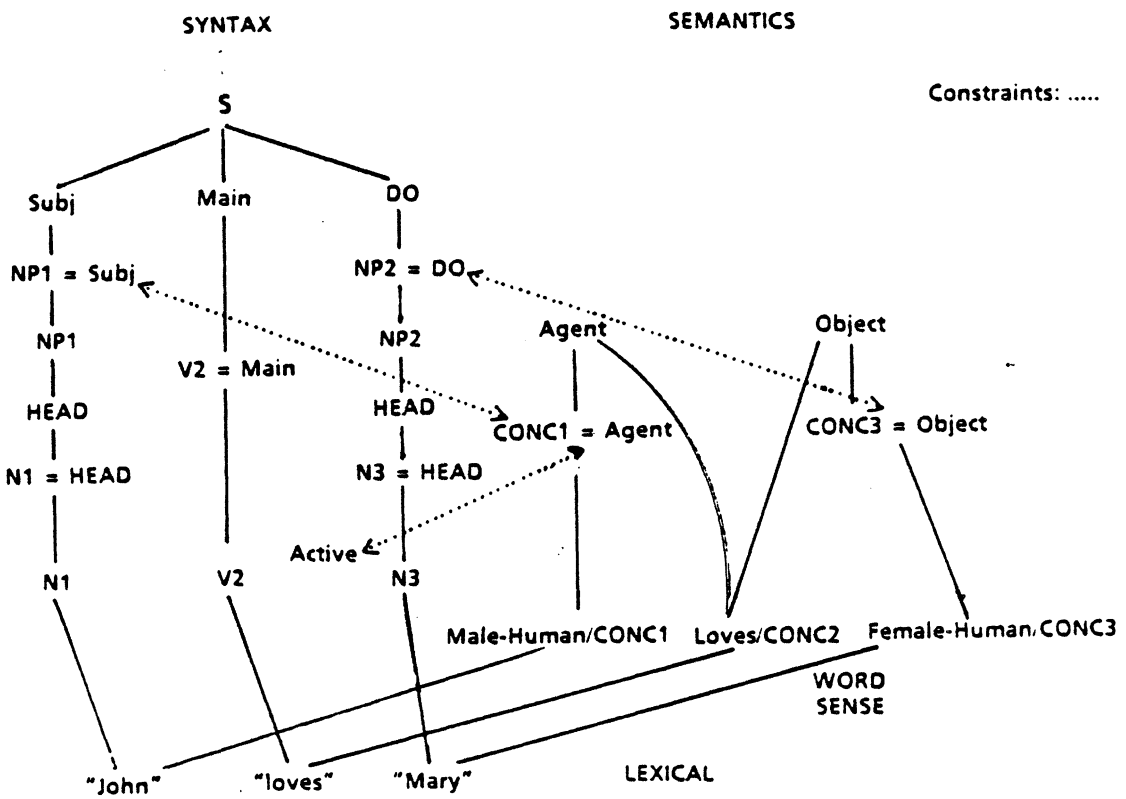



Figure 1.3. A schematic representation of the parse of "John loves Mary".

syntax and semantics is restricted to links between bindings and the pathway

through the word sefise level[2]. The fact that the verb is active and the first NP (Noun Phrase) is th Subject supports the binding of the first Concept to the Agent role.

This ends the overview of the model. It is appropriate at this point to mention what has actually been implemented. One version of the disambiguation mechanism in a word sense buffer position was implemented and is described in Chapter 3. A preliminary version of the semantic system was implemented and is described in Chapter 4. An inheritance hierarchy, which is a fundamental part of the lexicon, was implemented and is described in Chapter 7. The syntactic processor, along with the word sense buffer and lexical level, was implemented and is described in Chapter 5. What has not been implemented is a system for mapping the constraints between bindings in syntax and semantics, and the complete design of the semantic processor described in Chapter 4. The following section is a guide to the rest of the thesis,

### 1.3. Description of the Rest of the Thesis

Chapter 2 reviews previous AI work in lexical disambiguation, gives a formal definition of connectionist models, and briefly describes the simulator used for the thesis networks. Relevant psycholinguistic studies are presented in the chapters describing the parts of the model they pertain to. Chapter 3 concerns the process of *lexical access,* that is, the activation of the definitions of words from their lexical representations. Chapter 4 describes the semantic interpretation network, and reviews the semantic priming literature. Chapter 5 details the operation of the syntactic processor. Chapter 6 reviews neurolinguistic data relevant to the model, and discusses the implications of the model for that area of research. Chapter 7 presents a formal basis for connectionist inheritance hierarchies, which are an integral part of the semantic

[2]**The links between the bindings are "hard-wired" in the model presented here, but actually require dynamic linking in a complete model.**

interpretation system. Chapter 8 concludes the thesis with a review of t implications of the model for the various disciplines contributing to Cogniti Science.

# CHAPTER 2

# PREVIOUS MODELS

## 2.1. Introduction

In this chapter we review previous Artificial Intelligence approaches to lexical ambiguity, related models of cognitive processing from psychology, and also introduce connectionist models. Notably lacking from this chapter are results related to lexical ambiguity from psychology and neurolinguistics; these are included in the relevant chapters.

## 2.2. AI Models of Sentence Comprehension

### 2.2.1. Introduction: Syntax vs. Semantics

Although there has been considerable work in Natural Language Understanding, only a few researchers have directly attacked the semantic lexical ambiguity problem in the past, although the numbers are growing: most notably Wilks (1976), Riesbeck and Schank (1976), Small and Rieger (1982), and Hirst (1984). The answer proposed usually involves some notion of "context." Given enough context, the argument goes, nothing is ambiguous. Context is used either to constrain the search for the proper meaning of the word, or select it from a set of choices. If the wrong one is chosen, a program can backtrack. Psychological results to be discussed in the next chapter suggest that as far as psychological reality goes, programs which access all meanings and then select the proper one are closest to the human processor in operation.

It is impossible to separate approaches to ambiguity from approaches to sentence understanding in general; the AI approach to sentence understanding

can be divided into two schools: (1) those who follow the linguists, that is, they initially apply syntactic rules to get a first cut at the structure of the sentence, and then use semantic routines to build a meaning representation (Winograd, 1971; Marcus, 1979; Gigley, 1982); (2) those who apply semantic analysis to the words to map directly into some kind of meaning representation, only using syntax when necessary (Wilks, 1976; Riesbeck and Schank, 1976; Small and Rieger, 1981), These positions are rarely taken to their extreme, that is, no one believes syntax or semantics alone is enough, although their work is frequently misunderstood in this way (cf. Gigley (1982) for a recent example). The syntactic approach has the advantage of having more structure in the system, (syntactic analysis being well developed independent of semantics), but the disadvantage often of either having to carry along different possible parses that could easily have been resolved semantically, or making decisions that turn out to be wrong later. Applying semantics first usually results in programs whose structure is hard to follow because of the complexity of semantic interactions (and lack of syntactic structure!) but they are usually more oriented towards the disambiguation problem. We will consider each of the approaches in turn, and afterwards consider some approaches that could be classified as "mixed/[1]

With respect to the two types of ambiguity, lexical and structural, our work has concentrated mainly on the former, although we do have some things to say about the latter (see Chapter 5). Hence, in this review, we will mainly be concerned with lexical ambiguity, however, we will briefly mention approaches to structural ambiguity in each system. As a global comment on the syntax-first systems, it should be noted that most of them can handle word sense disambiguation that can be done by means of syntax alone (such as noun-verb senses), but must usually rely on the semantics component to disambiguate within-class and structural ambiguity.

### 2.2.2. Give me an S, Give me an N, ... : The Syntax First School

#### Winograd's SHRDLU System

Winograd's (1971) SHRDLU system, while being of the syntax first school, was one of the first to combine syntactic and semantic processing in a relatively graceful way. Rather than applying syntax to the sentence as a whole first, the two components ran conceptually as coroutines, with the semantic component verifying the partial results of the parser. As soon as a noun group was constructed, the semantic component would check it for consistency with the known world, and could instruct the parser to break it up differently. For example, in "put the blue pyramid on the block in the box" the parser will construct a noun group [blue pyramid on the block], which the semantic component then finds has no referent in the system's micro-world. The parser is then redirected to find [the blue pyramid] and parses the rest as a location Our system will not be able to to resolve ambiguities of this kind that depend on knowledge of the state of the world, since a model of the world is not incorporated in the system. However, certain kinds of knowledge about the world are incorporated in the exploded case system, since constraints on role fillers are really a kind of world knowledge. Integration of perceptual information as simulated in SHRDLU, and a model of the world are topics for future research.

Winograd's treatment of word sense disambiguation is to use selectional restrictions combined with semantic markers in the dictionary. This allows his program to make quick checks for compatibility, such as when an adjective requires an animate object to modify. Thus in his system, there is a certain amount of semantics in the syntactic component. His "semantic" component really corresponds to a world model, rather than linguistic-semantic information. This is a common theme in AI and Linguistics: what counts as "syntactic" and what "semantic"? The answer is usually a matter of taste.

**Marcus Style Parsers**

Marcus was one of the first AI researchers with a perhaps substantial claim of psychological reality: his PARSIFAL system is based on a limited processor and data structures, combined with a set of grammar rules that are "activated" as needed. It embodies his Determinism Hypothesis (DH), which says that natural language can be parsed by a mechanism that doesn't backtrack or maintain alternative hypotheses; no structure is ever built that is not part of the final parse. He uses lookahead and a "wait and see" strategy to avoid this sort of back up. He explicitly states that he does not consider word sense ambiguity, and if there is structural ambiguity, PARSIFAL will simply note it, choose one and go on. He claims that the structural choices made by PARSIFAL are just the ones humans would make, and where PARSIFAL makes an error, humans would fail also, in "garden path" sentences. PARSIFAL is mentioned here because a system based on it, ROBIE (Milne, 1982; 1983) which does attack lexical ambiguity will be reviewed next and because it is interesting to note that the parser described in Chapter 5 follows the three principles of parsing that follow from the DH.

By using examples from English, Marcus showed that the following three principles of parsing must be used by any parser following the DH:

(1) It must be partially data driven, but

(2) reflect expectations derived from the partially constructed parse tree, and

(3) use some lookahead.

The data structures used by PARSIFAL to achieve these goals are a three element buffer (which can hold constituents of any size), giving it two lookahead items (actually four counting the extra used for parsing NP's), and a stack of under-construction tree nodes, providing the syntactic context The interpreter which applies the pattern-action grammar rules is constrained to match the patterns against the contents of the buffer, the top element of the stack (the constituent currently being parsed), and the S node on the stack that

dominates the top stack element. As some elements enter the buffer, such as lexical items that signal the start of an NP, they automatically trigger the execution of some grammar rules, temporarily suspending other operations (like an interrupt). This provides the data-driven behavior deemed necessary by the DH. At other times the interpreter has to decide which rules to apply, since at times, there may be more than one applicable. This is done by prioritizing the rules, generally by the rule that "the most specific rule applies first." Constituents are built on the stack from elements in the buffer. As a constituent is completed, it is popped from the stack and dropped back into the buffer, where further rules can match against it, attaching it to the constituent that is now at the top of the stack.

Milne (1982; 1983) has extended Marcus' parser to handle syntactic ambiguity. His claim is that by simply using multiple definitions of words (accessing them all in the course of a parse), and allowing patterns that match one or the other of the definition features (e.g., Noun or Verb) eliminate the others, many ambiguities are resolved. By adding number agreement tests, he claims that most syntactic lexical ambiguities can be handled. He also further reduces the abilities of the parser by restricting the patterns of the rules to only match buffer elements, rather than the active node stack. However, the state of the stack is reflected in which rule packet is currently activated (rules in PARSIFAL come in packets, which can be activated or deactivated by other rules). For example, one packet is active when the matrix S dominates the top node on the stack, another when an embedded S dominates. He also claims to reduce the number of buffer positions to two, but some of his "cleanups" of Marcus' rules use three buffer elements.

Even with these restrictions, he still gets surprising coverage, and manages to eliminate many of Marcus' diagnostics, which were ugly rules that handled some function word ambiguity that related to structure, such as *that* as a complementizer or determiner. Given the psychological data on lexical access (see Chapter 3), his proposal has psychological reality as far as it goes- people

apparently do access all of the meanings of a word, especially when the meanings belong to different syntactic classes, no matter how biasing the context towards one of the meanings (Seidenberg et al. 1982). However, every time his parser cannot handle some construction, he claims it either doesn't occur often (which may, in fact, be a valid argument: presumably, people wouldn't use unparseable sentences regularly, and if his model is correct, then its unparseable sentences should not appear often), or the sentence would be a garden path. This often involves cases where a three element buffer is necessary. Also, his cleanup of the Marcus rules sometimes still uses a three element buffer.

The syntactic disambiguation mechanism employed by the model presented here is quite similar. All syntactic classes for an ambiguous lexical item are activated at once, and the ones that fit with the currently viable tree(s) are selected. However, a concurrent mechanism is used for semantic disambiguation which uses the same framework, giving the model here an advantage over Milne's. Also, since our parser is not committed to making decisions at every turn, the model can maintain alternative parses for a time.

**Gigley's HOPE System**

Gigley's HOPE system is an attempt at being psychologically and neurophysiologically plausible, motivated in part by psycholinguistic results, but mainly by her desire to have a system which is "lesionable" without reprogramming or redesign. She uses HOPE to simulate aphasic processing of language, and to suggest possible further experiments with aphasics. HOPE bears much resemblance to our own model, using neuron-like units, spreading activation, and parallel computation among all units.

An overview of HOPE is shown in Figure 2.1. We have taken the liberty of renaming some of her levels in accordance with our own usages. The phonetic level corresponds to our lexical level, but words are encoded in a phonemic representation, which leaves a better structural basis than our model

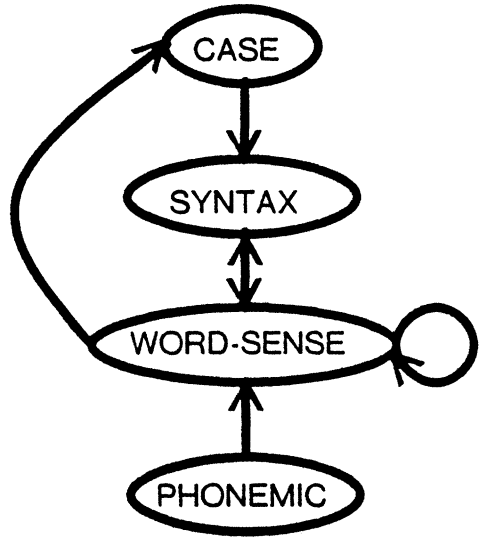



Figure 2.1. Overview of HOPE.

for incorporating bottom up input from a perceptual network. The main difference between her model and the one presented here is that she places the syntactic level between the semantic and the word sense levels, following the syntax-first school. Even though the model strongly resembles ours, there is no direct feedback to the word sense level from the case representation: it must pass *through* the syntactic level. All meaning nodes for a word are initially active in the word sense level after a word is "heard", and noun-verb disambiguation is accomplished by feedback from the grammar in the syntax network. She uses a categorial grammar as the mechanism to set up expectations for following words, so no semantic expectations are used. It should be noted that she allows an independent pathway to the case level. However, this level is constructed by procedures which build the case frame pieces as they are recognized by the grammar, and queries the user for selectional restrictions ("Is 'dog' animate?"). As such, this level is not neuronally plausible and has not solved the noun-noun ambiguity problem.

Her examples do not show any instances of noun-noun disambiguation, but this would presumably have to be done on the basis of this level. Finally, this level uses the general Agent, Object, sort of representation, without a mechanism for disambiguation on the basis of more specific constraints (see Chapter 4).

### Selman and Hirst's Connectionist Parser

In very recent work, Selman and Hirst (1985) have described a connectionist parser that uses the Boltzmann machine (Fahlman, Hinton & Sejnowski, 1983) computational procedure and the localist encoding scheme. That is, like the model presented here, a unit stands for a value of a parameter. Unlike our model, the units use a probabilistic rule for updating their state, and the probability density is parameterized by a global variable, the *temperature.* When the temperature is high, no matter what the input to a unit, the state change is a fifty-fifty proposition (the units have two states). As the temperature is lowered, a unit's state increasingly reflects its input. This process of "simulated annealing" is based on an idea reported in (Kirkpatrick, Gelatt & Vecchi, 1983). A prime benefit of the computational formalism is that it allows them to derive rules for connection weights between units. Aside from the different computational mechanism, the system they describe is very similar to the one reported in Chapter 5 in structure. The main difference is that their grammar is simply context free, which allows them to share more nodes than our grammar allows (see Chapter 5). Also, they have not yet attacked the problem of semantic interpretation in this framework.

## 2.2.3. Give me an Agent, Give me an Instrument, ...

### Wilk's Translator

Wilks' translator (Wilks, 1976) was one of the first to consider semantic word sense ambiguity. He believed that a "highly connected" (semantically) representation of a sentence must get most of the meaning and grammar right.

His meaning representation is based on sixty or so primitives; all other meanings are constructed by means of "formulas" made up of these, which contain preferences for surrounding constituents in the sentence. For example, the formula for "drink" in the verb sense, says that it is usually done by an animate agent, and usually involves a liquid object Finding the meaning of a sentence begins by matching templates (Agent-Action-Object triples) to the heads of formulas for the words in the sentence (the "head" is the type of thing this sense of the word denotes). For example, "Small men sometimes father big sons" has two assignments of formula heads, depending on the sense of the word "father":

KIND MAN HOW MAN KIND MAN

KIND MAN HOW CAUSE KIND MAN

and the template MAN CAUSE MAN matches the second (correct) assignment; no template matches the first. In cases where this is not enough to disambiguate, the preferences of the formulas are matched against each other. The assignment of formulas which satisfy the most preferences wins. This is what Wilks means by the most highly connected representation must be right.

In a sense, we are doing something very similar to Wilks in the way different meanings can reenforce each other through the case frame artifice — the case slots impose preferences on their fillers, and in turn, the fillers reenforce the verbs whose slots they fill, disambiguating verb senses (see Chapter 4 for examples of this).

ELI

The ELI parser of Riesbeck and Schank (1976) had a different approach to ambiguity. They believe that most examples of ambiguity are isolated sentences, and few sentences are encountered in the real world without some surrounding context. They conclude from this that parsing must be done in context, and a parser should never notice ambiguity. ELI used a system similar to ours in that a case grammar was used to represent the meaning of a

sentence. The words could either set up expectations for case fillers or predicates whose cases they could fill, or they could satisfy expectations for case fillers. The words thus would interact with one another and the context to maintain a narrow band of expectations, so that by the end of many sentences, the parser would have very specific expectations. The word senses are arranged in order with respect to the current context, and the highest "rated" one is chosen. ELI must therefore backtrack on a sentence such as "the old man's glasses were filled with water", but they too (along with Marcus) claim that people will have to backtrack also. In the system reported here, all word senses are activated in parallel. Only those that get reinforcement from case nodes will eventually "win". A subsequent failure of the semantic representation which forces restructuring will have the effect of inhibiting the previously "chosen" meaning, and allowing another to become active. Thus "backtracking" consists of higher level constraints forcing the network to form different stable coalitions. This comment is speculative, as there is yet no mechanism in the model for detecting semantic anomalies.

### 2.2.4. Mixed Strategies

#### Word Expert Parsing

Small and Rieger's Word Expert Parser (Small and Rieger, 1981) is based on the view that language knowledge is embedded in words themselves, rather than rules about words. Each word is procedurally represented as an "expert" which knows all of its possible senses and how they can fit with other words. These experts communicate among themselves until they reach an agreement on the meaning of the sentence. The relation between this and ELI is that words can generate expectations for other words, although this is not based on case relations, but on the way the word can be used idiomatically with other words. Small and Rieger view all words as being more or less idiomatic, and feel that a parser should begin with the irregularities of language rather than the regularities. For example the word "take" can mean different things if it is

followed by "off", "up", "out", etc., and the expert for "take" will set up expectations for those words, and react accordingly if one of them appears to its right. No choices are made until all the words in the sentence agree, so no backtracking is necessary. Also, this system lends itself to a parallel implementation at least at the word expert level. Our system imposes more structure on the interactions between the words through the case and syntax levels. The information about idioms such as "bug off" is contained in conjunctive connections from those two words to a node on the word sense level which represents the meaning.

While WEP and the work on which it was based have led to interesting results, there are reasons for questioning their underlying assumptions. Psychological data on lexical decision and aphasia, cited above, suggest that the processing mechanisms used in these models are not correct (Small and Lucas, 1984). Physiological evidence shows that the human brain functions in a fundamentally different way than do traditional computers and programs. We claim that the type of model architecture described below has a better chance of matching these kinds of data than does the more traditional symbol passing framework and that it employs a cleaner processing mechanism than WEP.

**Hearsay-II**

The Hearsay-II speech understanding system demonstrated the viability, if not the efficacy, of a highly distributed set of knowledge sources (KS's). The KS's are arranged in a hierarchical fashion with respect to the "blackboard", their medium of communication. The blackboard is two-dimensional; one axis represents time across the sentence, the second represents levels of abstraction, ranging from segmental to sentence. Knowledge sources are "fired" by matching with data on the blackboard that they can make hypotheses about. Once they get a chance to run, they place new data on the blackboard that represents a hypothesis of what their input represents at a higher level. Competing hypotheses are weighted based on the confidence the KS has in

them. New instantiations of KS's are created as input becomes available to them and they are allowed to run based on the dictates of the focus module.

The system presented in this thesis can be roughly viewed as replacing the blackboard with connections between all KS's, and replacing KS's with connectionist versions of their functions. All decisions about KS's "firing" are completely local. Thus the system proposed here is analogous to an implementation of the "neurological Hearsay" proposed by Arbib (1980).

**Hirst**

Hirst's (1984) semantic interpretation system explicitly attacked the problem of ambiguity, using a variety of methods that had never been under one roof before. The input is initially analyzed by a Marcus style parser, PARAGRAM, and partial output of the parser is analyzed by a semantic interpreter loosely based on Montague semantics (Dowty, Wall, & Peters, 1981) called Absity, in tandem with the parser's operation. Every syntactic object is in one to one correspondence with a semantic object[1]. Disambiguation of word senses and of case slots is done by a set of procedures with a superficial similarity to Small's Word Experts called Polaroid words. There is one of these per word or slot, each of which determines its correct sense in cooperation with the others. However, they represent several improvements on the basic idea of mutually disambiguating words represented by Word Experts. First, Word Experts had to handle the entire problem of parsing and semantic interpretation. In Hirst's system, syntactic parsing is factored out making the job of the Polaroid words that much simpler. Also, Word Experts were hand crafted pieces of code, often pages long. Polaroid words for meanings within the same syntactic class use the same procedure. Their operation is based on declarative knowledge about the word's meanings stored in the FRAIL (Charniak, 1981) knowledge base. Part of their operation in disambiguating

[1]This suggests that this approach may be useful for our system to exploit (see Chapters 4 and 5) as a way to ease the computation of the correspondence of semantic and syntactic objects: we intend to examine this possibility in the

their word's sense is accomplished by marker passing through the FRAIL frame representation system. When markers sent by Polaroid words intersect in the FRAIL knowledge base, the meaning of the words involved corresponding to the source of the markers are given preference.

Contrasting with a spreading activation paradigm, the markers are artificially restricted to spread by a fixed number of links, in an attempt to limit "false positives" at higher reaches of the knowledge representation hierarchy. This makes the abilities of the system to disambiguate limited by the design of the knowledge base - whether there are 5 or 6 links between related words, for example.

The major difference between his system and ours is the underlying processing mechanism used. The model presented here uses a spreading activation paradigm for all of the functions required, rather than a particular part of the system. As such, it is a more unified approach than Hirst's, where some parts of the system use marker passing, and others more conventional programs. While the Polaroid Words nominally ran in parallel, they were in a coroutine discipline and communicated through a shared memory. The parallelism in our model is of a finer grain, uses no shared memory, and is viewed here as an inherent part of the computational mechanism.

**Waltz and Pollack's Spreading Activation Parser**

Pollack and Waltz's (1982; 1985) spreading activation parser has much in common with our own. They too, use neuron-like computing units which compute by spreading activation and lateral inhibition, and they use a case representation that conflates cases and selectional restrictions. The major differences from the approach advocated here are:

(1) The network used for competing syntactic hypotheses is built by an interpreter as the input comes in, and is *then* run, rather than using a

---

future.

completely connectionist implementation from the start. The problems involved in building a fixed network which responds flexibly to the input are non-trivial, and are circumvented by this approach.

(2) There is no overall organization to their approach into syntactic and semantic modules, at least one is not in evidence in the published accounts. Rather, the syntactic and semantic systems seem to be overlapping in a hodge-podge, leading to the same sort of complexity inherent in similar approaches, such as the Word Expert Parser. Also, it is unclear how such an organization would explain the wealth of results from psycholinguistics and neurolinguistics which favor independence of these two systems.

(3) Word senses for the same word are mutually inhibitory. It is unclear whether word senses are shared between different words with the same meanings. If so, this system would imply that *I had a* **ball** *at the* **formal dance** would be hard to understand. If not, it is unclear how a word used with two meanings in the same sentence would be understood.

## 2.3. Related Cognitive Models from Psychology

### 2.3.1. Introduction

The model presented in this thesis has historical roots in predecessors from psychology, especially Collins and Loftus' (1975) spreading activation model, which itself is based on an earlier model from AI (Quillian, 1969). Collins and Loftus' model is described here for comparison purposes. Also, Posner and Snyder's (1975) two-component model of stimulus processing has a fair amount of support in the data (cf. Neely, 1977); although the model described in this thesis makes no provision for a separate attentional processor, a provisional mechanism for this is outlined here.

### 2.3.2. Collins and Loftus

Collins and Loftus' (1975) spreading activation model is an extension of Quillian's (1966, 1969) model of semantic memory and sentence

comprehension. A concept is represented as a node in a network, with relations to other concepts represented as labeled links. The links were labeled depending on the relationship between the concepts. Quillian posited five kinds of relations: (1) superordinate ("isa") and subordinate, (2) modifier links, which link a concept to concepts it modifies, (3) disjunctive sets of links, which encode alternate definitions of a concept, (4) conjunctive sets of links, which group concepts together that form a definition, and (5) a residual class that allows any concept to act as a relationship between any pair of concepts. These links could be nested or embedded to any depth, in order to be flexible enough to express anything expressible in a natural language. They were also marked with a number representing their importance to the concept.

A search through memory for a relationship between two concepts consisted of spreading "activation tags" from each of the two concepts across all of the links. These tags also contained a symbol representing the source node and the immediate predecessor it had come from, to allow an interpreter to trace the activation back to the source. When an intersection was found between two paths, the interpreter evaluated the path to see if it met syntactic constraints imposed by the sentence (this was a model of sentence comprehension). If not, other paths were evaluated in the order in which they were obtained. Semantic priming effects[2] (Meyer & Schvaneveldt, 1971) could be explained as the result of the path from the prime word being tagged by activation before the target word's search began, speeding intersections.

Concepts were stored in memory in hierarchical relationships, so that concepts could inherit properties from superordinates, but the hierarchy was not strict; if a robins were often seen flying, the fact that "robins can fly" was stored directly with the "robin" node, rather than inherited. (This was called the theory of "weak cognitive economy" by Collins and Loftus). This gave the model the ability to account for Rosch's (1973) typicality results.

---

[2]Semantic priming refers to the ability of a word to speed a subjects reaction to a following related word. These

Collins and Loftus extended this model with several additional assumptions to account for new data. Their embellishments were intended to make it more "quasi-neurological," although they still allow paths to be "evaluated" by some unspecified process. As there are analogs (and opposite choices) to their extensions in the model presented in the rest of the thesis, we enumerate some of their assumptions that are related:

(1) Activation spreads in a decreasing gradient from its source, inversely proportional to the strength of the link.

(2) The longer a concept is actively processed (by the serial attentional process), the longer activation is released from the node.

(3) In the absence of continued input, activation at a unit decays over time.

(4) Units are thresholded, so that they don't "fire" until activation from different sources sums to an amount above the threshold.

(5) The conceptual (or semantic) network is organized along lines of semantic similarity. The closer two concepts are related, the more links there are between them. Note that two concepts may be "close" in the sense of having a direct link between them, but unless there are many links between them, they are not closely related.

(6) The names of concepts are stored in a lexical network separate from, but connected to, the semantic network. These nodes are organized according to phonemic and graphemic similarity.

(7) A person can control whether she primes the semantic network, the lexical network, or both, depending on the task.

(8) In deciding whether two concepts match, evidence from different paths in memory sum together, and a decision is reached when the evidence exceeds a positive or negative criterion.

---

are fully reviewed in Chapter 4.

(9) If the decision is whether something is a subclass of something else, the superordinate connection between the two is enough evidence to make the decision.

(10) If comparing two concepts that share a superordinate, but are mutually exclusive, the "mutually exclusive" link is enough to force a negative decision.

An attempt will be made to relate these points to decisions made in our model as we proceed through the thesis. Some of the basic differences can be discussed now, however. The model presented in the later chapters does not posit an interpreter evaluating paths in the network. All decisions are made locally by each unit, but they may be revised by activation from units making opposite decisions. Thus this is a distributed decision making model, where a consensus is necessary for a global decision.

Second, the links in our model are not labeled in any way. Whenever it is deemed necessary to make the relationship between two concepts explicit and controllable, a node representing that relationship is used between the two concepts. This is not much different than Quillian's model, since he used the same device in many instances; we simply have eliminated all labeled links.

Third, there is no assumption in the connectionist paradigm (described in detail in the last section of this chapter) that activation is attenuated as it travels; this is controlled by the function computed by each unit. A unit may, in fact, amplify the signal, simply pass it along, use it to raise its own activation but not send it because of a threshold, or ignore it completely. The spread of activation is thus controlled by unit functions. Another difference, then, is that these units are not restricted to summing their inputs, and may have different functions from one another.

While no model of the "serial attentional process" is given here, it is assumed that this too (or something that gives the appearance of being a serial attentional mechanism), can be implemented in this framework without the

need to posit a homunculus that has global access to, and control of, the network. We retreat from the dangerous desire to speculate on the nature of this process and its implementation. However, a brief suggestion of some mechanisms that might be a part of this is given in the next section. Other relationships to Collins and Loftus' model will be mentioned at appropriate points in the thesis,

### 2.3,3. Posner and Snyder

Like many other researchers (cf. Collins and Loftus point 2 above) Posner and Snyder (1975) have described a model of long term memory retrieval that has two components[3]: The first is an automatic process that is not under the subjects control (and hence is strategy free), and incurs no resource cost. The first process is supposed to take place in cognitive domains where the relationships are often used and "overlearned," as in the relationship between a the set of phonemes that make up a word and the word itself. The second is an intentional, strategy dependent mechanism that taps the resources of the limited-capacity "central processor." Posner and Snyder assume the *logogen* model of Morton(1969). A logogen is a memory structure corresponding to words (or other perceptual events) that are familiar to the person. They are activated by input from feature detectors; when the input exceeds a threshold, the logogen fires (this is the same kind of spreading activation process assumed by Collins and Loftus). This corresponds to recognition of the word. Semantically related logogens share features, providing a mechanism for semantic priming. If a word is preceded by a semantically related word, it has residual activation from the shared features that makes it reach threshold faster than if an unrelated word had been processed before it. The spreading activation process has three properties: (1) it is rapid, (2) it occurs without intention or conscious awareness, and (3) it does not affect the retrieval of

[3]This discussion of Posner and Snyder follows the one given in (Neely, L977), one of the best supporting studies reiatmg to their model.

unrelated information.

The limited-capacity attentional mechanism also facilitates processing of logogens it is focused on. Its three major properties are: (a) it is slow acting, compared to the first process, (b) it operates only through conscious awareness and intention, and (c) it slows the retrieval of information unrelated to that in focus. The manner in which it slows retrieval is of import: It does not affect the automatic activation of unrelated logogens by the first process; only the *readout* of that activation. In order to retrieve information that is not in focus, attention must be *shifted* to the corresponding logogen. This attention shift is hypothesized to take time proportional to the semantic distance from the concept in focus (using some model-dependent measure of semantic distance).

Posner and Snyder's own test of their theory had some methodological problems, which Neely's experiments overcame in an ingenious way (see Neely (1977) for a discussion). His experiments were a strong confirmation of the Posner and Snyder theory. He developed a way to separate the contributions of the two processes experimentally. He used a lexical decision task (subjects have to decide whether a string of letters is a word or not) using a class name (bird, body, building, or xxx as a control) as a prime. In the bird-prime trials, most of the time the word targets were birds. However, on the body-prime trials, most of the time the word targets were building parts, and vice-versa for building-prime trials. The subjects were instructed to expect this. Now, by occasionally using body parts on the body-prime trials, and by using a variety of gaps between the prime and target, he was able to assess the time course of the effects of the two processes. As predicted by the theory, at short prime-target durations, given a "building" prime, when the subject was expecting a body part, "building" would prime "door", through the automatic process, but as the interval increased, response to "door" was inhibited by the attentional mechanism. Similarly, no facilitation for "door[M]" was found from "body" at short intervals, but facilitation did occur at longer intervals, when the slower acting attentional mechanism kicked in. There were several other predictions

from the theory about the various conditions in this experiment which were strikingly upheld; the reader is referred to the original paper for a thorough discussion.

In this thesis, no mechanism for the limited attention-process will be incorporated; it is assumed that unless a sentence is some kind of "garden path" (*the old man picked up his glasses and filled them with water*), where "backtracking" is required, or *double entendre*[4] where alternate meanings are so activated that they enter conscious awareness, sentence processing is in general an automatic process. However, in a complete model of sentence processing, some mechanism must be specified which accounts for the apparent "backtracking" behavior on garden paths, and suspension of decision processes on *double entendres.*

On the other hand, inhibition is definitely used in the "automatic" process of sentence processing as specified in this thesis. We differ from Posner and Snyder in this; automatic spreading activation in this model includes negative activation between competing alternatives for the definition of an ambiguous word (the most frequently used words are ambiguous; see Gentner, 1982). The alternatives garner positive feedback from the developing representation of the sentence, and the one with the most "wins." The process could conceivably be redesigned so that this competition is unnecessary; the alternative definitions could be set up so that lack of feedback causes decay. However, the results of at least one researcher (Lucas, 1984) appear to show inhibition from the subordinate meaning of a word to the dominant one. The lack of an inhibitory effect found in other studies of the process of lexical access using allegedly equi-biased words[5] (cf. Swinney, 1979; Seidenberg et al., 1982; discussed in the next chapter) may be a result of the combination of positive evidence (from the

[4]I used to have a job working for the Rural Electrification Department, hooking up power lines to outhouses for the Indians. I was one of the first people to *wire a head for a reservation.* - Utah Philips.

[5]Only Lucas (1984) has come up with a method that reliably assesses the frequency of a word's meanings.

lexical item) and negative evidence (from the winning definition).

A minor step toward attentional control can be mentioned at this point. In the model developed here, concepts that are mutually inhibitory have "hard-wired" inhibitory links between them. However, a different mechanism for mutual inhibition between concepts developed by Shastri and Feldman (1984), specifies a separate unit that computes the maximum of the output of the competing units, and sends that back to them as inhibition. This mechanism was developed as a way to reduce the number of connections needed for mutually inhibitory networks containing N units from N*(N-1) to 2*N, but it has other useful implications. By assuming control on the unit that actually sends the inhibition by enablement from other units, the mutual inhibition can be controlled. Some mechanism like this could be an integral part of attention, and could also be a part of an automatic process that used inhibition. In going from being a novice at some task, such as driving, to being a skilled driver, units which control the selection of alternative actions could originally require conscious activation, but as the skill is rehearsed, their action could become more and more routinized, to the point where the original links from the conscious mechanism have been overridden by automatic control networks (but not replaced; the conscious mechanism can still "take over" where decisions have to be made, as in passing a car, or backtracking on an ambiguous sentence). Obviously, this is just a sketch, but it makes the point that there are mechanisms for control of inhibitory processes in connectionist models.

## 2.4. Connectionist Models

Connectionist models consist of simple processing units connected by links. A unit or node is a computational entity comprised of:

{q}: a small set of *states*

p: a continuous value in [-1,1], called the *potential*

v: an *output*, in the range [0,1.0] in discrete jumps of .1 (11 values tota

i:a vector of *inputs*,

and functions for updating these:

p <- f(i,p,q)

q <- g(i,p,q)

v <- h(i,p,q)

We will term an application of these functions an *update* of the unit. Note that there is no interpreter for a connectionist network; all updates are done locally by each unit in parallel. There are no constraints on the functions that can be used, though they are usually kept simple. Finally, note that there is no mention of time in the definition. That is, in serial simulations these parallel networks, the units could be scheduled for updating in various ways: They could be kept in lock step (synchronous) or they could be updated in random order, with some units perhaps being updated several times before another gets a chance to be updated (simulating asynchrony).

A *connection* (or *link*), is an identification of an element of a unit's input vector with the another unit's output, along with a *weight*, a value between -1 and 1. Any value transmitted on the link is multiplied by the weight before it is passed to the unit. Links with negative weights are called *inhibitory* links. These are drawn with a small circle at their head in the figures. There is another kind of link, called a *modifier link*, modifier links. Modifier links are node-link connections that have the effect that when the unit at their tail has positive output, they block activation from crossing the link at their head. These are also drawn with a small circle at their head, but since they are always incident on other links, there is no confusion between them and inhibitory links.

The above definitions are relatively abstract, and since there are various instantiations of these definitions that are often employed in simulations and models, we will go into them here. First of all, since the input is a vector (rather than a set), we can think of a unit as having various input *sites*. For

example, inhibitory links are usually connected to one site. The potential function is then often broken down into three stages: Site functions, which are applied to the inputs at one site, an *evidence* function, which is applied to the result of the site functions, and an *activation* function, which computes the actual potential given the result of the evidence function, the current potential, and the current state. The activation function usually employs a decay parameter so that if the evidence goes to 0, so does the activation. A *conjunctive connection* is used to refer to two links that must both have non-zero input for the site function to pass a non-zero result to the evidence function. We will use an output function that thresholds the potential (thresholds are usually greater than 0, so negative activation is not spread) and rounds it to the nearest tenth (this is not always strictly followed; see Chapter 3). A unit that has non-zero output is called *firing.*

In the so-called *localist* connectionist models, (see Feldman & Ballard, 1982) an object in the domain is represented as a unit or small set of units (see Hinton & Sejnowski (1983) for a more distributed approach). The basic idea is that a unit stands for a value of a parameter (the *unit/value principle*) and collects inputs from other units which represent evidence for that value, positive or negative. For example, in vision, (see Ballard, 1984) a unit could represent the presence of an edge at a certain angle at a particular (x,y) coordinate on the retina. The unit's output represents its confidence, on a scale of 0 to 1.0 (in discrete increments of .1), that there is an edge at the point in the visual field that this unit refers to. In the sentence processing model presented here, for example, units will be used to represent words, word meanings, and relationships between them. Thus, at run time, a unit's output represents a confidence level in a *hypothesis* about the parameter it refers to. An output of 1.0 (or, the maximum possible after decay) represents certainty about the parameter value represented by the unit. The links between the units are weighted at the input sites, reflecting the importance to the receiving unit of the evidence from that link. For example, units representing different

values of the same parameter can be connected with inhibitory links in a so-called *Winner Take All* (WTA) network, which guarantees that one value eventually "wins". The importance of the evidence in this case is high, since competing values for a parameter are mutually exclusive. Thus, much of the information encoded in the network is contained in the connections between units (hence the name "connectionism").

Connectionist networks are a natural architecture for solving relaxation style problems. Their "activation passing" is iterative, and constraints between hypotheses can be easily encoded in the networks as positive or negative links between mutually compatible or incompatible hypotheses (represented as processing units). The typical way to go about building connectionist models is to first decide on which elements of the domain we want to model, choose a way to encode those as units, and then to wire the units together in such a way as to encode *constraints* between the elements. Finally, we must choose an appropriate function for combining the evidence.

The fact that no restrictions are made on the unit's functions allows arbitrary functions to be used, but the intent is that these functions can be replaced by more complex networks of simpler units. Thus a unit can be an abstraction of a larger set of units. Care must be taken here, though; because in simulations units are often kept in lock step, what may work when computed by one unit may not work when computed by several. These and other timing issues are not addressed in this thesis.

The implementations described in later chapters use an interactive connection network designer and simulator, ISCON (Small et al., 1982), written in Franz Lisp on the VAX 11/780. ISCON allows the user to define types of units, create, modify and connect them, and run simulations with or without graphic output. The definition of a type includes specifying input sites and associated functions, and the functions associated with computing the new state, potential, and output from the results of the site inputs. The simulator

allows the user to stop at any point and view the nodes of the network, and modify it if desired. Performance degrades for networks of over a few hundred nodes, so large networks are converted from ISCON to a representation suitable for use by a simulator written in C by Sumit Bandopadyay and Mark Fanty which runs around 500 times faster.

The following chapters use connectionist models to simulate the processes of lexical access, semantic priming, word sense disambiguation, sentence parsing and interpretation, and property inheritance in a semantic network. The fact that the paradigm can be used for all these tasks speaks to the flexibility and efficacy of the above definitions.

# CHAPTER 3

# LEXICAL ACCESS

## 3.1. Introduction

The process of accessing all of the information about a word, phonological codes, orthographic codes, meaning and syntactic features is called *lexical access.* We will mainly be concerned here with the access of meaning and syntactic class, and will use the term "lexical access" to refer to this process. It is useful to distinguish three stages the processing of lexical items, of which access is the second stage: decoding the input and matching it with a lexical item, accessing the information about that item, and integrating that information with the preceding context These are termed prelexical, lexical and postlexical processing, respectively. An important research question is discovering whether, to what degree, and through what channels these levels interact Does each level receive the completed output of the previous level (the "modular" view"), or can processing at one level affect processing at adjacent or even more distant levels (the "interactive" view), or is the answer somewhere between these extremes?

## 3.2. **Psycho linguistic** Studies of Lexical Access

Recent studies in lexical access have borne directly on the question of whether preceding context only has influence at the integration (postlexical) level or whether it can affect the lexical processing (or lexical access) level. The empirical question is whether the context of a sentence constrains the search for the contextually appropriate meaning of a word or not. The interactive view holds that context affects the lexical access level, so that only a

single meaning is accessed (the Prior Decision Hypothesis). The modular view holds that all meanings of the word are initially accessed, since the lexical access mechanism can't "know" what the context requires, and all meanings are then passed to the integration level, where context selects the proper one (the Post Decision Hypothesis). Early research produced mixed results, some studies supporting one hypothesis, some the other (Conrad, 1974; Foss and Jenkins, 1973; Holmes, 1977; Lackner and Garret, 1972; Swinney and Hakes, 1976).

Recent work by Swinney (1979) and others (cf. Tanenhaus, Leiman, and Seidenberg, 1979; Seidenberg, Tanenhaus, Leiman, and Bienkowski, 1982) has shown that the *time course* of these effects are important. The cross modal priming experiments discussed above provide a tool for studying lexical access. The subject is required to attend to a sentence containing an ambiguous word presented aurally, while performing a lexical decision task presented visually. This allows the decision task to be placed anywhere in the sentence, where it may be used (via the semantic priming effects) to measure the relative activation of the different meanings of an ambiguous word at different time points. If a word related to one meaning of the ambiguous word in the sentence is primed, we may conclude that that meaning has been accessed. This is superior to previous approaches in that relatively "normal" sentence processing is possible, and definite evidence of the activation of a particular meaning is obtained (rather than just an indication of increased processing load, as in phoneme monitoring experiments.)

When the target is immediately following an ambiguous word, Swinney found priming from both meanings, but when the target is three syllables later, (approximately 1000-1500 milliseconds) only priming from the appropriate meaning is found. This occurred even when there was strong biasing context for one meaning. An example sentence is: *Rumor had it that, for years, the government building had been plagued with problems. The man was not surprised when he found several spiders, roaches, and other* **bugs** *in the corner of*

*his room.* Both meanings of "bug" were found to be activated by the semantic priming measure. Swinney's initial experiments concerned noun-noun ambiguities with equi-biased readings. These results were also shown to hold for noun-verb ambiguities (Prather and Swinney, reported in Swinney, 1982) and strongly biased noun-noun ambiguous words (with one frequent and one infrequent meaning) (Onifer and Swinney, 1981). In the latter study, there was no significant difference in the priming obtained for the dominant and subordinate meanings at the end of the word. This suggests that lexical access may be independent of frequency effects. Further support for this hypothesis may be found in a study by Yates (1978), but his experiment did not employ an on-line measure.

An interesting variation on these experiments by William Onifer, reported in Swinney (1982), was done on schizophrenics. Schizophrenics have a well-documented symptom that involves their interpreting ambiguous words in terms of the most frequent meaning of the word, regardless of the use in the sentence. The results for normals replicated with schizophrenics except with respect to which meaning remained activated. That is, they accessed both frequent and infrequent meanings initially, but by three syllables later priming was obtained for only the most frequent meaning of the word, regardless of the sentential bias. Swinney notes that this is support for the view that lexical access is independent of and prior to the decision process that chooses the pertinent meaning of the word, since this decision process appears to be selectively impaired in schizophrenics.

These results have been confirmed in concurrent research by Tanenhaus, Leiman and Seidenberg (1979) and Seidenberg, Tanenhaus, Leiman and Bienkowski (1982). Their experiments used a similar cross-modal priming paradigm, but the task was to say the word ("naming") presented visually, rather than make a lexical decision. Also, the ambiguous word was the last word in the sentence. They studied the time course of priming as well, but in a much narrower time interval: the test word was presented at 0 and 200

milliseconds after the end of the ambiguous word. They found the same pattern of results as Swinney, multiple activation followed by selection, but they were able to show that selection happened within 200 milliseconds. Subsequent experiments by Lucas (1984) at more time points have further narrowed the decision time to between 125 and 150 milliseconds after the end of the word (but see the discussion below).

In addition to narrowing the decision window, Seidenberg et al. discussed two types of context which may differ in their effects on lexical decision. They contrasted *pragmatic* context, resulting from world knowledge with *semantic* context, resulting from associative and semantic relationships between word meanings, as in the following sentences.

(1) The man walked on the *deck.* (pragmatic)

(2) The man inspected the ship's *deck.* (semantic: ship -> deck)

(3) The man walked on the ship's *deck.* (semantic and pragmatic)

The first sentence contains a pragmatic bias towards the "ship" related meaning of deck; one is more likely to walk on that kind. The second sentence contains a word highly semantically related to one meaning. The third contains both types of information. They did experiments which contained a completely neutral context, a pragmatic context, or a semantic context. The results were that multiple access was obtained for neutral and pragmatic context, but selective access (only one reading active at the end of the word) for the semantic context. This result held for noun-noun ambiguities, but not noun-verb ambiguities, where multiple access occurred in all conditions (including syntactic context, such as *they all began to* ___ or *the carpenter picked up the* ___ ) These results are summarized in Table 3.1.

Our discussion of these results is based on the account given in Seidenberg et al. (1982). The selective access found for noun-noun ambiguity is contrary to the findings of Swinney (1979), where multiple access for noun-noun ambiguities was obtained in a strongly biasing context. However, there

Table 3.1. Summary of Results of STLB's Experiments

| Context Type | Ambiguity Type | Outcome |
|---|---|---|
| Neutral | Noun-Noun | Multiple Access |
| Syntactic | Noun-Verb | Multiple Access |
| Pragmatic | Noun-Noun | Multiple Access |
| Priming | Noun-Verb | Multiple Access |
| Priming | Noun-Noun | Selective Access |

are several differences between the two experiments which may explain the discrepancy. First, Swinney's experiments used the lexical decision task rather than naming, which may be subject to "backwards priming" from the target to the ambiguous word (see Koriat, 1981). These effects appear to be found in lexical decision tasks, but not naming (Seidenberg, Waters, Sanders & Langer, 1984). Second, Swinney's ambiguous words appeared in the middle of a sentence, rather than at the end, as in the Seidenberg et al. (1982) experiments. The fact of a word being sentence-final may make a difference to lexical access. If so, this effect would have to apply differentially to noun-noun ambiguities, and only in a semantic context. The most probable explanation, however, according to Seidenberg et al., is that Swinney did not differentiate and control for the two types of context distinguished in their experiments. It appears that many of his materials did not contain strongly associated lexical items, and when they did, the associate was often more than four words away from the ambiguous word. If the priming effect decays rapidly, then the priming words may have been too far away to affect lexical access.

It remains to discuss why there should have been the selective access result in the first place. Seidenberg et al. (1982) attribute the result to intralexical priming by the strong associate preceding the ambiguous word. It should be noted that the only meaning of "intralexical" in this context that makes sense is actually "intrasemantic": A single *meaning* of the word, and not the lexical representation of the word itself, is primed. Also, as pointed out by

Hirst (1983), priming cannot spread from the meaning to the lexical item itself, or priming of all meanings would result (eventually, at least). This is in accordance with results that semantic priming is not transitive (DeGroot, 1983) as discussed in the next chapter. So, the appropriate meaning of the word is primed by the associated word's meaning and blocks or inhibits the alternate reading. This is a result of the "organization of semantic memory." An interesting question here is: Where is the line between "semantic memory" and pragmatic knowledge? What constitutes semantic vs. pragmatic context? The only definition we have is an operational one, that semantic context is one that shows priming and pragmatic context is non-priming. The only clue we have to the difference is that semantic context seems to require a lexical item associated with one reading of the ambiguous word. Thus semantic context has to do with the mental representation of word definitions interacting, while pragmatic context seems to require inference.

However, Lucas (1984) has shown that pragmatic context primes meanings as well. She used the lexical decision task to look at more time points in the decision process than any of the previous studies, and also looked at words with more than two meanings. An innovation of her work was to use non-homographic homophones[1], which allow a precise measurement of the frequency of the various *meanings* of the word. This can just be obtained by looking up the entry in the Kucera & Francis (1967) word frequency norms for the spelling of the meaning of interest. The time points used were: the *beginning* of the priming word, to measure the effects of context, and 100, 125 and 150 milliseconds. According to her results, the time course of meaning access is roughly as follows: If the context pragmatically biases one meaning, then there is priming for the pragmatically biased meaning at the beginning of the word (before it is heard). Thus this study is evidence that pragmatic context can prime in addition to lexical context. By 100 ms after the end of

[1]"Non-homographic homophones" is a fancy way of saying "words that sound alike but are spelled differently." such as *heir* and *air*.

the word, the unprimed meaning is active as well as the primed one. However, if the subordinate is appropriate, it inhibits the dominant meaning, whereas if the dominant is appropriate, the subordinate just decays.

This is consistent with the hypothesis that frequency information is used to select the most appropriate meaning unless semantic information overrides that decision. In the case of appropriate-dominant, no competition is necessary since the right meaning was chosen by the frequency strategy. In the appropriate-subordinate case, the semantic information must overcome the frequency information. This implies it is the semantic information that is causing the inhibition. It is unnecessary for the dominant meaning to inhibit the subordinate when it is appropriate, since it will win anyway, based on frequency. The system appears to have applied a principle of least effort.

There are a few criticisms which may be leveled at this study. First, it uses the lexical decision task, which as mentioned above, is subject to backwards priming. As is discussed in the next chapter, lexical access and naming appear to tap different levels of the system; that is, lexical access reflects processes happening at a post-access level. This could explain the finding of a pragmatic priming effect here when it was not found in the STLB study, which used naming. Second, the pragmatic context effect finding is compromised by the fact that the subjects still heard the word (the target onset coincided with the onset of the prime word) which by the time they hit the key (which actually occurs several hundred milliseconds after the word) may have been partially due activation from the word. Finally, a close look at her materials reveals that about a third of the sentences contained lexical items preceding the prime that were semantically related to or predictive of the pragmatically biased meaning of the prime. Fischler (1977(a)) has shown that words that are semantically but not associatively related to the target still have a priming effect in the lexical decision task (see the discussion in the next chapter). This compromises the result that pragmatic context is responsible for the priming.

The next section describes STLB's model of the lexical access process in order to provide a foil for the model presented in the following section.

### 3.3. STLB's Model of Lexical Access

Seidenberg et al. (1982) present a model to account for their results. It is based on four implications they draw from their research. First, that the results support a modular, autonomous account of the lexical access process. The only contextual effect, selective access of noun-noun ambiguities, was due to intralexical priming, which is local to the lexicon in their view. Second, the results indicate that there are at least two classes of context which interact with word recognition in different ways. This suggests that there may be more types of context, and thus a complete model would specify a taxonomy of context types and their representations. Third, the difference in the results for noun-noun and noun-verb ambiguities suggest that syntactic information is encoded in the mental lexicon. Indeed, in any computer model of parsing, syntactic information about a word is always encoded in the lexicon. It is difficult to imagine where else it would be. The point is not vacuous, however. What they are really interested in is *how* syntactic information is encoded. It is possible that a word's syntactic class is encoded with the lexical representation or with the meaning representation. The distinction will become clear in the comparison of their model, which chooses the former, to the one advocated here, which chooses an intermediary position. Finally, the results suggest that studies which illuminate the time course of comprehension processes are essential to decoding the structure of the processor(s).

STLB's model is a combination of Morton's (1969) logogen model and Collins and Loftus' (1975) spreading activation model. A lexical logogen governs recognition, and is connected to semantic memory where it activates its meaning(s) via spreading activation. The meaning nodes are accessed in the order of relative activation levels, which reflect frequency. The meaning nodes may be primed by the access of words highly related to one meaning, which is

the only exception to the automaticity and autonomy of lexical access. They posit that if there are large differences in activation due to frequency or priming, then selective access obtains. Since this has been shown to be false for frequency by Onifer and Swinney (1981), then perhaps they would attribute this result to the nature of the task (lexical decision) and relegate this to a post-access effect.

In order to account for the difference in noun-noun vs. noun-verb results for semantic context, they posit that nouns and verbs have different connections to the semantic network from the lexical network. However, they also assume that they have different nodes with identical recognition procedures in the lexical network (see Figure 3.1). Now, the story goes, for noun-verb ambiguities with one meaning primed, both nodes get recognized, and both meanings are accessed. In the noun-noun case, if one meaning is primed, that *pathway* is followed first. Note that this explanation implies serial evaluation of the possibilities in the noun-noun priming case.

### 3.4. A Connectionist Model of Lexical Access

Our model for the lexical access process is shown in Figure 3.2. We show the network for the word "deck", since it is at least four ways ambiguous, with

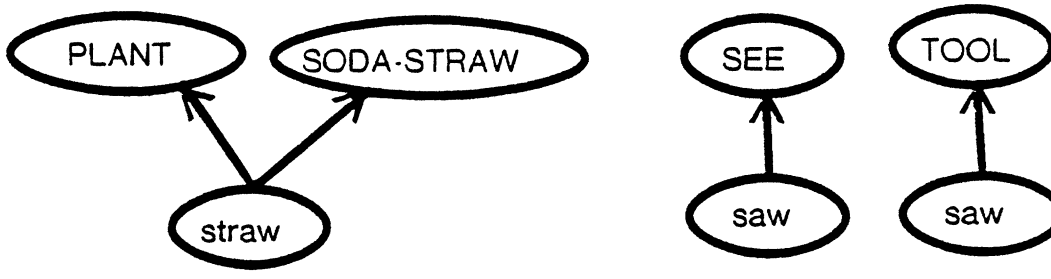



Figure 3.1. STLB's model of lexical access.

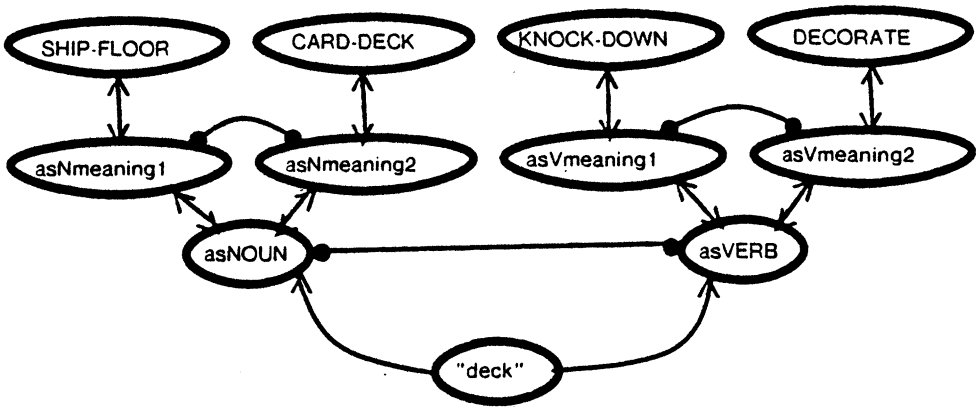



Figure 3.2. Our model of lexical access.

two noun meanings and two verb meanings. The network for a noun-noun ambiguous word would just consist of the left half of this network, (right half for verb-verb), and a noun-verb ambiguous word would just have the outer "V" of seven nodes. The lowest node represents the lexical item and is assumed to be activated by a phoneme or letter recognition network (such as the one described in McClelland & Rumelhart, 1981). The top row of nodes represent the various meanings of the lexical item and are assumed to be connected into a sentence processing and/or an active semantic network. The lexical node activates its meaning nodes through a discrimination network, starting with the grossest distinctions possible, then progressively finer ones. Note that the most efficient way to do this is to make two-way splits between large classes of alternatives (divide and conquer), if possible (but we don't assume all splits are two-way), since the inhibitory connections are minimized this way[2]. We assume that syntactic information is more discriminatory than

[2]For n nodes to be mutually inhibitory, we need O(n*n) inhibitory connections. If we arrange put them at the leaves of a binary tree discrimination network, we need O(n) inhibitory connections, but 2*n-1 units, so we are making a connection/unit tradeoff. This is motivated by the observation that we can't assume the network is pre-wired (in humans). The connectionist model of forming connections involves recruiting units that are on the path between two units (Feldman, 1982). Thus, by conserving connections, we are really conserving units as well.

semantic information, i.e., that the distinction into "noun" and "verb" divide the possibilities up more than divisions based on meaning.

The alternatives at any discrimination inhibit one another, so that one path through the network eventually "wins" and the meaning nodes that the other paths support fade away. This is the decision process. We assume that this process is driven by feedback to the meaning nodes from higher levels in the network. In the case of a biasing sentence, this would be from higher level nodes representing the role that meaning could play in the sentence (see Chapter 4). (We also assume there is not a direct link to such role nodes.) In the case of semantic priming, we assume the meaning node is directly primed by a node representing the relation of the priming meaning to this meaning, as in the Collins & Loftus (1975) model of semantic priming. The unfortunate meaning node that does not get top down feedback (or does not get as much) will not be able to provide as much feedback to the pathway nodes which activated it, and its pathway will be inhibited by the pathway nodes that do get more feedback.

In order to account for the modular nature of lexical access, we had to make two simple assumptions about the units. We assume that the units are thresholded (i.e., they can collect activation but they will not fire until they cross threshold, as in Morton's (1969) "logogen" model) and that top-down links have lower weights than bottom-up links. A unit may thus be activated above threshold by bottom-up evidence, but not by top-down evidence. This combination of threshold and weighting acts as a barrier to top-down information affecting lower level processes by itself, such as recognition. It may come in to play, however, *after* recognition of the lexical item has begun, in the decision process. This assumption is independently motivated at all levels of our networks by the need to prevent top-down activation from hallucinating inputs.

An interesting feature of this network is that the meanings themselves are not mutually inhibitory. When one considers constraints between units, there is no functional reason to assume that a particular *meaning* in isolation from its source (a particular lexical item) is not compatible with another meaning. However, it *is* reasonable to assume that the *assignments* of different meanings to the same use of a word is inconsistent. Indeed, if the meanings themselves were mutually inhibitory, we would expect that a word with the same meaning as an inappropriate reading of a previous word in the sentence (assuming the meaning node is shared) would be harder to process than a control word However, as we saw in the Swinney experiments, a word related to the unbiased meaning is not suppressed after decision, it is just not primed. For example, mutual inhibition at the meaning level would imply that it should be hard to understand "I had a *ball* at the *formal dance.*" Our model would predict, however, that people would be slower at processing sentences such as "I had a *ball* at the *ball.*"

### 3.5. An Example Run

We present the result of running the model using the ISCON simulator (Small et al., 1982) in Figure 3.3. It will be helpful to refer to Figure 3.2 to understand the trace. We include a driver node, m1 (not shown), that provides constant feedback to SHIP-FLOOR throughout the simulation. (In a complete model this would be a node representing one of the types of SHIP-FLOOR. For example, m1 could be PART-OF-SHIP, activated by the context prime "ship's"). The units average their input from three sites, bottom-up, top-down, and inhibitory. The first two sites take the maximum of their inputs, and the inhibitory site uses a parameterized arctangent function to enhance the difference in inhibition between two units that are close to each other in activation level. This helps avoid the problem of two units getting into equilibrium without one suppressing the other below threshold. Bottom up weights are 1.0, top-down are .5, and inhibitory weights are -0.5. The threshold

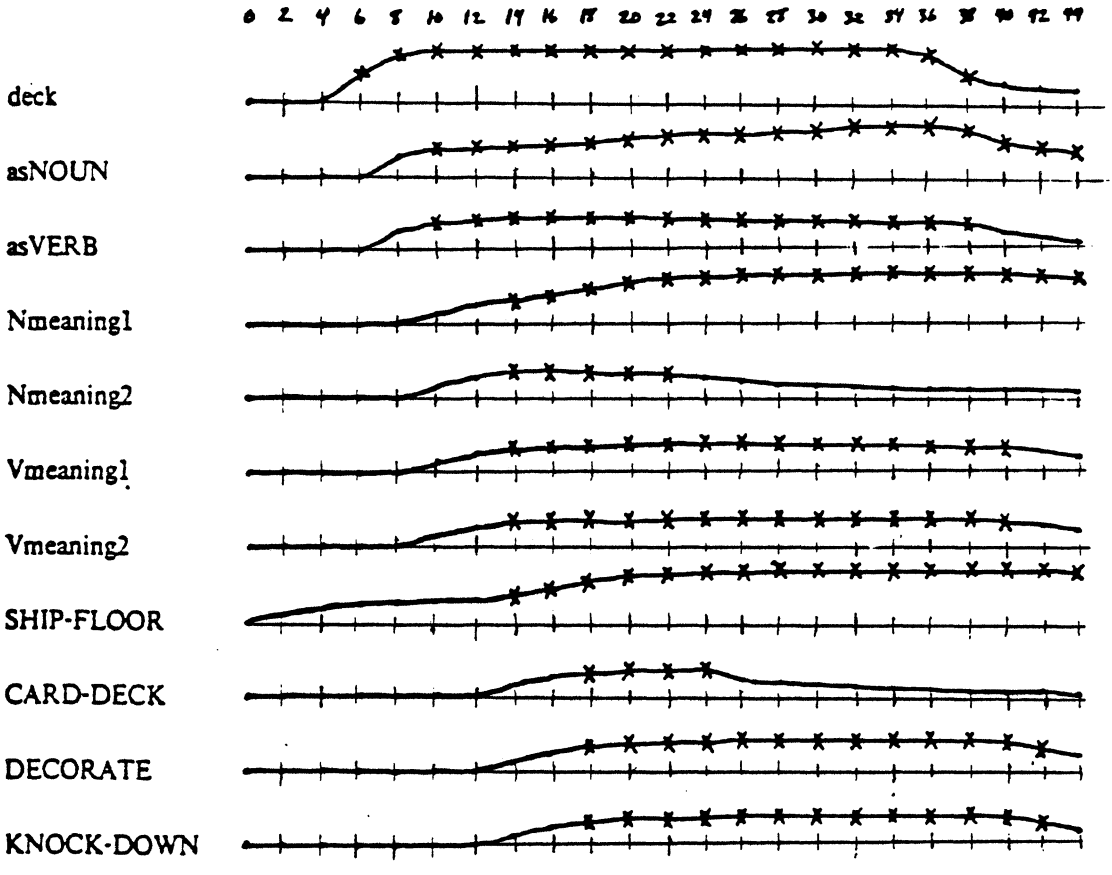



Figure 3.3. Trace of the simulation of the network in Figure 3.2 (x means firing).

is set at 0.3. The potential function is similar to the one used by McClelland & Rumelhart (1981).

At step 5, SHIP-FLOOR has been primed by the context prime m1. Now we activate "deck", and continue feeding it for 30 steps. We skip along to step 13 where the semantic discrimination nodes (the "as Xmeaning" nodes) have just fired (not visible at Figure 3.3's resolution), but their activation has not spread to the meaning nodes yet. Notice that SHIP-FLOOR has been primed now to near threshold. Thus the bottom-up activation from "as Nmeaning1"

causes it to fire in step 14, while the other meaning nodes have to accumulate more activation for several steps before they will fire. This gives SHIP-FLOOR a chance to increase the relative activation of nodes that are on its feedback path, before the other meaning nodes fire. This allows the nodes on that path to begin to win over their competition so that by step 24, "as Nmeaning2" has been suppressed. This results in CARD-DECK fading from lack of support. Also, "as Nmeaning1" is no longer inhibited by "as Nmeaning2", so it rises, giving more support to "asNOUN", which then suppresses "asVERB". Later, KNOCK-DOWN and DECORATE fade due to lack of support from "asVERB".

### 3.6. Discussion

This model makes several claims about lexical access. First, decisions within a syntactic class happen "nearer" the meaning nodes than decisions between classes, so the incorrect meaning nodes fade faster when within the same class as its competitors than when its competitors are in different classes. Thus noun-noun decisions are faster than noun-verb decisions, as was seen in the sample run. Thus it predicts that verb-verb ambiguities, which have not been tested (to our knowledge) in the psycholinguistic literature, will act like noun-noun ambiguities. However, the STLB study used homonyms (words with unrelated meanings). Verbs tend to polysemy (related meanings). Because this may affect the results, we restrict our claim to verb-verb homonyms.

In order to explain different context effects we have to mention some claims about context. We saw how in our model feedback does not flow freely downward from the priming node (m1) through the meaning node (SHIP-FLOOR) because it is blocked by SHIP-FLOOR's threshold. However, when activation comes up from "deck" through the other nodes, the barrier is broken, and feedback flows down. If we assume that higher levels of processing act the same way, then in the case of pragmatic context, no

feedback to meaning nodes would occur before the meaning node actually fired because it is too far away in the network. By this time, multiple access has occurred, and a target word to be named (say, "spade") can take advantage of the priming from all of "deck"'s meanings.

The case illustrated in the sample run was one of priming context with a noun-noun ambiguity (ship's->deck). Here, the contextual priming word is so closely related to one of the ambiguous word's meanings that they are not far away in the semantic network and direct priming of the meaning occurs (eg., "ship's"->SHIP-PART->SHIP-FLOOR). A decision will be reached much more quickly than in the case of pragmatic context, where the feedback has to come from "farther away" (semantically) in the network. Therefore, the model claims that there will be faster decisions in strongly priming contexts. Yet, contrary to STLB, multiple access did occur in our version of a semantic context. We rely on our prediction of the relative speed of ambiguity resolution in different contexts to resolve this. Naming presumably requires at least two stages, recognition and production. The word to be named is presented at the end of the contextually primed ambiguous word. If the decision for the ambiguous word is over before the recognition stage of naming completes, the naming process could not make any use of priming from the alternate meaning of the ambiguous word[3]. Thus we claim multiple access always occurs, and if the word to be named were presented slightly *before* the end of the ambiguous word, we would see multiple access.

Finally, in the case of four way ambiguous words such as "deck", the model predicts the pattern of results seen in the sample run: In a semantic context, the alternate meaning within the same class would be deactivated first, then the meanings in the other class.

---

[3]This claim can be relaxed if we assume the barrier (the threshold) is "leaky", that is, with enough top-down activation, the meaning node might actually cross threshold before it got bottom-up activation. It would then be able to prime the semantic decision node below it to the point where the alternate meaning never gets active. This can be made to happen by using more priming from mL. The model is therefore in the "chameleon" class with respect to this particular issue.

Several differences from the STLB model should be pointed out. First, arcs in this model's network pass activation, and are automatic (like wires). There is no notion of "following an arc" as in the STLB model. Second, there is no reason to assume a different set of connections to the semantic network, or separate nodes with identical recognition procedures. Third, this model places the decision processes in the access procedure itself, while requiring the information for the decision to enter the process as feedback from the semantic network. Finally, we assume that multiple access *always* happens, but the speed of the decision process varies with frequency and priming.

## 3.7. Conclusion

We have designed and built a model of lexical access within the connectionist framework that accounts for the data and makes empirically verifiable claims. This model has several advantages over STLB's in that (1) we don't have to posit nodes with identical recognition procedures, (2) the decision process is motivated by the discrimination network and the difference between nouns and verbs "falls out" of that representation, and (3) it is a computational model. With respect to Artificial Intelligence, we have a parallel model which tackles the major problem of the decision process between the possibly many meanings of a word. An interesting problem now is specifying the levels above this which drive the decision process. These are the subjects of the following chapters.

# CHAPTER 4

# A BASIS FOR SEMANTIC DISAMBIGUATION

## 4.1. Introduction

Our model of lexical access made no assumptions about the source of the disambiguating information. In this chapter we specify one of the origins of disambiguating feedback: the semantic portion of the model. The basic idea is to use case structure (Fillmore, 1968; Schank, 1972; Bruce, 1975; Cook, 1979) to represent the meaning of the sentence, and also to provide the feedback to those word meanings that best fit with the developing structure.

There is considerable data on disambiguation at the sentence level. However, since many of the studies disagree, it seems premature to use their data when even the lowest level organization of the lexicon is not yet known. Given that Seidenberg et al. (1982) hypothesize that semantic priming led to the anomaly in their results (selective access for noun-noun ambiguities), a survey of the semantic priming research is in order, to provide insights into the semantic organization of lexical memory supporting this hypothesized function. This will lead to a preliminary model of semantic priming. Coupled with the linguistic work on case grammar, the following hypothesis emerges: Case roles are cognitiveiy real objects that contribute to the disambiguation process using the same mechanism as semantic priming. Cases constitute a semantic relation between words, and the process than the one used by our model of semantic priming suffices to explain both phenomena.

This chapter will thus begin with a review of the semantic priming research, and a brief introduction to case grammar, since this is the major

linguistic tool the model uses in interpreting the "meaning" of a sentence.

Following this introductory material, we sketch a model of semantic priming consistent with at least some of the data, leading into a description of the semantic disambiguation model. Examples from a preliminary implementation demonstrate the feasibility of the approach.

## 4.2. The Data

### 4.2.1. Semantic Priming

#### What is it?

The semantic priming effect, discovered by Meyer and Schvaneveldt (1971), has been used for several years as a window into the cognitive representation of the lexicon (Meyer, Schvaneveldt & Ruddy, 1975; Warren 1972). The basic effect is that subjects are faster and more accurate at responding to a word (the *target*) in some task (e.g., reading the word aloud, referred to as a *naming* task) if the subject is previously exposed to a semantically related word (the *prime*). For example, Meyer & Schvaneveldt (1971) found that subjects are significantly faster at saying "DOCTOR" (that is, the onset of their response is faster) if it is preceded by "NURSE" than if it is preceded by "BREAD". They are also faster at classifying "DOCTOR" as an English word (a *lexical decision* task) if they have just classified "NURSE" versus just having classified "BREAD". Their reason for studying this effect was to determine how context affects the processing of words. However, researchers interested in the cognitive organization of the lexicon can use this type of evidence as clues to that organization. Other studies (discussed below) have shown that "higher order" relations also produce priming in certain tasks.

#### How might it work?

Most models of sentence processing (cf. Forster, 1979; Garrett, 1978) posit several *levels* of processing (see Figure 4.1), including (a) encoding the stimuli

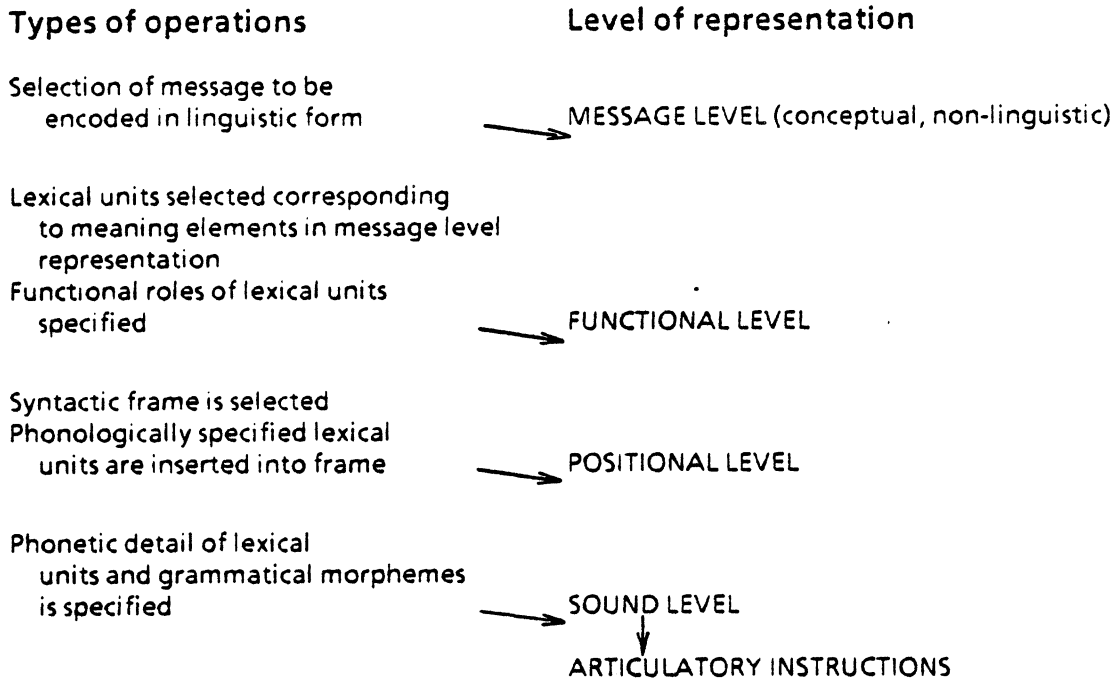



Figure 4.1. Levels of representation in the language processor (from Saffran, 1982).

into phonemes and/or graphemes; (b) assembling these into morphemes and/or lexemes; and (c) their ultimate translation into the syntactic and "message" levels. The lexical level entries are often postulated to be organized in some sort of semantic network (Collins & Loftus, 1975; Forster, 1979) in order to explain semantic priming effects. As such, it is often argued that the effects are solely due to processes at the lexical level, at the expense of making that level rather complex. As Forster says,

> ...semantic priming is an intralexical effect, rather than an interlevel effect. That is, there is no evidence to suggest that priming involves levels of processing other than the lexical level. Once we have postulated a semantic network defined over lexical entries ... we have given the lexical processor sufficient computational power to encompass the effect.

However, one could now argue for different effects depending on the relative levels of items within the semantic network. The appeal to levels of processing

has been traded off for a many-leveled level!

Forster goes on to say:

> It would be quite a different matter if it could be argued that the priming effect was produced by the action of the message-level processor on the lexical processor, so that the context effect of a sentence fragment and the context effect of a single lexical item were seen as different manifestations of the same phenomenon.

The model described here accounts for these two types of context as "different manifestations of the same phenomenon", but without the direct action of the message level on the lexical level. Rather, the process and structure which account for the phenomena are uniform among levels.

Returning to the discussion of the phenomenon itself, the *way* in which the effect arises, either through spreading activation (as in the Collins & Loftus (1975) model described in Chapter 2) or through some ordered search across links in the network (Forster, 1979) is still in dispute, but our connectionist orientation leads to sympathy for the spreading activation account. The effect is explained as the result of activation from the logogen representing the prime to the logogen of the target, lfpre-activatingtf it, so that it operates more quickly than in an unprimed state. Given the above discussion, it is of interest to discover the nature of the pathways that can give rise to the effect. That is, what kind of links can be crossed in the posited semantic network? How far can the activation spread? Does it go "down", "up", or "across" levels in the semantic network? While many of Collins & Loftus' original detailed hypotheses have been shown to be incorrect, these errors were based on their particular hypotheses about spreading activation, and do not implicate the notion itself.

**Possible pathways**

Many different relationships between words have been shown to elicit priming, leading to a picture of the organization of lexical memory as a richly interconnected structure. Table 4.1 provides a partial list. Not all of those listed are "semantic" relations. One difficulty in this type of research is controlling for the other types of priming relationships while trying to test for a particular one.

The earliest type of "semantic" priming found was due to associative relationships between words (Meyer & Schvaneveldt, 1971). These relationships are derived from word-association tests. Because of this, it is hard to say whether this type of priming is due to relationships in the semantic network hypothetically overlaying the lexicon, or whether this is simply an artifact of the test used to get the associations, which engages the speech production system. That is, association norms may just be due to frequencies in which the two words follow one another in normal speech. If the task used to obtain the effect is one which also engages the production system, such as naming, then it is suspect to attribute the effect to a semantic relationship.

Table 4.1. Priming Relationships (N = Naming, L = Lexical Decision)

| Relation | Example Prime | Example Target | Study | Task |
|---|---|---|---|---|
| Associative | nurse | doctor | Meyer at al. (1975) | L,N |
| Name Identity | APPLE | apple | Warren (1977) | N |
| Phonemic | hair | bare | Hillinger (1980) | L,N |
| Graphemic | couch | touch | Seidenberg et al. (1984) | L,N |
| Superordinate | bird | robin | Neely (1977) | L |
| Semantic | bread | cake | Fischler (1977(a,b)) | L |
| Perceptual | ball | cherry | Schreuder et al. (1984) | N,L |
| Conceptual | banjo | harp | Schreuder et al. (1984) | L |

However, the effect arises in the lexical decision task as well (Meyer & Schvaneveldt, 1971). Given that the stimuli pass through various levels of processing, the question arises as to which levels are affected by semantic priming. Meyer et al. found that associativity interacts with stimulus quality. That is, the priming effect is increased when the quality of the stimulus is degraded Assuming an additive stage model (Sternberg, 1969), one can conclude that stimulus quality and associativity affect the same stage of processing. This and other considerations led Meyer et al. to conclude that associativity affects the encoding stage. It is hard to interpret this result. Could perceptions be semantically encoded at a very early stage of processing? One does not have to go this far. The usual model (cf. Becker, 1980; Meyer, Schvaneveldt & Ruddy 1975) assumes only that the detectors are primed through semantic relationships to already recognized words, so that the perceptual processes that feed those detectors are not the locus of the effect

However, there is reason to believe that it is *not* the encoding stage which is affected by semantic priming. Meyer at al.'s argument depends critically on the assumption of an additive stage model. In this model, the information processor is assumed to operate in discrete stages, with each successive stage operating only after the previous stage has completed processing. If this assumption is rejected, as in McClelland's (1979) Cascade model, the argument does not hold up. The Cascade model maintains the stage assumption, but assumes that successive stages can begin to operate on the partial results of previous stages, that is, before they have completed processing. In such a model, interactions between variables do not imply that they operate on the same stage. If one stage is affected, this may be reflected in its output to succeeding stages. If the succeeding stages are affected by a different variable, the effects of the two variables can interact, through the interaction of the input changes to the later stage and the processing changes at that stage. Assuming such a parallel processing model, which is consistent with the connectionist paradigm, we can tentatively reject the idea that semantic

priming affects the encoding stage. This illustrates how the model one chooses as an analytical tool influences the interpretation of the results.

Task Effects

Another important consideration concerning the loci of priming effects needs to be discussed before considering the rest of the evidence. In recent years, there has been a growing body of evidence suggesting that the tasks of naming and lexical decision lap into different levels of processing. This was first proposed by Forster (1979). His idea is that naming reflects processing at the lexical recognition level, and lexical decision reflects post-lexical processing. This has since been successfully applied by other researchers (West & Stanovich, 1982; Seidenberg, Waters, Sanders & Langer, 1984). Forster's rationale for the distinction is that in order to say a printed word, it would seem sufficient to simply access its representation through the spelling, and from there access the phonological codes. The "meaning" level need never be accessed. In lexical decision, many non-words fit the phonological and orthographic constraints of English. Hence it appears necessary to access the meaning of the word to see if it has one. If so, the subject can decide it is a word. Given that the meaning does appear to be accessed in naming anyway, as evidenced by the Seidenberg et al. (1982) study, this can't be completely correct We will survey some of the research that supports the view that the two tasks access different levels, and then submit a hypothesis based on spreading activation that appears to bring the data together.

There is abundant evidence for the view that naming and lexical decision tap different levels of the comprehension system. Koriat (1981) showed that lexical decision is sensitive to "backwards priming". That is, facilitation occurs when there is an associative relationship between the target and the prime (but no "forward" association between the prime and target). This effect held only on early experimental blocks, giving way to forward association dominant priming on later blocks. Koriat interpreted this as evidence of an automatic

priming effect (the backwards priming) giving way to an expectation effect as subjects noticed the relatedness of the primes and targets. If so, it is hard to see why an "automatic" effect disappears on later trials. Whatever the interpretation of these results, they are not replicated in naming experiments. Seidenberg, Waters, Sanders & Langer (1984) showed that there are no effects of backward priming in naming.

Koriat (1981) also found no effects of prime-target association strength on the amount of facilitation in lexical decision. This was in a high validity condition, i.e., where the predictive value of the prime (in the sense that the subject can expect a related word, rather than a particular word) is high. Some investigators assume that under such conditions, the limited capacity attentional mechanism is deployed. In a low validity condition (that is, where the probability of prime-target association is low), there was no priming due to associativity in a long SOA[1] condition, and a mild effect in a short SOA condition. However, in a color naming paradigm (which appears to measure the same level as naming) Warren (1972) found significant effects of associative strength on the strength of facilitation.

Further evidence of the differences between naming and lexical decision is that lexical decision appears to be sensitive to strategic effects (that is, ways of doing the task that appear to be consciously applied) (Becker, 1980) whereas naming does not (Seidenberg, Waters, Sanders & Langer 1984).

The final straw in the argument rests on some recent investigators' more sophisticated interpretations of the term "semantic". Fischler (1977b) found that words that he judged to be semantically related (e.g. bread-cake) that were not associatively related produced an equivalent amount of priming as associatively related words in a lexical decision task. In fact, he found that semantic relatedness correlated better with priming strength than associative strength did. This suggests that researchers have been seeing a priming effect

[1]Stimulus Onset Asynchrony, i.e., the time between the offset of the prime and the onset of the target.

from semantic relatedness confounding their results with associated materials. More recently, Schreuder et al. (1984) distinguished between what they termed *perceptual relatedness* which is based on perceptual similarities, such as shape, color, size, etc, and *conceptual relatedness* which is based on more abstract properties such as functional properties, class membership, etc. By controlling these two dimensions, they were able to show that both perceptual and conceptual similarity primed additively in the lexical decision task, but only perceptual similarity primed in a naming task. This suggests that perceptual similarity information comes into play earlier in processing than conceptual information, given that naming latencies are generally shorter than lexical decision latencies. It also conforms to the hypothesis that naming and lexical decision tap different levels of lexical processing.

### A **Spreading Activation Account of Task Effects**

The point of view taken here is that the basic idea that naming and lexical decision tap different levels of the system is correct, but this must be modulated by consideration of the time course of activation spread through the system. As the duration of the prime and/or the SOA increases, there is more time for the effects of the prime to spread to different layers of the system (see Figure 4.2). Specifically, the point of view we will adopt is that naming at short prime durations demonstrates effects of lexical recognition processes, i.e., the graphemic and phonological codes, and at longer prime durations demonstrates continuously greater effects of the semantic organization of the lexicon. Also, the task of naming a word uses the language production system, whereas lexical decision does not. We hypothesize that some processes and pathways specific to production are thus engaged. For example, as mentioned above, words that often follow one another in speech may develop associations that are contingent on the engagement of the production system. On the other hand, lexical decision would not be affected by these pathways, except at long prime durations and/or long SOA's. Also, since naming does not require a

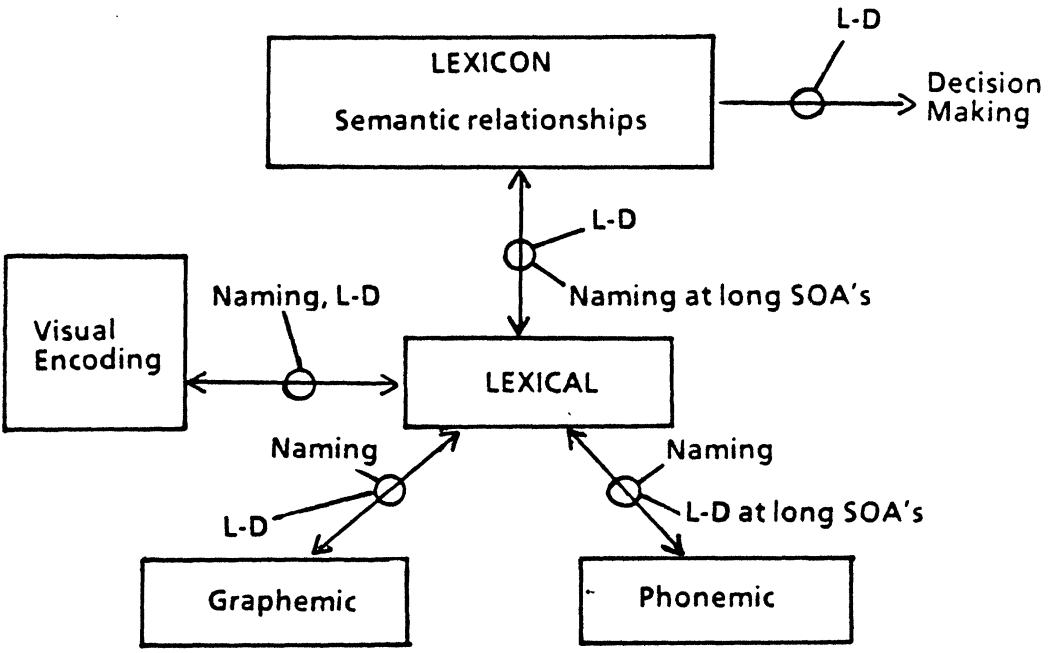



Figure 4.2. Loci of priming effects.

decision, but involves overlearned, automatic mappings, it is less likely to engage the limited capacity attentional processor hypothesized by Posner & Snyder (1975). As noted in the discussion of Posner & Snyder's model, Becker's (1980) results would lead us to expect that since the set of objects under consideration (all the words in a subject's vocabulary) is large, lexical decision effects would be facilitation dominant, i.e., the effects will be the result of facilitation of the related targets, rather than inhibition of the unrelated targets. Also, since no one has a detailed model of how decisions are arrived at, there are surely going to be effects of the decision making apparatus which can not be factored out.

A detailed model has not been worked out, but we also assume that prime duration and SOA have a different kind of effect (see Figure 4.3). While both

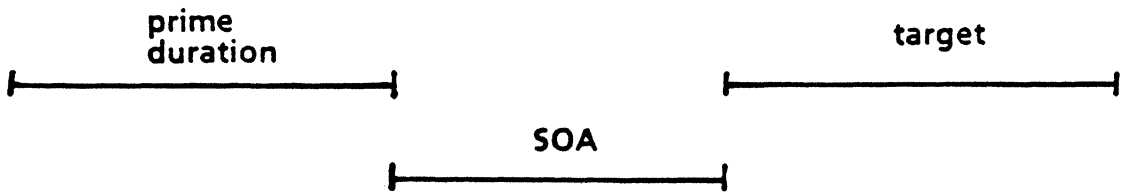



Figure 4.3. Stimulus Onset Asynchrony (SOA) and prime duration.

would give the activation a longer time to spread, increased SOA should give rise to weaker effects than increased prime duration. The reason is simple: while the prime is still being shown to the subject, the detector for that prime is still being driven by bottom-up input, and so will continually fire. On the other hand, with increased SOA, the activation will still spread, (we don't assume that activation is automatically attenuated at distances, as Collins & Loftus do), but the source will begin to decay after a while.

We are now ready to assess the research in terms of what it says about the cognitive organization of lexical (and possibly other) information. We assume that naming studies reflect the structure at or near the lexical recognition nodes plus the effects of the speech production system, and that lexical decision may also reflect influences attributable to the decision process, making data from lexical decision tasks somewhat harder to interpret with relation to the structure of the lexicon. These assumptions must be tempered by time course considerations. That is, the longer the SOA during a naming task, the "farther away" from the recognition node the priming effect may be coming from.

No study shows this better than Warren (1977). In a naming study, he varied the duration of the prime from 75 msec to 225 msec, with 0 msec SOA. He used associated prime-target words with various semantic relationships and found no effect of association strength. However, when he analyzed the results

taking into account the *type* of semantic relationship between the associated words, he found significant differences in the amount of priming at different prime durations. Synonyms primed when the duration of the prime was 75 msec, decreasing to zero at 150 msec (see Figure 4.4). Sex shifts (*boy-girl*) and weak antonyms (*soft-hard*) showed little priming at 75msec prime duration, increasing to 20 msec facilitation when the prime was on for 150 msec. One might take this as evidence for a Collins and Loftus style model. Synonyms are probably initially activated because they are strongly related to the prime, but have to be suppressed after a short time in order to avoid confusion with each other. This is especially true since these were strongly associated synonyms, pointing to a production system effect. That is, we suppose strongly associated synonymous words are arranged in a WTA which is only active during production, since the selection of which word to say would presumably

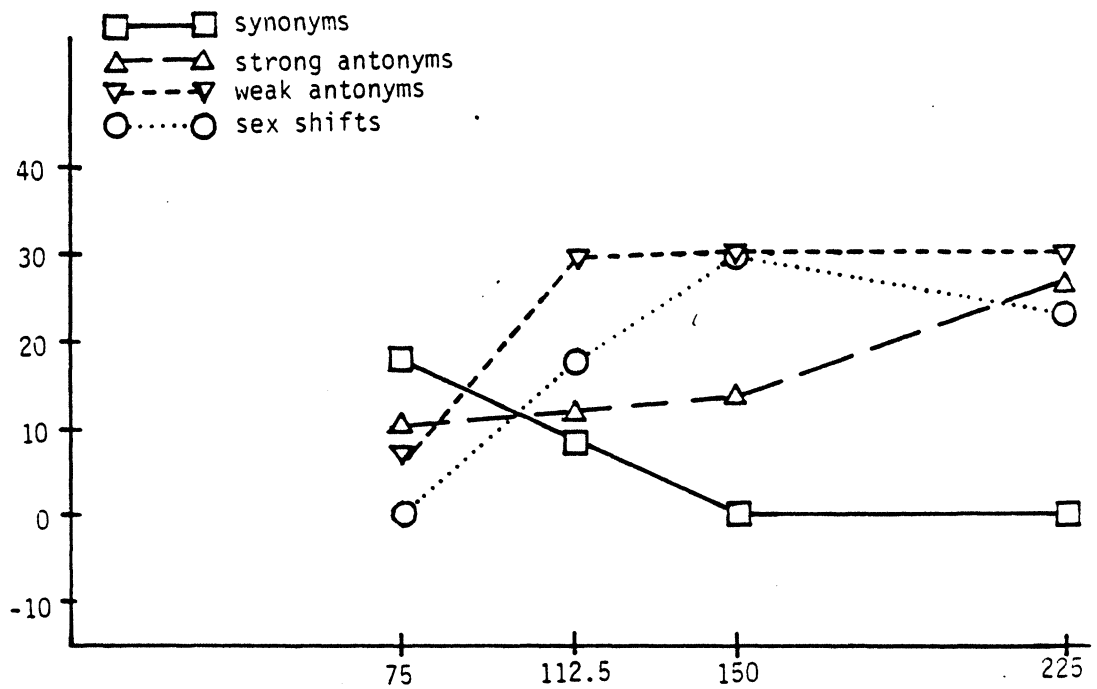



Figure 4.4. Warren's (1977) results: Msec of facilitation as a function of prime duration and type of semantic relation between prime and target.

select a set of synonymous words, whereas we would only want to say one of them (see Dell (1980; 1984) for a connectionist speech production system that would support such a process). Also, Warren found that strong antonyms were activated earlier than weak ones. One might conclude from this that the stronger relationship was primed earlier, supporting Collins and Loftus. But as Warren says,

> There is some question whether a definitive test of the Collins and Loftus (1975) hypothesis regarding the rate of spread of activation in memory is possible. They propose several factors that determine the state of activation at any particular unit in memory after the presentation of a prime: (a) the strength of the connection between the tested unit and the priming unit, (b) the time elapsed since prime presentation, (c) the rate of spread of activation, and (d) the distance of the two units in the network. It would seem that any observed level of of activation could be accounted for by these factors.

Also one might mention the function used by the unit to compute its activation from its input. Unfortunately, the model proposed here has similar problems. Refutable theorys apparently require strong simplifying assumptions. For example, one could make stronger assumptions about the organization of the memory units, i.e., about the number of units between the prime and target, and assume no other connections between prime and target. Then, concern arises about the strength of the connections between the units along the path. However, it seems that even with this, just the variation in the strengths of connections along the path could explain many things. Also, assumptions about the number of units along a path can be changed to match the data. We will not presume to provide an answer to these problems.

Warren has shown that an association does prime in naming, and that the time course of the priming varies with semantic relationship. This supports our hypothesis that as SOA increases, semantic relationships that are "farther

away" in the lexicon begin to take effect[2]. On the other hand, as noted above, Fischler (1977b) showed that priming in lexical decision appeared to be correlated with semantic relatedness rather more than associative relatedness. And finally, there is the Schreuder et al. (1984) result that conceptual similarity, which is probably most related to what Fischler called semantic relatedness, does not prime in naming. How can these data be reconciled?

First of all, recall our hypothesis that association priming is due to a production system effect, since the naming task engages the production system, and association norms are gathered using word association tests. Then the fact that pure semantic relatedness without association does not affect a naming task appears to be consistent[3]. One caveat to this is that as the duration of the prime or SOA increases, we would expect to find more effects of semantic relationships, since the there would be more time for activation to spread "up" to the higher levels of the system, and feed back "down" to the recognition level. That there are effects of the type of semantic relationship within the association effect is not surprising, given that such relationships have an effect on the lexical decision task. Especially so, since the time course of these effects are what is affected. That is, the association effect in naming is increased by the engagement of the speech production system, but then the time course of the effect is mediated by the semantic organization once the semantic pathways between prime and target have been opened.

The result that perceptual similarity primes in both naming and lexical decision implies that the locus of this effect is at the lexical recognition level, since this level has to be passed through for lexical decisions, so that the effect is simply additive to whatever higher level effects are present. This effect is

[2]Stanovich & West (1983) provide further support for this model. They found that sentential context has effects on naming at long SOA's.

[3]Actually, Seidenberg et al. (1984) found effects of semantic relatedness in naming. However, their study did not control for perceptual relatedness, shown by Schreuder et al. (1984) to affect naming latencies. Schreuder et al. did control for perceptual relatedness and found no effects of semantic relatedness in naming.

most likely due to a "dual encoding" (Pavio, 1971) of concepts in both a visual and more linguistically oriented code. This is an automatic effect, and so it is not surprising that it is additive to the conceptual relatedness effects. However, there may be effects that are "automatic", but only enabled when the system is in "production mode". At least, the association effect may be facilitated by the subject's use of the speech production system. If so, any effect found in naming and not in lexical decision should be attributable to the production system. Therefore one would expect to find association effects independent of semantic effects. That is, if we could factor out associative relatedness from semantic relatedness, we would expect no effects of associative relatedness in lexical decision, because the production system is not engaged. Unfortunately, no one has come forth with a methodology capable of this.

However, a recent study has shown a naming-specific effect that supports our hypothesis. Seidenberg, Waters, Barnes & Tanenhaus (1984) found that when the spelling of a word did not match the usual pronunciation (as in "done"; most words spelled like this are pronounced with a long "o"), it took longer to pronounce than a word of the same frequency with a regular spelling-sound correspondence (e.g., "bone"). However, in a lexical decision task, there were no effects of spelling-sound regularity. This supports our hypothesis, since it is the phonemic level that is implicated here; the "wrong" phonemes have been activated by the spelling, and they interfere with the task of pronouncing the word. One would not expect such interference in a lexical decision task, since the task of producing the proper phoneme is not involved.

On the other hand, it is not surprising to find lexical decision effects without corresponding naming effects. Indeed, these are what lead to the hypothesis that a later processing level is reflected by this task. Inspection of Table 4.1 shows this to be the case.

Finally, we note that Neely (1977) found that when superordinates are used as primes in a lexical decision task, there appear to be independent

contributions of automatic spreading activation and attention. This is consistent with the idea that the level of meaning organization is involved in lexical decision, and that attention is necessary for decision.

An interesting study relating to the range of activation spread was done by DeGroot (1984). She found that in a lexical decision task, priming between associates (which were also presumably semantically related) was not transitive. That is, if two words are associatively related, such as BULL and COW, and COW is associatively related to MILK, then BULL does not prime MILK. That is, COW is not a "pathway" for priming between its associates. This is evidence against the Collins & Loftus model, which assumes no barriers to spreading activation. We will present a model later which is in accord with DeGroot's data.

### 4.2.2. Case Grammar

The representation we will use for the "meaning" of the sentence will consist of nodes representing the meanings of the words, nodes representing semantic roles in the sentence, and nodes representing assignments of the meanings to the roles. The roles we use are called *cases*, first proposed by Fillmore (1968). Although not evident in this early work, later interpreters (Cook, 1979) proposed that the major tenet of case grammar is that *semantics* is the central component of language analysis, and that the case structure of the sentence *is* the "deep structure" underlying the sentence. Given that this is the basis for what we will term "semantic interpretation", we will give a brief introduction to case grammar here. Our model will not reflect all of the aspects of case grammar that we will touch upon, however, we plan to rectify this situation in future work.

Although this discussion will be based mainly on Cook (1979), we follow Bruce (1975) in defining a case as a binary relation which holds between a predicate (in the linguistic sense; usually a verb) and its argument (a semantic role associated with the verb). For example, in the sentence "John hit Jack",

"hit" is the predicate with two cases showing in this sentence:

Agent(hit, John)

Object(hit, Jack)

The Agent case represents the person or thing performing an action, the Object case is an obligatory case found with every verb, representing the thing being acted upon, or the thing in the state described by the verb, or the thing that is changed by the process described by the verb. Other cases listed by Cook (1979) include Beneficiary (the person or entity that benefits from an action) Experiencer (the person experiencing an emotion, sensation, etc.) and Location (the physical location of something). These are the so-called "inner cases" that are required by the verb, and make up its *case frame.* The combination of the verb, these cases and their fillers results in a propositional structure which represents the meaning of the sentence, independent of tense, aspect and negation, which are seen as higher order predicates applying to this structure. What are called "outer" or "modal" cases, such as Instrument, Cause, Time, etc. are cases of this higher order, surrounding-system. There are obvious parallels between these outer cases and the sort of information represented in frames (Minsky, 1975).

Cook sorts verbs into three types, State, Process, and Action. All verbs take an Object case. Action verbs additionally take an Agent case. Cook asserts that the Experiencer, Benefactive and Locative cases are mutually exclusive. This leads to a 3 by 4 matrix of verb types shown in Table 4.2. Evidence of the usefulness and completeness of this case grammar matrix derives from Cook's having used it to assign a deep structure to 5,000 clauses **in Hemmingway's *The Old Man and the Sea.***

Another aspect of case grammar is a set of *realization rules* for mapping the deep structure to a surface structure. Part of this mapping is reflected in the left-to-right ordering of cases in the case frame. This specifies the subject choice hierarchy. The first case is usually the Subject, [f for some reason

Table 4.2 Cook's Matrix of Verb Types
(from (Cook, 1979), p. 203)

| Verb Types | Basic Verbs | Experiential | Benefactive | Locative |
|---|---|---|---|---|
| State | Os<br>be tall | E,Os<br>like | B,Os<br>have | Os,L<br>be in |
| **Process** | 0<br>die | E,0<br>enjoy | B,0<br>acquire | O,L<br>move, iv. |
| Action | A,0<br>kill | A,E,0<br>say | A,B,0<br>give | A,O,L<br>put |

(such as passivization), a case is missing or moved in the surface structure, the next case in the list is moved into the Subject position. For example, in *give* **[_A,B,O], we go from *Kathy gave JellyBean a biscuit,* to *JellyBean was given a** *biscuit* to *A biscuit was given.* These same rules are applied in reverse by semantic interpretation programs (Bruce, 1975; Hirst, 1984) to assign case structure to sentences. Besides being marked by being in "top level" roles such as Subject or Direct Object, cases are often flagged by prepositions. For example, in *JellyBean put the puppy in its place,* the Locative case is marked by $^{M}$in$^{M}$. This is the sort of information that must be captured by realization rules (and, by parsing rules). Often this kind of flagging is verb dependent, and must be represented somehow with the verb. For example, even though *give* and *bribe* are both Benefactive Action verbs, the Object is flagged by being the Direct Object of *give.* The Object of *bribe* (usually money) is covert, (it is *lexicalized* in *bribe),* but it may appear in the surface structure overtly; if it does, it is flagged by *with.*

Also, verbs impose selectional restrictions on the fillers for their cases. For example, the filler for the Object case of *eat* must be marked 4-[food]. Semantic interpreters that use cases often include this type of information in

the lexicon with the verb. While complicating the definitions, this type of information often aids in the disambiguation of the sense of the verb, and the sense of the nouns[4]. For example, in *Gary wrote his first draft*, because the Object of *write* must have the feature +[text], it is possible to select the proper meaning of *draft*.

In this work, we will use a combination of selectional restrictions and cases to help disambiguate word senses. Although there are "hooks" for this in the model, we will ignore, for the moment, Cook's observation that verbs can be divided into different classes that differentially specify flags for their cases. However, it should be clear that case grammar is a viable formalism for a semantic interpretation system, and appears well suited for the purposes of this work. We now proceed to the design of the system.

### 4.3. A System for Semantic Interpretation

We will begin by presenting a model of priming which accounts for true "semantic" priming, i.e., priming through the conceptual similarity relation. We will not try to account for associative, phonemic or perceptual priming, although our model can be extended to include these. We will then proceed to our model of semantic interpretation which includes semantic priming as a source for meaning disambiguation.

#### 4.3.1. A Priming Mechanism

We will describe a priming mechanism which is consistent with much of the data. Basically, it is a review of some of the discussion in chapter 3. That is, we assume that non-identity priming passes through some intermediate nodes. We assume that for every relationship between two words or concepts, there is a superordinate node that encodes that relationship and links the two (except in the case where one of the concepts *is* the superordinate concept). We also assume that weights of subordinate-superordinate links (i.e., bottom-

[4]See Hirst (1984) for a good example of the use of this type of information in semantic interpretation.

up links) are such that if the subordinate node is firing, it causes the superordinate node to fire. On the other hand, we assume that superordinate-subordinate links (i.e., top-down links) are of a lower weight, and that thresholds on units are such that if the superordinate node is firing, the subordinate is primed to near threshold, but does not fire unless it also has bottom-up input.

There are several things to note about this model (see Figure 4.5). fn the following discussion, assume that A fires from bottom-up (perceptual) input. Then this will cause B, C and D to fire in turn, as activation spreads up bottom-up links. So one prediction is that superordinate categories are automatically accessed, without further input. It is assumed that this will facilitate further processing of superordinate categories. It is not as if the superordinate categories themselves were recognized (as in name identity priming) since there is necessarily a time delay before they get activated. Furthermore, E, F and G will merely be primed, and not fire. So, consistent

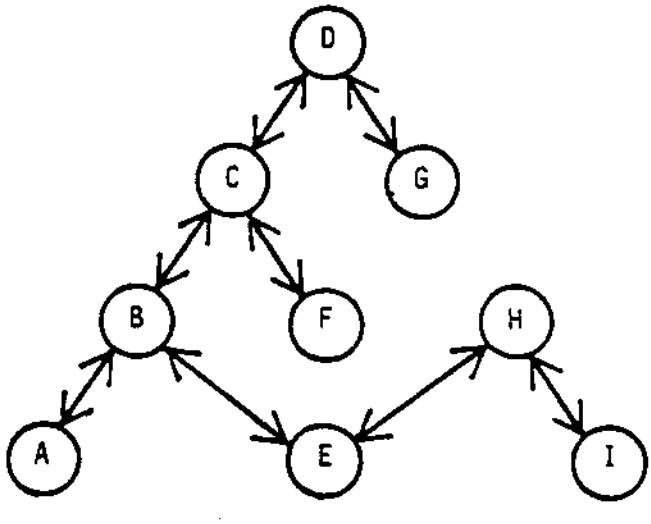



Figure 4.5. A model of priming due to hierarchical relationships.

with the data, coordinates will be primed, and we predict that higher level "aunts" and "uncles" will also be primed. We have not made any assumptions so far about the activation decreasing as it spreads. For the current model, in an attempt to make falsifiable predictions, let us assume it does not until proven otherwise. If it did, we would expect that the higher level relatives would be primed less. As it is, we only predict that there will be a delay in priming the farther away a higher level superordinate or relative is.

Finally, H will not be affected at all, (since E is not firing) so that I will not be primed. Thus this agrees with the DeGroot (1983) results that priming is not transitive. To see this in more detail, since (as in Chapter 3) we are assuming that the superordinate sites compute the maximum of their inputs, subsequent input of I would cause H to fire, but this would not increase the activation of E, since the maximum of Fs top-down input would be the same (modulo decay of B and E and weights on links). Therefore, E would not fire, and H would not get any more activation than if A had not been activated at all. *

Alternatively, assume that C is activated by bottom-up input (i.e. C is a category, and the prime is the category name). Now, D (the superordinate) will get activated, and G (the coordinate) will be primed Also, consistent with the Neely (1977) results, B and F will be primed Response to A and E will be enhanced due to the fact that B is primed, as follows: If A is recognized, B will fire faster due to being primed. Thus A will get feedback sooner from B. Note that A must be recognized (we identify firing with recognition) before this effect takes hold.

With this priming mechanism in hand, we are ready to proceed to the description of the semantic interpretation model.

### 4.3.2. Overview of the System

Recall that by "semantic interpretation", we mean the assignment of case roles to conceptual objects specified by the noun phrases of the sentence. In

particular, we are only interested in dealing with some fairly simple sentences from the Seidenberg et al. (1982) study. We have implemented a preliminary version of this system (reported in Cottrell & Small, 1983) and designed an improved version, which is the major topic of this section. The operation of the model is as follows. Major lexical items are activated in sequence (determiners are currently ignored). After a settling period, the result of the network's operation is a stable coalition of units representing the case structure of the sentence, where the only units remaining active are: (a) units representing the appropriate word senses for the sentence (disambiguation); (b) units representing the appropriate cases for the selected verb sense (case frame selection); and (c) units representing assignments of the word senses to the cases. This is the first system (that we know about) that incorporates completely distributed, parallel processing of sentences in a manner consistent with (and potentially falsifiable by) psycholinguistic, neurolinguistic and anatomical data. Our focus is cognitive modelling in this relatively new framework, and we do not solve any problems that are still the subject of current research in more traditional paradigms. We will not describe a model that interprets quantificational scope, resolves anaphora, or handles any of the more difficult issues in semantic interpretation.

A little history and overview will set the stage for discussion of the current model. Perhaps by seeing where we started, the reader will see how we got here. The preliminary version of the semantic interpreter reported in (Cottrell & Small 1983) consists of three levels of units:

(1) The Lexical Level. This is the "input" level. There is a unit for every word in the language.

(2) The Word Sense Level. This has a unit for every meaning of a word, with units at the lexical level connected to all of their "meanings" at this level. Alternate meanings of a word are mutually inhibitory.

(3) The Case Level. This has units for every possible relationship between the predicates and objects. We posited an "exploded case" representation; that is, on the order of several hundred case roles that are more specific than Agent, Object, etc., but fail into those classes (see Fahlman, 1979). These nodes as a result are connected to fewer word senses than Agent and Object would be, and carry much more information directly. Units at the word sense level representing fillers for cases are connected to those cases; verb senses are connected to their case frame. The case units are set up so that they need both filler and verb to fire; otherwise they are strongly primed. A syntactic processor was posited at this level on an equal footing with the case network, but the idea was not expanded beyond this in (Cottrell & Small, 1983).

The operation of the model consists of a flow of activation from the lexical items (introduced in sequence) to their meanings. The meaning nodes in turn, activate the case nodes. The relation that best fits the input will then "win." Winning involves the formation of a stable coalition, that is, a group of connected nodes in which the overall excitation exceeds the overall inhibition. The model can be said to have succeeded if the proper case roles form a coalition with the right meanings for the sentence. Since many words are ambiguous, the network must decide on an interpretation based on word sense frequency and relational knowledge expressed at the case level. Thus in a sentence such as "Bob threw the fight", the sense of "threw" is disambiguated by "fight", since "fight" only fills the SPORTING-EVENT case of the "intentionally lose" meaning of "threw", and not, for example, the MOVEABLE-OBJECT case of the "propel" sense of "threw". See Figure 4.6.

This model had its problems. Since the syntactic module was not yet implemented, only sentences which were not semanticaliy reversible could be interpreted. This was seen as a feature rather than a bug; if one considered the model to be a "lesioning" of the overall model, with access to syntactic information removed, then it appeared compatible with results about Broca's

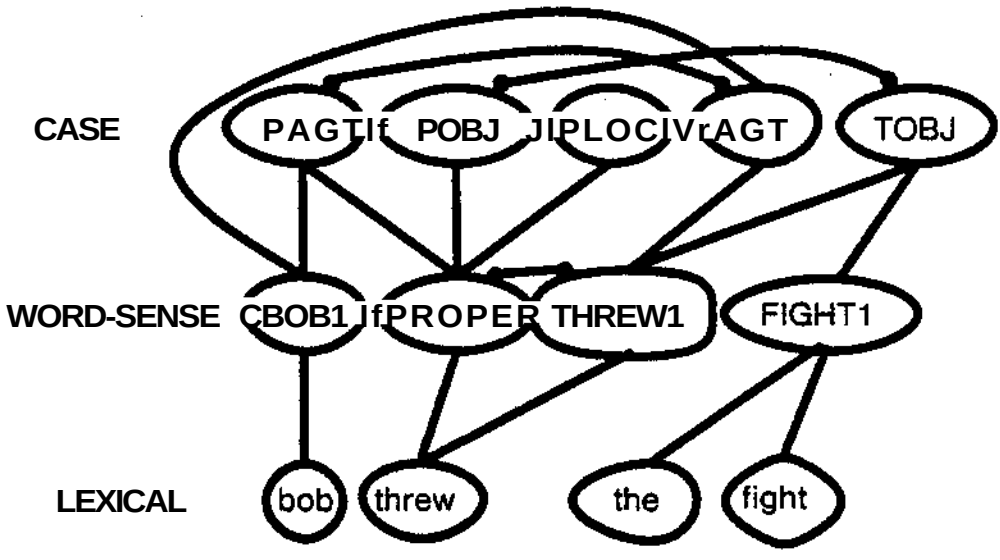



Figure 4.6. Subset of the network for "bob threw the fight."

aphasics. The problem, however, is deeper than that. In a semantically reversible sentence, such as "John loves Mary", no assignment of the Agent and Object cases, even a random one, was possible. The reason is that the assignment of a word sense to a case was implicit in the combination of 1) the two of them being connected and 2) the two of them being active in a stable coalition. In the case of "John loves Mary", both (the meanings of) "John" and "Mary" would be connected to LOVE-Agent and LOVE-Object. After activating "John" "loves" and "Mary", all of these nodes would be active, resulting in both of them being "assigned" to both cases.

A second problem with this design is that there is no obvious way to interface syntax with the case assignment mechanism. In order to implement the Passive transformation, for example, the syntactic information would have to operate on the connections themselves (an acceptable alternative in connectionist networks), disabling some and enabling others. The problem here is that there is no obvious way to make this a general transformation. It

would have to operate on every connection to every exploded Agent and Object case for verbs subject to the Passive transformation. This is totally implausible for an operation that is generalized by language learners very quickly.

Finally, there is not enough underlying structure to this model. There are no generalizations between related verbs, related nouns, or related cases. The units are only connected through the case structure, (related nouns would fill the same cases) which is inadequate for modelling many types of priming.

The current model was designed to overcome these problems (see Figure 4.7). The lexical level activates word sense representations in a word sense buffer, through the lexical access network described in the previous chapter. The buffered word senses achieve their "sense" through connections into a lexicon, where information about the sense is stored. The lexicon, for our purposes, is just an inheritance hierarchy of concepts, with connections to exploded cases at appropriate points. Thus, in this system, the fact that a word sense can fill a case can be inherited from a superordinate in the lexicon.

The cases themselves are arranged in hierarchies, one for each of the more abstract Fillmorean cases, with Agent, Object, etc. at the roots. Thus the fact that a MOVEABLE-Object is an Object in Fillmore's sense is inherited through this hierarchy. Verb word senses in the buffer are connected directly to their appropriate case frames in the case hierarchies. The intersection of activation from a filler and a verb is still the mechanism which activates a case node; the difference is that the activation from the filler arrives indirectly through the lexicon. In the previous model, all cases of the same general type were mutually inhibitory. Here, competition between different candidates for the Object case, for example, happens as competition between coordinates in the case hierarchy, reducing by several orders of magnitude the number of inhibitory connections necessary. The result of this disambiguation process is that a *path* to the root of the hierarchy "wins" (as in the lexical access model).

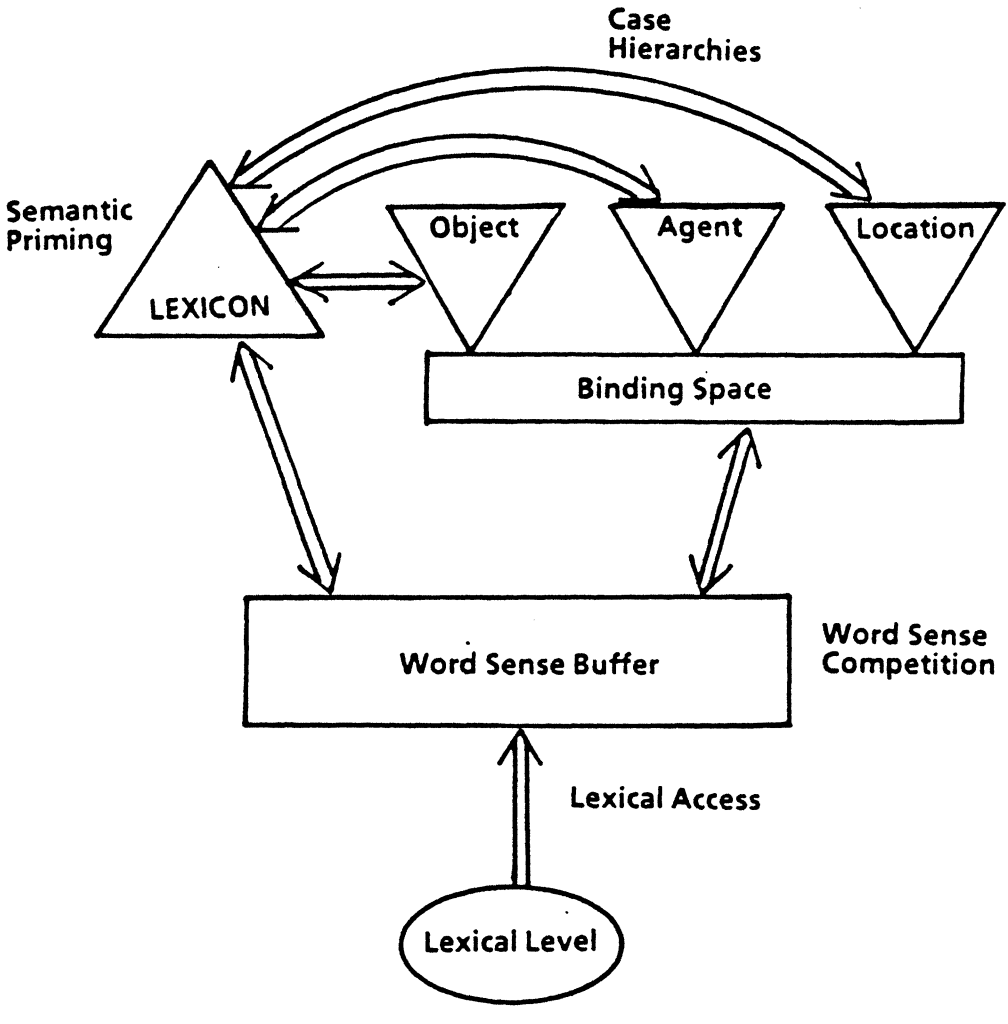



Figure 4.7. Overview of the case system.

Bindings between word senses and cases are represented by nodes which encode the assignment of a buffer position to a case; these binding nodes are mutually inhibitory, so that, for example, only one of "John" or "Mary" gets assigned to the Agent case. This replaces the direct connections of the previous model. Finally, these bindings are to the roots of the case hierarchies, rather than the exploded cases themselves, greatly reducing the number of such

nodes. This provides a clean place for the interface with syntax.

In the previous model, the result of the interpretation was a stable coalition of nodes representing case-filling word senses, a verb word sense, and the case frame for that verb sense. The new model includes two paths between case fillers and cases, rather than direct connections: A path through the lexicon, and a path through the root of the case hierarchy and the binding node network where assignments are decided.

This completes the overview of the model. We now move on to a more detailed description. We begin with a model of the semantic network organization of the lexicon, as posited by Forster (1979) and others, based on our model of semantic priming above. Then we describe how cases are represented, and finally how they become assigned to conceptual objects. In the course of this we describe how word senses are disambiguated. It is perhaps important to point out that the complete model described here has not been implemented, although portions of it have in other contexts. A role assignment mechanism similar to the one described here (called the *binding* mechanism) is used in the implementation of the syntactic processor described in the following chapter, and foundational work on inheritance hierarchies (necessary for the semantic network) is described in Chapter 6. However, we do go through an example of the operation of the system "by hand", and the last part of this section describes example runs from the implementation of the earlier version of the model.

### 433. A Model of the Lexicon

We assume first of all, that the representation of word senses that are activated by our lexical access mechanism described in Chapter 3 are buffered. A mechanism for this is described in Chapter 5. By virtue of their buffer position, word senses are thus tagged with a unique identifier which will be important later, when we discuss how they become bound to the case they fill in the sentence. This means that what we labeled, for example, SHIP-FLOOR,

in Chapter 3, actually is a unit labeled SHIP-FLOOR/CONC3, where CONC stands for "concept" and "3" means this is in the third buffer position. Other meanings for the same word exist in the same buffer position, so CARD-DECK is labeled CARD-DECK/CONC3 as well.

These labeled word-senses are then linked (two-way) to a concept node in a semantic network representing that meaning. Thus words are, in one sense, simply pointers into this network through their various senses. Figure 4.8 shows a pair of buffer locations with the some of the nodes corresponding to the senses of "Tom" and "threw" in positions 1 and 2, and their links into the lexicon. (From now on, when we refer to "the lexicon" or "the semantic network", we will be referring to the same entity.) For our present purposes, we can just assume that the organization of this network is just the usual IS-A hierarchy as outlined in (Fahlman, 1979) and elsewhere. The details of the representation of knowledge in this hierarchy as shown in Figure 4.8 are not our major concern. What is important is that the actual implementation follow the assumptions given in the previous section, i.e., links between subordinates and superordinates are of two kinds: *bottom-up* links which are weighted so that the subordinate firing causes the superordinate to fire, and *top-down* links which are weighted so that if the superordinate is firing, then the subordinate will be primed to just below threshold. Additionally, the potential function of each unit is assumed to be such that if it is firing, top-down feedback can increase its output incrementally depending on the amount of top-down feedback. Conversely, we have to assume that once a unit is active from bottom-up input, increases in that input do not increase its output, since this would cause a "vicious cycle" of subordinates and superordinates increasing one another's activation until they saturated. Thus, bottom-up input activates a unit, but it is top-down feedback that makes discernible changes in its output[5]. Thus, all we require is that once a unit activates its superordinates, if one of

[5]We also assume that *decreases* in bottom-up input cause a unit to decrease its output, and if the bottom-up input stops, the unit will decay back to resting state.

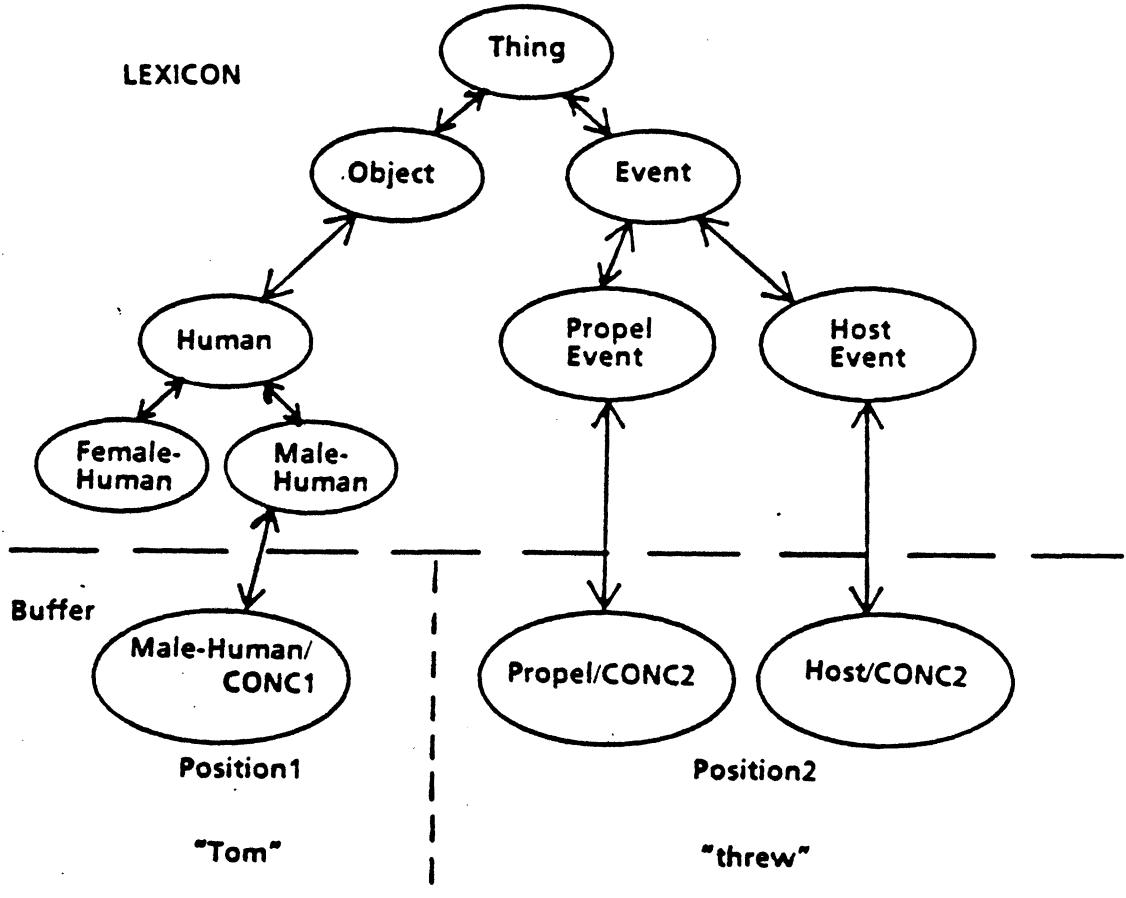



Figure 4.8. Two buffer locations with nodes for "Tom" and "threw", and their links into the lexicon.

those superordinates begins to get some feedback from outside the hierarchy, it will be communicated down to the subordinate, whose activation will then increase.

Referring to Figure 4.8, these rules mean that if HUMAN gets feedback from the case hierarchy (see below), increases in its activation will be communicated downward to MALE-HUMAN and thence to MALE-

HUMAN/CONC1. FEMALE-HUMAN will simply remain primed unless a buffer node below it becomes active[6]. With these additional assumptions about activation functions, the model of semantic priming as outlined in the previous section constitutes the model of the lexicon.

In addition to this we assume that links from conceptual nodes in the semantic hierarchy connect to nodes representing possible cases they can fill (as described in the next section), and that these are two-way links, so that the case nodes can provide feedback through the top-down links in the hierarchy to the nodes in the buffer. The next subsection describes the organization of these case nodes.

### 4.3.4. Cases as Cognitively Real Objects

This section develops the basic model of case relations, and its interface with the lexicon. Identifying the case relations between the semantic objects of the sentence and the predicate of the sentence constitutes (for the purposes of this thesis) the semantic interpretation of sentences. Case relations and the semantic network overlaying the lexicon are adequate for explaining how word senses are semantically disambiguated in the STLB materials. In particular, we are not attacking the problem of sentences with more than one clause, where several case frames have to be related in the final semantic representation. We leave this to future research.

As noted in the section on Case Grammar, cases represent possible relations between the predicate of the sentence and the noun phrases. General cases such as Agent and Object, however, are not very useful as an aid to disambiguation. For example, in the sentence "Bob threw the fight," the fact that "fight" fills the Object case does not help determine what sense of "threw" is appropriate for this sentence. However, the use of more specific cases,

[6]The careful reader may be wondering how buffer nodes become attached to the semantic hierarchy. We are assuming that everything pre-exists in the buffer. The numbers of units implied by this implausible assumption can be somewhat ameliorated by coarse-coding methods (Feldman & Ballard, 1982), or by using a distributed representation of the buffer elements. This is discussed a bit more in the concluding chapter.

tailored to the sense of the verb, can reduce the problem considerably. The idea is to use cases that subcategorize the abstract cases used by Fillmore with selectional restrictions imposed by the verb sense. Suppose with the sense of "threw" we call "INTENTIONALLY-LOSE" (which felicitously abbreviates as I-LOSE), we have a particular kind of Object case, the "I-LObject" case, that represents a type of game. Then "fight" would fill this case role, but not the case role "PROPEL-Object" associated with the "PROPEL" sense of "threw". From this we can determine what sense of "threw" is intended in this sentence, since it is the only sense that has its (obligatory) Object case filled. We term this highly typed case system an *exploded case* representation. Given that we have a large number of nodes to work with in a connectionist system, it is feasible to use a large number of specific relations. This permits the encoding of more specific information in the cases and thus more constraints on the role fillers.

Verb senses in the buffer are associated with their case frames by simply being connected to the case nodes that constitute their case frame (see Figure 4.9)[7]. The potential function of the verb word senses and the weights from the case nodes are set up so that the obligatory cases must be filled in order for the verb sense to get enough feedback to remain firing and survive competition with the other senses. This is the basic disambiguation mechanism for verb senses. Verb senses with more cases filled win over verb senses with fewer cases filled. Verb senses with obligatory cases filled win over verb senses without their obligatory cases filled.

A case node is "filled" when it has input both from a predicate in the word sense buffer and a filler in the semantic network. When it gets input from both, it begins feeding back to its verb through direct connection and to the filler via the semantic network (see Figure 4.10). This is the basic

[7] A more consistent approach would have the verb representations in the lexicon mediating between the verb senses in the buffer and their cases in the case hierarchies. We use the more direct approach given in the text to simplify the presentation.

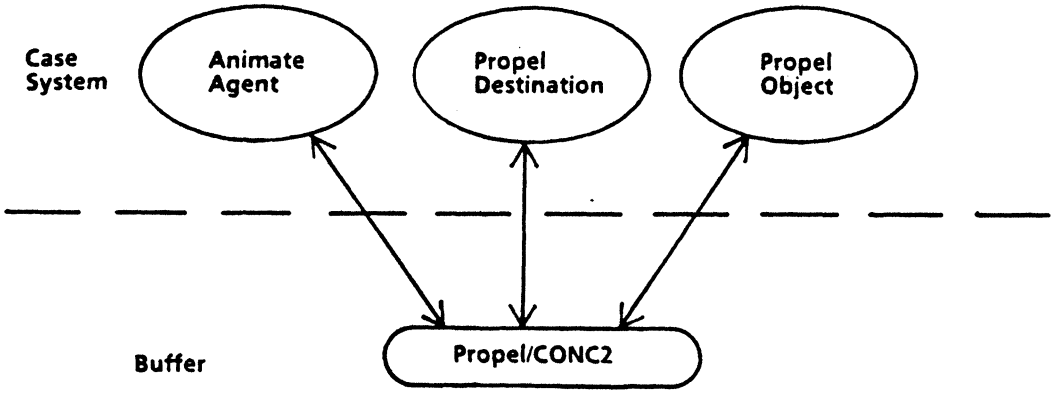



Figure 4.9. Verb connections from the buffer to their case frames.

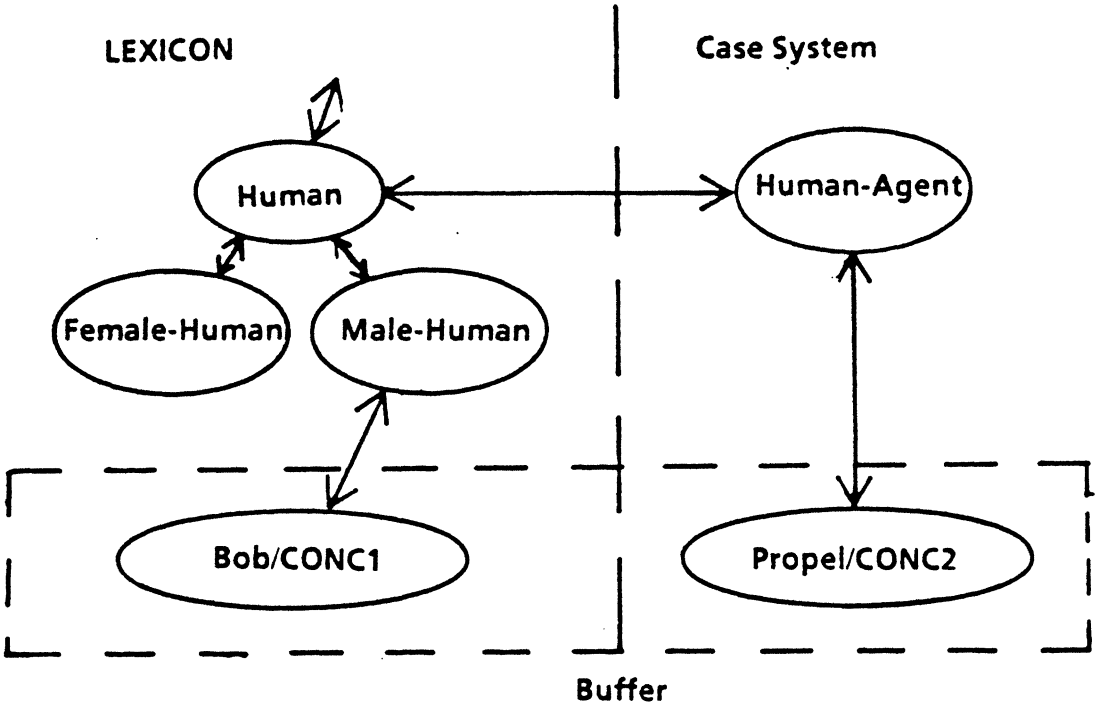



Figure 4.10. Feedback path from a case to its filler.

disambiguation mechanism for case-filling concepts: feedback from the case role via the pathway through the lexicon. HUMAN-Agent feeds back to

HUMAN in the lexicon, which then provides more feedback to MALE-HUMAN and thus MALE-HUMAN/CONC1. This is thus a mechanism for cases to "vote" for their fillers. Noun senses that fill a case get more feedback than noun senses that don't. Notice that there is the possibility of crosstalk in this design. If a word later in the sentence has the meaning FEMALE-HUMAN, it will get feedback from the HUMAN node. This implies that in sentences such as, *Mary went to the* **john**, the MALE-HUMAN meaning of **john** will initially get more feedback than the other meaning. This is a prediction of the model design, and may turn out upon implementation to be undesirable. An alternative would have been to make the feedback from HUMAN in Figure 4.10 go through a WTA so that only the subordinate that caused the activation of HUMAN would get the feedback. Then this WTA could be released upon the binding of MALE-HUMAN/CONC1 to a case role, allowing other subordinates to get feedback. These factors will have to be explored further when the system is implemented.

As it stands, the design leaves exploded cases "off by themselves", with no inherent organization. There are important regularities between exploded cases that are not captured this way and inefficiencies in the number of connections from the semantic network. Many verbs have exactly the same restrictions on the Agent, and so should share this node with each other. Since the cases are strongly typed, one exploded case can be an instance of another kind. All Agent cases are instances of Fillmore's Agent case, for example, and some verbs are not going to place any restrictions on their Agents. It would be inefficient to connect every node in the lexicon that could be an Agent to the most general Agent node. An more efficient method is to arrange the cases in several hierarchies based on the selectional restrictions on the cases, with the most abstract, Fillmorean cases at the roots. Figure 4.11 shows an example arrangement of the Object case hierarchy with verbs attached at appropriate points. There is a tantalizing similarity to the semantic hierarchy used for the lexicon; perhaps in future work we can merge them. They are at least logically

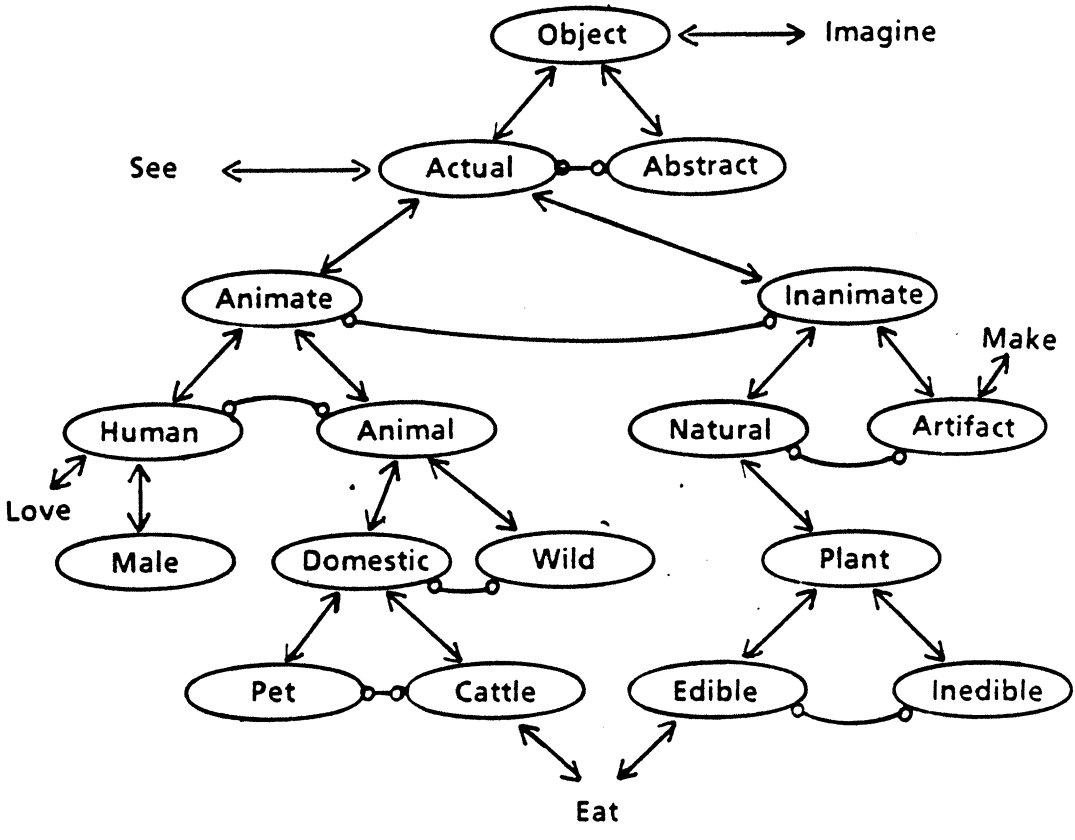



Figure 4.11. The Object Case hierarchy.

separate; they have different control characteristics. Case nodes require dual input from a predicate and a filler in order to feed back to the filler; ordinary

concept nodes only require bottom-up input are mutually inhibitory. Not so in the lexicon, where we want them to facilitate one another.

Figure 4.11 also shows another interesting design decision: certain verbs are connected at more than one entry point to the hierarchy. For example, EAT is connected to EDIBLE (PLANT) as well as CATTLE This is a radical departure from most case theories, forced on us by our trying to arrange cases hierarchically[8]. Since we don't restrict ourselves to trees, it is tempting to create a new superordinate every time some "case" does not fit into the structure, ft is more flexible to simply allow disjoint sets of selectional restrictions, connecting the verb to the most general minimal contrast coordinate[9] for a particular subset of restrictions. For example, to connect EAT to the one node that includes everything one might eat, both cattle and edible plants, given the hierarchy in Figure 4.11, this would be ACTUAL (OBJECT). As discussed earlier, this is an overgeneralization that would be relatively useless as an aid to disambiguation. It may be an artifact of a particular arrangement of the hierarchy, but we would venture to guess .that any arrangement would lead to this situation for many verbs.

Now we can make use of this structure by connecting nodes in the lexicon to their most specific roles in the case hierarchy, and allowing activation to spread up to the more general cases, informing them that there is a filler about. In this arrangement however, we must insure that only one case of each type wins, so members of the same type are mutually inhibitory. All nodes that are immediate children of another node are thus mutually inhibitory. This guarantees that eventually one path to the root case (say, Object) will "win"

**[8] A good example of this is trying to handle the different types of Agency of "'threw" in "The tornado {FORCE) threw the house through the air" versus "Bob (AGENT) threw the bail." Some case systems might have two lexical entires for "threw" with different case frames to cover this. But both entries still "mean" PROPEL. Others would have the Agent case be general enough to cover both. Given the exploded case representation and the case hierarchies, allowing more than one type of the same general case appears to be a clean way to resolve the issue. We leave it to the linguists to decide if it is warranted.**

**[9] Minimai contrast coordinates are mutually exclusive subordinates of the same node.**

(note the similarity between this and our lexical access model; here, though, the discrimination network is inverted.) The linkages are exactly the same as for the semantic network hierarchy, except for mutual inhibition between subtypes of the same node; activation spreads "up" to the root, but not "down" past where there is activation from the lexicon. Thus, we still have the idea of Agent and Object, but these form the root nodes of hierarchies of more specific, typed cases.

**Details of the Case Hierarchy Semantics**

There is more to the design than has been presented, since more control information is necessary. For example, if a specific case in the Object hierarchy is activated by the semantic network, and this activation spreads up the Object hierarchy, what prevents everything in the semantic network connected to this path from getting feedback? Presumably, the root of the semantic network, THING, can fill the most general Object case, so how do we prevent OBJECT-Object (the root of the Object hierarchy) from feeding back to THING in the semantic network, thus feeding back to everything?

Second, in the first version we gave with unorganized exploded cases, the semantics of a case node firing was that both the predicate of that case and an appropriate filler were firing. Now, it seems, we need a case node to fire if it only has a filler, in order to inform more general cases that they have been filled. How do we prevent this from feeding back to the semantic network when there is no predicate for these cases?

The problem is that we have many more control problems (or information channels) here than we can handle with a single node. The answer is to replace the nodes of the case hierarchy with a small network encoding the desired semantics. Let us be clear about that semantics first, and then describe the control network that encodes it.

We call a case *filled* if it has input from the semantic network. A case is *filled+* if it is filled or if a subordinate case is filled+. We call a case *satisfied*

if it is filled+ and one of its associated (connected) predicates is active. A case is *satisfied+* if it is satisfied or a subordinate is satisfied+. Finally, we use *superordinate** to indicate the reflexive transitive closure of superordinate. Now, the proper behavior should be:

(1) A case that is filled and has a satisfied superordinate* should feed back to the semantic network.

This condition says that only if a case has an associated predicate firing *and* it is getting direct stimulation from the semantic network, should it feed back to the semantic network. Figure 4.12 shows the basic idea. The nodes above where the predicate is attached to the hierarchy are not allowed to feed back to the semantic network. If the filler is at or below where the predicate attaches,

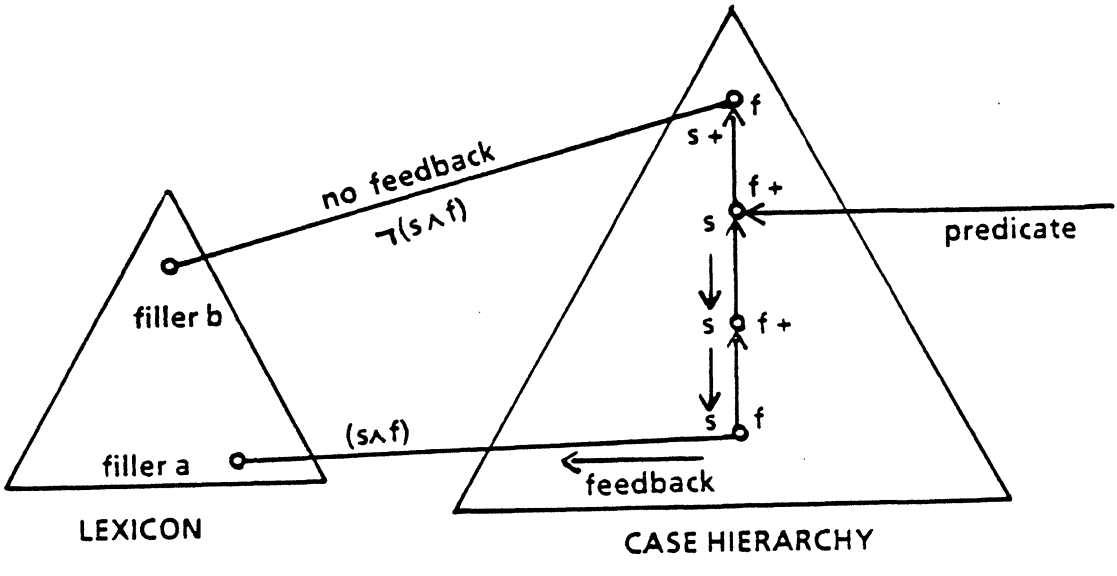



Figure 4.12. Filler "a" gets feedback, filler "b" doesn't (it is too general a filler for the predicate's case).

then all is well, and the case hierarchy feeds back to the semantic network where the filler is attached. If a predicate is below a filled case, nothing happens.

(2) After the network has converged on an interpretation of a sentence, each case hierarchy should have no more than one satisfied case, and thus only one satisfied + path to the root.

This condition says that when we are done, no more than one case in the hierarchy is associated with a predicate by being in the condition "satisfied." Notice that the existence of only one satisfied+ path to the root does not imply that there is only one satisfied case. There may be several satisfied cases on the path. This could only be due to the activity of more than one predicate linking into the path, since "satisfied" requires that a connected predicate is firing. It is the domain of the disambiguation mechanisms to reduce the number of predicates to one, should this occur, so all the machinery of the case hierarchy need do is provide "path disambiguation." Nor does the existence of only one satisfied case imply that there is only one filler for that case. There may be many fillers active at or below the satisfied case, since satisfied on requires that the case be filled+. Actually assigning a word sense to a case is the province of the binding mechanism to be discussed in the next section.

Implementing the semantics we want requires several information channels between cases in the hierarchy and between these and the semantic network. For example, a case that is filled must communicate this upwards, implementing the filled 4- relation. A case that is satisfied must communicate this downward, so that the filled case may then begin feeding back to the lexicon. For the purposes of the binding mechanism described in the next section, the satisfied + path to the root must be implemented, and this is distinct information from filled+, which necessarily also reaches the root. If we want to restrict ourselves to one unit per case node, one choice is to encode each type of information as one of the small number of output values, since a

connectionist unit has only one output. A cleaner solution is to apply the unit/value principle and use a different unit for each type of information to be communicated. Then the only problem is choosing connections and functions for the units to encode the control implementing the semantics. Thus each case "node" will now consist of several units reflecting different states of information about the case.

Figure 4.13 shows the network for a case role. It is necessary to separate "filled" from "filled+" and "satisfied" from "satisfied+" since in each pair, different semantics are required. For example, the feedback to the lexicon should only occur if the case is filled and satisfied, not just filled+ and

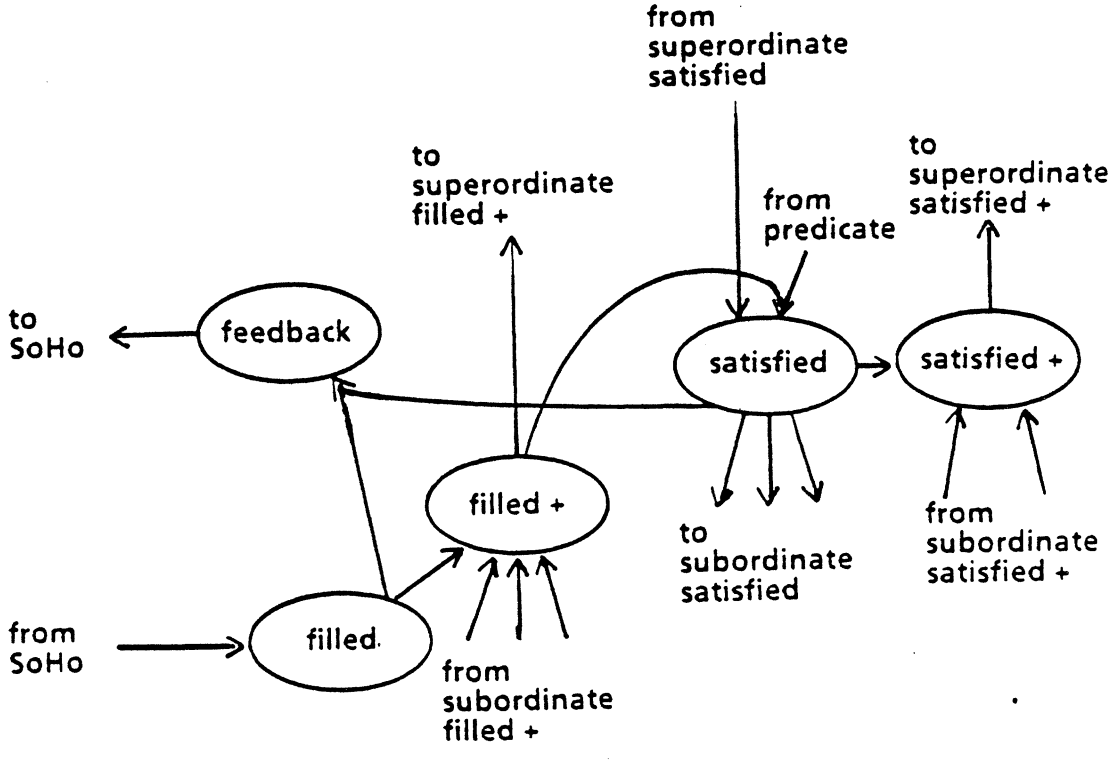



Figure 4.13. The control network for a case role.

satisfied. So we need separate units to encode filled and filled+. Also, we want to communicate upwards to the root that there is a satisfied case with the satisfied+ nodes. However, we need to communicate downward that there is a satisfied case, in case the filler is below (more specific), so that it is then enabled to feedback to the lexicon. Propositional schemas of the potential functions for these units are given in Table 4.3. The reader may check that these implement the semantics. Presumably we would use a numerical version of these that would reflect input strengths. Also we would need to have the satisfied+ nodes be mutually inhibitory between coordinates to implement the constraint that only one satisfied+ path to the root should "win" (this is not shown in the figure).

### 4.3.5. Bindings

A technical problem that now arises is the "binding problem," that is, how word senses become associated with the cases that they fill. The pathway through the lexicon is not sufficient for this, since the lexicon is shared, and many concepts have the same superordinate nodes, resulting in "crosstalk". (This is a "feature, not a bug", as explained in the Implications of the design

Table 4.3. Propositional Schemata for Case Network Units. C[i]'s are from lexicon, F+[j]'s are subordinate F+'s, S' is the superord. S, S+[j]'s are subordinate S+'s.

| Unit Name | Potential Function Schemata |
|---|---|
| F | OR(C[1],...C[n]) |
| F+ | OR(F,F+[1],..., F+[m]) |
| S | OR(AND(OR(P[1], ... P[k]), F+), S') |
| S+ | OR(S,S+[1],..., S+[m]) |
| Feed | AND(F,S) |

section). What we are aiming for here is a link between the two, corresponding to an *assignment* of a word sense to a case. This forms the basis of the stable coalition between the verb, its cases, and their associated fillers. We adopt essentially the simple (rather than the unit-efficient) solution given in (Feldman, 1982). That is, for every possible binding of a word sense in the buffer to a case, there is a unit connecting the two. We call such units *binder units*. To avoid the combinatorial explosion, we make this binding not to the exploded case, but to the root case of the hierarchy corresponding to the exploded case (see Figure 4.14). We can think of these binder units as corresponding to the assignment arrow in a programming language. In a

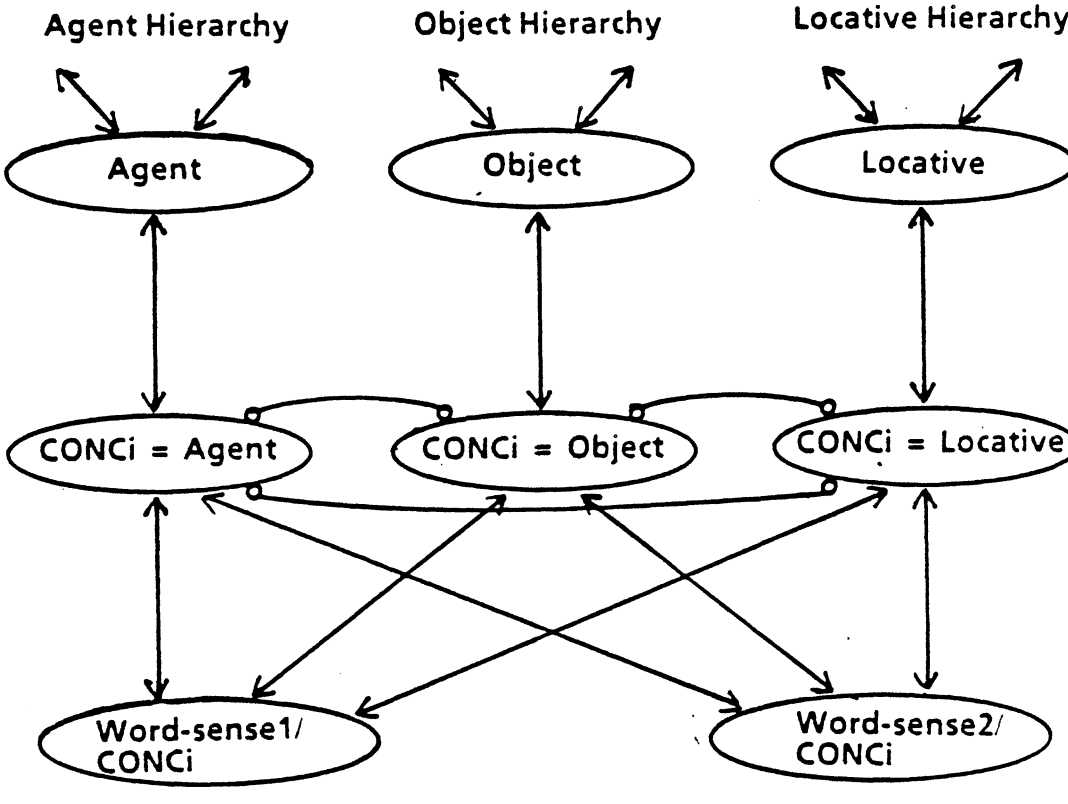



Figure 4.14. The binding space for CONCEPT1.

successful parse, then, the appropriate word sense for each buffer position is connected to the appropriate case through a binder unit at the root of the appropriate case hierarchy. For each position in the buffer, there is a set of binder units to the roots of all the case hierarchies. These are arranged in a WTA so that each buffer position is constrained to fill only one case. This is called the *binding space* for that buffer position. In addition, there are constraints between the binding spaces for the buffer positions. The Agent case should only be filled by one buffer position. This constraint is represented by making all of the binders to the Agent hierarchy a WTA. The result is a two dimensional WTA: One dimension represents the constraint that a position fill only one case; the other that a case be filled by one buffered word sense (see Figure 4.15). (An implementation of this is described in the next Chapter.) Thus there is a competition between the assignments of fillers to their cases.

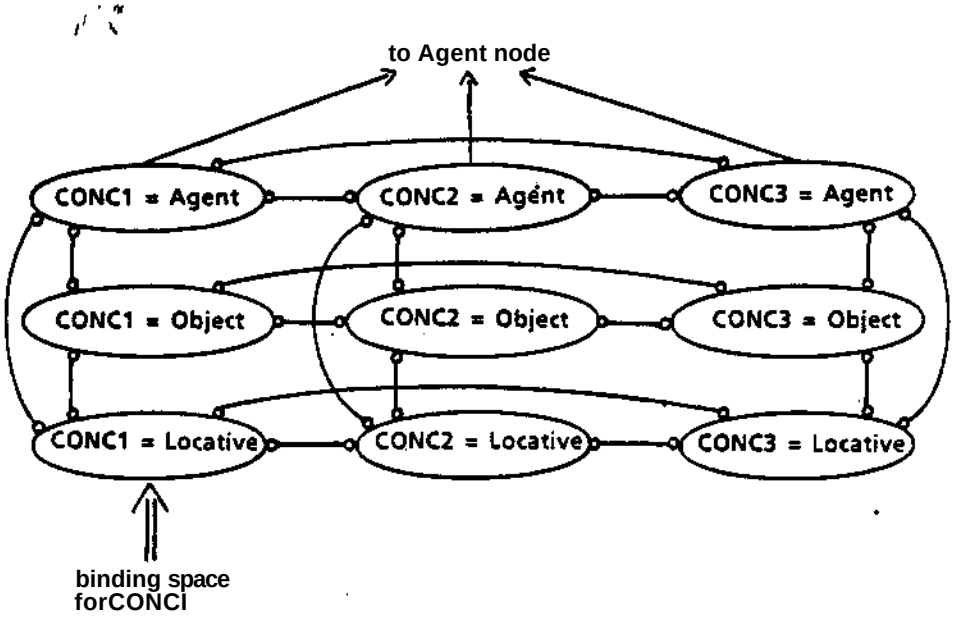



Figure 4.15. A two-dimensional binding space.

An interesting and useful consequence of this organization is that as word senses are assigned to cases through one of these binder units winning, the binding space for succeeding buffer positions are narrowed by these choices. If the word sense in buffer position 1 is assigned to be the Agent, then the binders to the Agent case in succeeding buffer spaces are suppressed, making the set of choices toward the end of the sentence fewer, and presumably speeding up the formation of the final coalition. This is similar to the building up of expectations in so called "expectation based" parsers, such as the Word Expert Parser (Small, 1980) or ELI (Riesbeck & Schank, 1976).

Another advantage of this organization is that it provides a clear place for the syntactic module to interface with the semantic module. Let us assume, for the sake of argument, that we can compute that NP1 is Concept1, thus associating the syntactic entity with the semantic entity. Our syntactic module uses binder units as well (see the next Chapter) to make assignments between syntactic constituents and their roles. Now, assume that it has computed that NP1=SUBJECT (that is, that binder is firing), and that the verb is PASSIVE. Then the connections shown in Figure 4.16 implement the Passive Transformation (Chomsky, 1965). They do this by inhibiting the assignment of the word sense in the first buffer position to the Agent case, without restricting the other possible cases it could fill. This is just one possibility for interaction between the two systems. Other syntactic and semantic relationships can be studied in this framework and it appears to be a fruitful approach for future work.

### 4.3.6. An Extended Example

As we have stated earlier, the full design we have just described has not been implemented. While running examples from a preliminary implementation are given in a later section, it is important at this point to work through an example "by hand", to see how the complete model works. In this section, we work through the processing of "John threw a rock." While this is

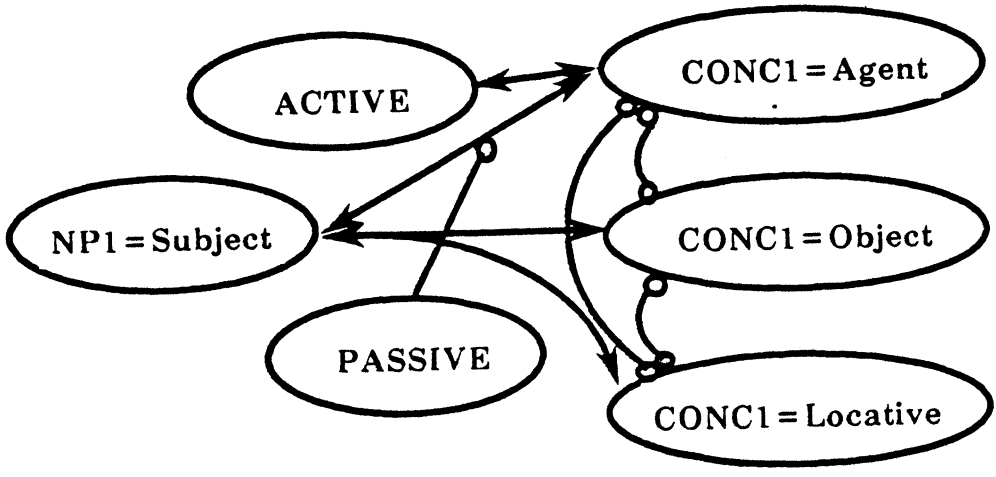



Figure 4.16. The Passive transformation.

not an exciting example, it proves complicated enough to do by hand. All of the content words need to be disambiguated. "John" could be a toilet or a person, "threw" could be PROPEL, HOST (in the sense of "threw a party"), or INTENTIONALLY-LOSE (in the sense of "threw the fight"), and "rock" could be a noun meaning "a piece of stone" or a verb meaning "cause to move back and forth"[10].

Also, for the kind of disambiguation phenomena we are trying to handle, an inheritance hierarchy, rather than a complete semantic network, is sufficient. One possible implementation of such a hierarchy is given in Chapter 6. We thus have to ignore some important representational issues in parsing. For example, our representation of an instance of a concept is to simply tag our words with "concept numbers", rather than to build an explicit frame in the semantic network. This is inadequate for the representation of complex concepts. It is even inadequate for simple things such as "a rock." In previous

[10]Recall that Seidenberg et al. (1982) showed that even in such situations, the verb meaning of "rock" is activated.

work, such as (Finin, 1980) or (Hirst, 1984), the word "a" is a frame specifier, signaling that an indefinite object frame is to be constructed. (Versus "the", which usually signals that an existing frame matching the concept is to be found) Such differences are likely to have an effect on sentence processing, and a mechanism for this should be included in a complete model. However, constructing a full-featured connectionist semantic network is beyond the scope of this thesis[11].

We assume the buffer and semantic network shown in Figure 4.17. We are employing several simplifications to keep the example manageable. The semantic network is stripped down to a few nodes of interest; we ignore the connections of the verb meanings to the lexicon, since they should not play a

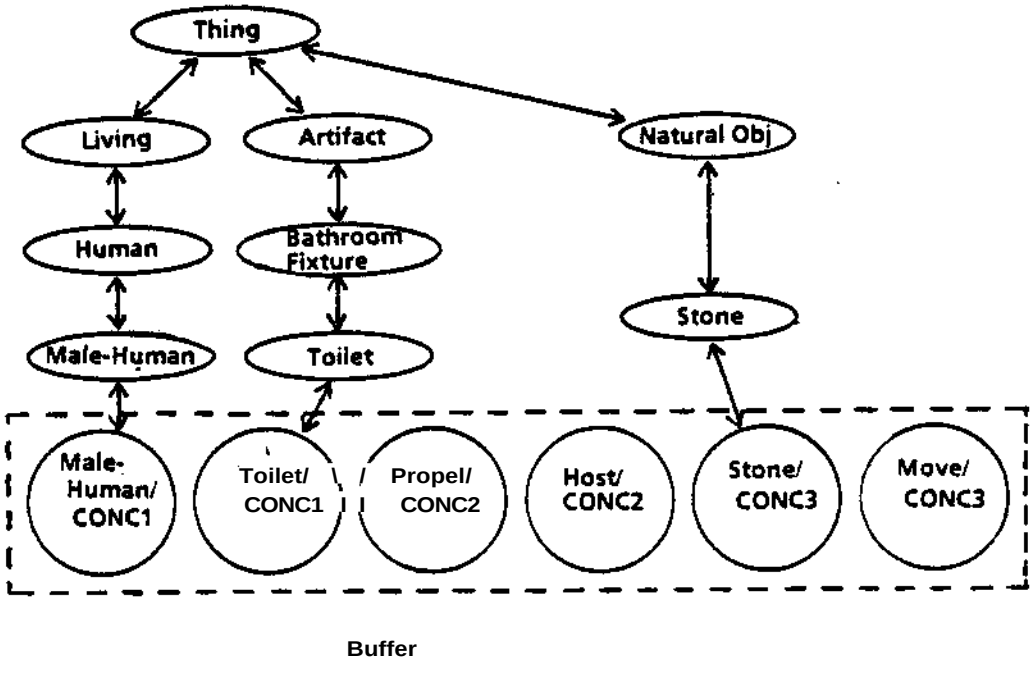



Figure 4.17. Buffer & lexicon for example sentence; verb connections omitted.

[11]However, see (Shastri & Feldman. 1984; Feldman & Shastri, L984) for a connectionist implementation of some of the features of a semantic network we would need.

major role in the example: and we only consider the PROPEL and HOST meanings of "threw". The meanings of "John" are represented as MALE-HUMAN (abbreviated M-H) and TOILET, and "rock"'s as STONE and MOVE. Since we don't have conceptual frames in our semantic network, we ignore the "a" in the example sentence, "John threw a rock." Finally, in the figures we just use one node per case in the case hierarchy, keeping in mind that this stands for the case network described above, with the attendant behavior. We simulate hearing the sentence by having the buffer nodes corresponding to the meaning instances of "John", "threw", and "rock" begin to fire at simulation steps 1, 6, and 11 respectively.

Figure 4.18 shows the Agent hierarchy with the connections to the verb meanings and the semantic network. All of the verb meanings share the ANIM(ate)-Agent case. We include FORCE as an Agent case, shared by MOVE and PROPEL. As we saw before, connecting the verbs to their most

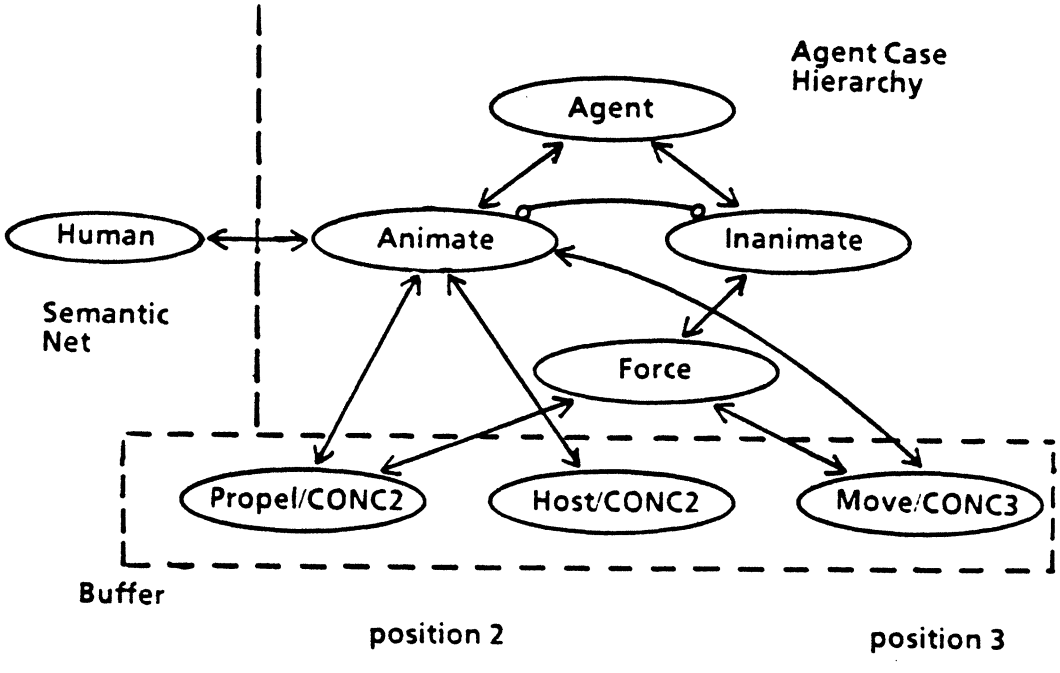



Figure 4.18. Agent hierarchy for example sentence.

specific cases thus sometimes requires assigning them more than one possibility.

The idea is that the predicate is connected to the maximally specific node that still encompasses all the possibilities. Yet, as we see, there are times when there may be more than one: when if we went up higher, we would include types that aren't fillers. So, we connect to the set of maximally general nodes such that their coordinates don't fit the constraints of the predicate (the minimal contrastive elements discussed earlier). Rather than having a separate FORCE case, as some case grammars do, we simply include it as a subtype of the INANIM(ate)-Agent type. Thus "Agent" is a rather general notion here. In keeping with the "upside down discrimination net" plan, disjoint subtypes ANIM and INANIM are mutually inhibitory. Recall that this is done to guarantee that one path in the hierarchy from the most specific filled case node to the root node "win"[12]. Also, note that HUMAN is connected to ANIM-Agent, so MALE-HUMAN inherits this property by activating HUMAN. We show no filler for the FORCE case, since none arises in the example sentence.

Figure 4.19 shows the appropriate (simplified) portion of the Object hierarchy. Notice that in many cases, corresponding superordinates in the lexicon and case hierarchy are connected. One of the objectives of the network for a case role given in Figure 4.13 was to prevent too much feedback in cases like this. In this example, the semantics of the case nodes prevents feedback above the Moveable node because no active predicate is attached above there, so no node above Moveable is satisfied.

Figure 4.20 shows the complete binding space for this example. Of import is the fact that buffer nodes are only connected to binders that are appropriate to them, e.g., TOILET/C1 is not connected to C1=Agent. M-

[12]Recall that this does not guarantee that a unique case is selected. There must be examples where a single path would contain a more general case that is appropriate to a different verb sense than the more specific one. (I can't think of any...). Such examples presumably are either globally ambiguous, or disambiguable from other information, either other cases appearing in the sentence, or syntactic information.

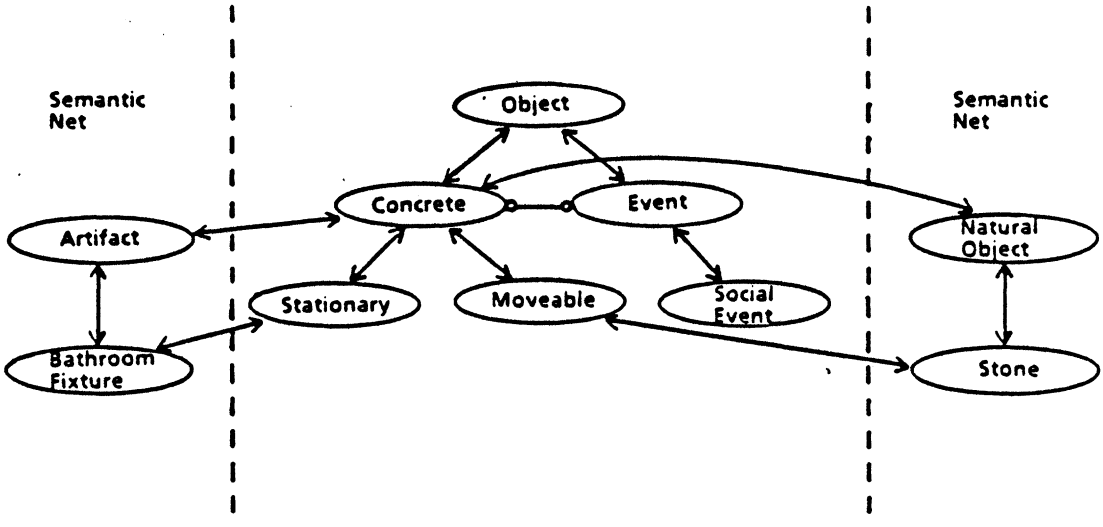



Figure 4.19. Object hierarchy for example sentence.

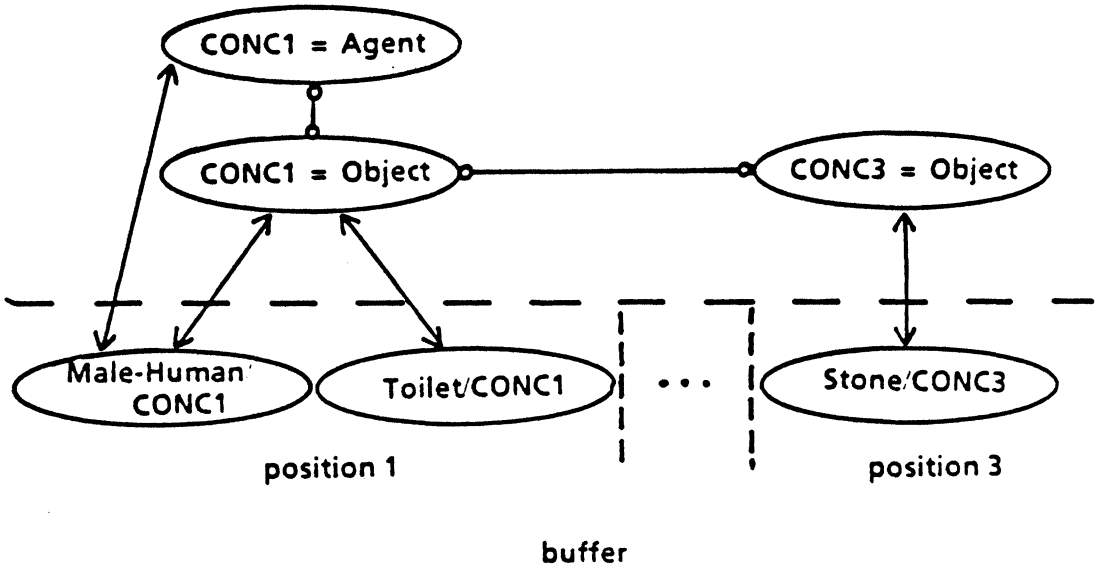



Figure 4.20. The binding space for the example.

H/C1 (MALE-HUMAN) is connected to both C1=Agent and C1=Object, since the Object class includes animate Objects, as in "John loves Mary." Finally, a look back at Figure 4.7 will help to give an idea of how this all fits

together.

The activation rules for the various types of units are given in Appendix 1. These rules are meant to reflect the basic design in an expedient manner. While not particularly plausible when taken as the behavior of single neurons, they reflect the use of the connectionist caveat that our units can be abstractions of the behavior of a larger collection of units.

Table 4.4 is the result of executing the simulation by hand. The values shown are unit outputs, rather than evidence (in the sense given in the Appendix). For case nodes, since these are complex, the states of the case are shown, as described in the section on Ugly Details. Although it appears rather formidable, we hope to guide the reader through it in as painless a manner as possible. Please bear with us. Helpful hints: The table is organized so that the various subnetworks are grouped together. The lexicon is laid out with subordinates above their superordinates, so that the spread of activation up the hierarchy can clearly be seen in the diagonal progression from left to right down the table. The letter symbols stand for states of the case node subnetwork, which are generally the same as in the text above, except that "pr" stands for "predicate", meaning that a predicate attached to that case node is firing. A blank means the unit is inactive. "Uninteresting" units, that once activated, never change or play any further decisive role, such as THING in the semantic network, have been deleted from the table, mainly because it would not fit on the page otherwise. For the record, FORCE and SOCIAL-EVENT enter state "pr" at iteration 7 and stay there, and INANIMATE is actively suppressed from iteration 8 on.

Here we go. At iteration 1, TOILET/C1 and M-H/Cl are activated (from "John" through the lexical access network as described in Chapter 3). In the next iteration, the binder units for this buffer position are activated. Also, in iterations 2 through 4, we can see the activation from the word senses spreading up the semantic network. As units in the semantic network become

Table 4.4. Unit Outputs of the Hand Simulation of "John threw a rock."
f: filled, f+: filled+, s: satisfied, s+: satisfied+, pr: primed

| Iteration | I | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MALE-HUMAN | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| HUMAN | | | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| LIVING | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| TOILET | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| BATH-FIX | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 |
| ARTIFACT | | | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 |
| STONE | | | | | | | | | | | | 4 | 4 | 5 | 5 | 5 | 5 |
| NAT-OBJ | | | | | | | | | | | | | 4 | 4 | 4 | 4 | 4 |
| ANIMATE | | | | f | f | f | s | s | s | s | s | s | s | s | s | s | s |
| Agent | | | | | f+ | f+ | f+ | s+ | s+ | S+ | S+ | s+ | s+ | s+ | s+ | S+ | s+ |
| MOVEABLE | | | | | | | pr | pr | pr | pr | pr | pr | s | s | s | s | s |
| CONCRETE | | | | | f | f | f | f | f | f | f | f | f | s+ | s+ | s+ | s+ |
| Object | | | | | | f+ | f+ | f+ | f+ | f+ | f+ | f+ | f+ | f+ | S+ | s+ | s+ |
| M-H/Cl | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 7 | 7 | 8 | 8 | 8 | 8 |
| TOIL/C1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cl = Agent | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 6 | 6 | 7 | 7 | 8 | 8 | 8 |
| Cl=Object | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| PROPEL/C2 | | | | | | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 6 |
| HOST/C2 | | | | | | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 |
| STONE/C3 | | | | | | | | | | | 4 | 4 | 4 | 4 | 4 | 4 | 6 |
| M0VE/C3 | | | | | | | | | | | 4 | 4 | 4 | 6 | 6 | 6 | 6 |
| C3=Object | | | | | | | | | | | | 4 | 4 | 4 | 4 | 5 | 5 |

active, they mark the related case nodes in the case hierarchies as 'Tilled[11]. For example, HUMAN becomes active in iteration 3, causing ANIMATE-Agent

(and ANIMATE-Object, not shown) to be filled in iteration 4. This reflects the fact that a male human, by virtue of being a human, can fill the role of an animate Agent. In processing terms, it is saying, "there's a filler around for this case, if anyone wants it" By iteration 6, all of the case nodes that can be filled by the two meanings of "John" are marked filled or filled+.

Now (iteration 6), the two meanings of "threw" become active in the CONCEPT2 buffer. These are directly connected to the case nodes representing their case frames. If one of these is marked filled+ , it becomes satisfied and begin to feed back to the lexicon (e.g., PROPEL/C2 becoming active causes ANIMATE-Agent to feed back to HUMAN). Otherwise, case nodes are simply be ready to be satisfied if a filler comes along. For example, MOVEABLE-Object and SOCIAL-EVENT-Object immediately become satisfied if a filler comes along. (We show this by entering $^{M}$pr" for "predicate" in the Table.) The activation levels of the semantic network nodes remain the same until and unless an attached case node sends feedback to the "filler" concept For example, ANIMATE-Agent becomes satisfied at iteration 7, causing HUMAN to raise its activation level at iteration 8. This ripples down the semantic network to M-H/Cl, making it become more active than TOILET/C1. M-H/Cl get a double dose at this step (10), because it gets increased feedback from Cl = Agent as well. The "satisfied+" state has spread up the Agent hierarchy, increasing the activation of Cl = Agent at iteration 9. This increase in evidence for M-H/Cl kills off its competitor, TOILET/C1. Thus "John" has been disambiguated by feedback from the Agent case hierarchy.

The role of CONCEPT1 then gets.disambiguated by a conspiracy between M-H/Cl and Cl = Agent These two nodes increase each other's activation until Cl = Agent defeats Cl=0bject We assume a maximum activation of 8, so the escalation stops there.

Back at iteration 11, the two meanings of "rock" enter the buffer as STONE/C3 and M0VE/C3. STONE in the semantic network becomes activated, causing MOVEABLEObject to become satisfied at iteration 13. This has several effects: MOVE/C3 gets feedback from MOVEABLE-Object, as does PROPEL/C2[13]. This causes PROPEL/C2 to win over HOST/C2 since its obligatory case has been filled. Also, MOVEABLE-Object feeds back through the semantic network to STONE/C3, allowing it to continue competing with MOVE/C3. Finally, the satisfied+ spreads up the Object hierarchy, giving C3 = Object a boost at iteration 16. This gives STONE/C3 a boost at iteration 17. The reader can verify, using the rules in the appendix, that this leads to a conspiracy between these two like with M-H/C3 and Cl = Agent, defeating M0VE/C3 by iteration 22. The result is a stable coalition between the MALE-HUMAN meaning of "John", the PROPEL meaning of "threw", the STONE meaning of "rock", and the binder nodes which encode the assignments of MALE-HUMAN and STONE to their roles in the sentence.

### 4.3.7. Some Implications of the Design

One thing left to discuss is what the design implies for the disambiguation phenomena in the Seidenberg et aL (1982) (STLB) study. First, recall that the disambiguation mechanism given in Chapter 3 already implies that within class ambiguities are resolved faster than between class ambiguities. The resolution was based on feedback to the meaning nodes (the buffered concept nodes here). In this chapter we have specified where that feedback comes from: the semantic network which overlays the lexicon. We now explain how this resolves the ambiguities in an appropriate example sentence of the STLB study. In the sentence, [M]Joe picked up the straw", STLB found multiple access

[13]**This appears to be a bug. One sense of "rock", the Noun sense STONE, has tilled a case in the Verb sense, MOVE! We don't think this is intuitively appealing, even in the case of agrammatic aphasics without syntactic information. One solution we have considered is to add binding nodes which bind buffer nodes to Predicate, which would then be a mechanism for the various possible predicates to compete with each other.**

of the meanings of straw. But the sentence "The farmer picked up the straw" elicited selective access. How does our model explain the difference? Given the action of the semantic network that is the lexicon in the model, the answer is simply that "farmer" primes concepts related to farming, while "Joe" does not. Thus the meaning of "straw" related to farming activates its primed superordinate nodes faster than the meaning related to drinking. Thus the "hay" meaning receives feedback sooner than the "soda-straw" meaning. Additionally, this satisfies the Object case faster, and receive feedback from that sooner than "soda-straw" does. The result is that the "hay" meaning wins. As noted above, due to the structure of the disambiguation mechanism given in Chapter 3 this happens faster than if the ambiguity was a noun-verb one. The above explanation requires more semantic relationships represented in the network than we have used so far, but the principle is the same as for the IS-A hierarchy.

This mechanism can also be used to explain "semantic garden path" sentences, such as "the astronomer married the star." Due to the priming of concepts related to astronomy, the "celestial body" meaning of "star" receives feedback sooner than the "famous actor" meaning. However, due to the case role mechanism, "celestial body" is not a candidate for "MARRIAGE-Object", and while initially more active than the "famous actor" meaning, it eventually loses out due to feedback from the case hierarchy to "famous actor".

### 4.4. Example Simulations

In this section we present the results of a small network built with the ISCON simulator (for Interactive Simulator of Connectionist Networks) (Small, Shastri, Brucks, Kaufman, Cottrell and Addanki 1982). This network is small, using only 40 units, and is based on an earlier design (as reported in (Cottrell & Small, 1983)), but will serve to illustrate some processing characteristics of our model, and two ways in which it can disambiguate verb senses. The differences with our current model are that it does not employ the case

hierarchy, the lexical semantic network, or binding spaces. Cases and their fillers are simply directly connected. The syntax portion of the model (discussed in the next Chapter) was not used either, so the sentences employed are not semantically reversible. This is an illustration of how far we can get without these "extras". The network successfully disambiguates the sense of "threw" (and "ball") in the following sentences:

1) bob threw a ball.

2) bob threw a ball for charity.

3) bob threw a ball to the dog.

4) bob threw the fight.

5) bob threw up dinner.

6) bob threw a ball up.

7) bob threw up a ball.

*8) threw bob ball up.

The last sentence illustrates what we feel is a desirable property of cognitive models of language understanding: the ability to "make sense" of an ungrammatical input. While syntax must eventually have a role in our model, it should not prevent understanding of these sentences, but only impose constraints on bindings that may be overridden. This is in sharp contrast to many previous AI models, which would most likely "break," or reject such input, without making sense of it.

We describe a trace of the processing of sentence 5, keeping in mind the possibility of sentence 7. Following this we discuss the processing of sentence 1, keeping in mind the possibility of sentence 2, as this illustrates a different disambiguation process.

The relevant subset of the network is shown in Figure 4.21. Note that typing information is encoded in the connections to the cases. FOOD "isa" VOBJ, but isn't usually a POBJ (here, exploded cases are lexicalized by tacking on the first letter of the predicate which defines them: PROPEL, VOMIT, THREW1 ("threw the fight") and GAVE ("threw a party")). Obviously, food can be propelled, and a low-weighted connection should be included to reflect this. For this example, we simulate hearing or reading the sentence by stimulating each word at the lexical level sequentially; the next word is introduced at each iteration of the simulation. Figure 4.22 shows a trace of the potentials of each of the relevant units for this example (POBJ and PREC are not shown. They do not fire in this example, as neither has a filler). We see

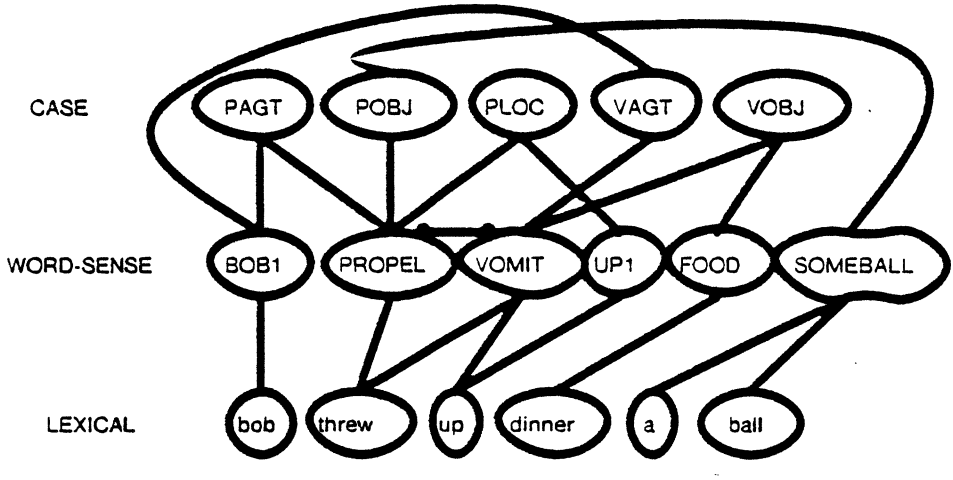



Figure 4.21. Subset of the network for example 1.

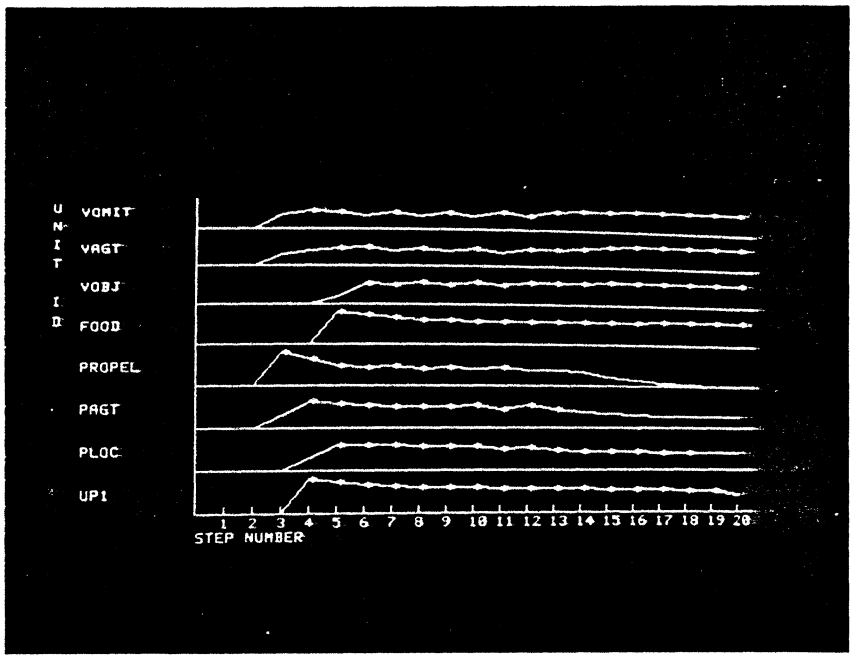



Figure 4.22. Graph of unit potential over time from the simulation of example 1. An asterisk indicates the unit is above threshold (i.e., firing).

that at iteration 3, BOB1 has primed PAGT and VAGT (along with TAGT and GAGT, not shown). These units will not cross threshold and fire until they get additional input from their associated predicates (at iterations 4 and 5, respectively). This is an example of how we prevent activation from spreading too much: conjunctive sites are used at the case nodes so that both the predicate and a filler must be present (i.e., firing) for the case node to fire and feed back to the filler and predicate. Nodes on the word sense level (fillers and predicates) have the feedback connections weighted so that they will not fire from top-down feedback alone; they must have bottom-up input first.

Also at iteration 3, the lexical unit for "threw" has excited the four units on the word-sense level (not all shown) representing its possible meanings. Figure 4.23 (from the same simulation) shows how collocations such as "threw up" are handled: there is a conjunctive connection from "threw" and "up" to VOMIT, so that VOMIT does not cross threshold until both are on. This is

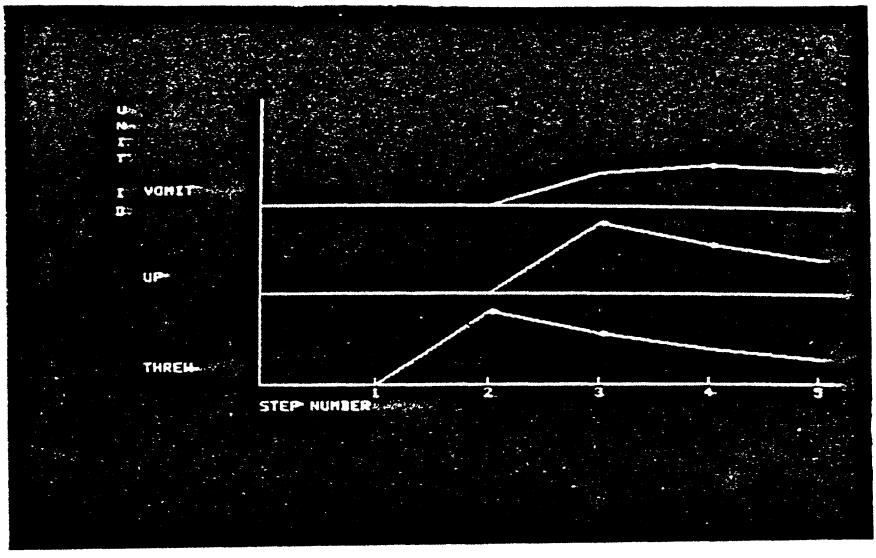



Figure 4.23. Processing a collocation.

consistent with some results of Swinney (private communication) which show that the sense of a collocation is not active until all the participating words are heard.

At iteration 4 in Figure 4.22 we see parallel activation of multiple hypotheses: both VOMIT and PROPEL are active. A mutually inhibitory connection between them helps insure that one will win. We also see part of the case frame for PROPEL has been primed (POBJ and PREC, not shown, have also been primed). One case, PAGT, has been filled; that is, the PAGT node has crossed threshold and is involved in a mutual feedback coalition with BOB1 and PROPEL. The rest of the cases are basically in "expectation" state: if a filler in the noun network comes along, they immediately cross threshold, as in the next iteration when activation spreads from UP1 to PLOC. (Please ignore that "up" is not really a noun; we use it here as an indication of location.)

We skip to iteration 8 (Figure 4.22). Here we show two cases filled for each predicate; FOOD fills the requirements for a VOBJ, and UP1 fills the

PLOC (location) case, while both PAGT and VAGT are filled by BOB1. However, all cases are not created equal. The reason PROPEL is no longer firing by iteration 12 is that weights are set in feedback connections from the case nodes so that obligatory cases, in this example PAGT and POBJ, must be firing in order for the verb to keep firing. Hence VOMIT wins here, since both VAGT and VOBJ have been filled. If, instead, "a ball" had been scanned, this would have filled the POBJ case, and the PROPEL coalition would have won. We can liken this to a voting procedure where the cases cast votes for their verb. The dropping off of PROPEL eventually leads to its cases also fading, so that by iteration 20 (Figure 4.24) shows all the units involved in the coalition) we have a stable coalition showing the result of the parse. This, for us, *is* the result: a pattern of activation on nodes representing the correct interpretation. Note that, since there is still residual activation on PLOC and PAGT, subsequent reinterpretation ("it splattered all over the ceiling") should be easier, although we have not yet investigated mechanisms for effecting this.

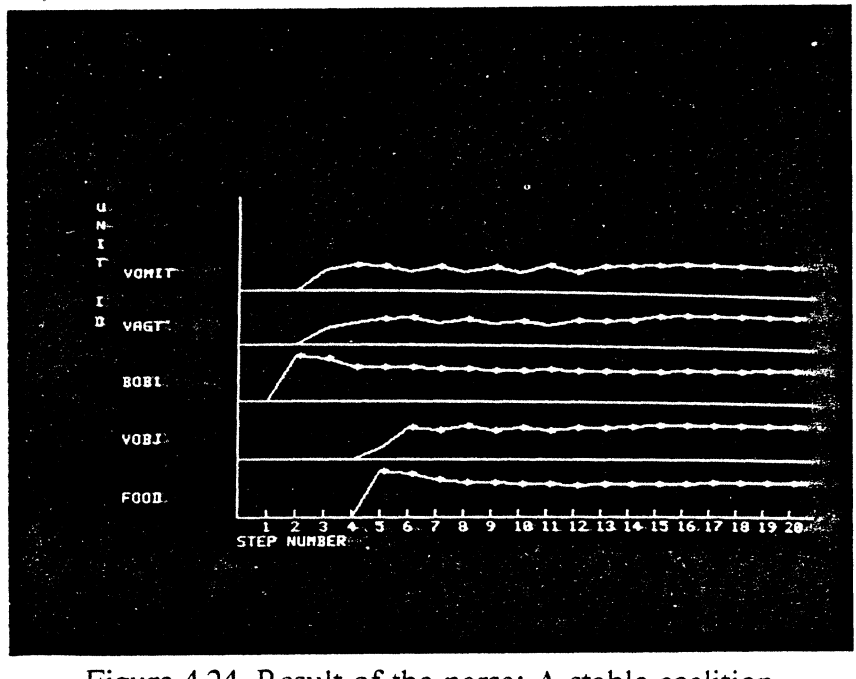



Figure 4.24. Result of the parse: A stable coalition.

The second example traces the result of the introduction of sentence 1, "bob threw a ball". Note that in this case, there are two meanings for "ball" represented in the network: the round kind and the "dance" kind. Here we have two senses of threw with two cases filled (Agent and Object), so the disambiguation depends on word sense frequency alone. This is represented by having different weights on the connections between the lexical nodes and the word sense nodes. Figure 4.25 shows the relevant subset of the network for this example.

Figure 4.26 from the processing of the "bob threw the fight" example mentioned earlier illustrates how connection weights reflect word sense frequency. (Processing is the same for the current example through iteration 3). Here we show the four senses of "threw" we have represented. The activation levels of the verb senses reflect the different weights on connections from "threw"; the senses considered more frequent are thus given proportionately more activation. This activation level difference is the same,

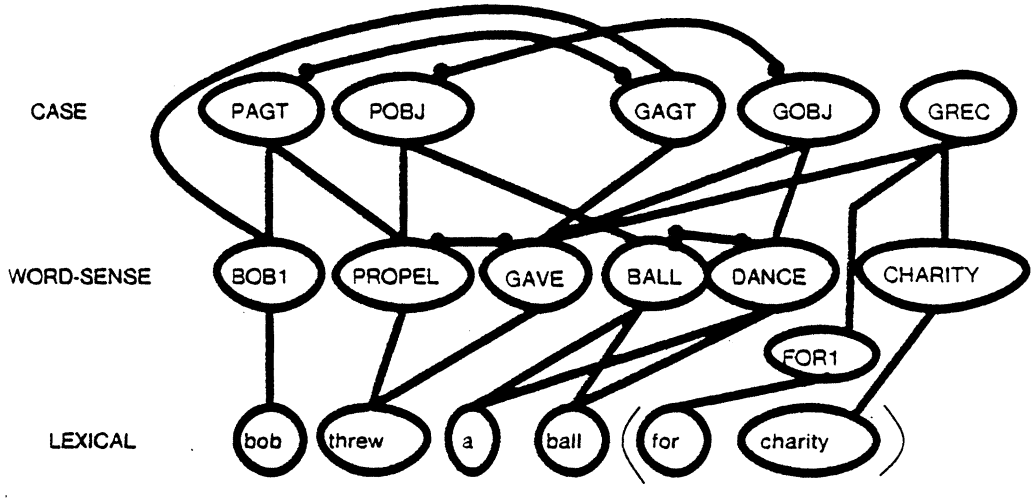



Figure 4.25. The subset of the network for example 2.

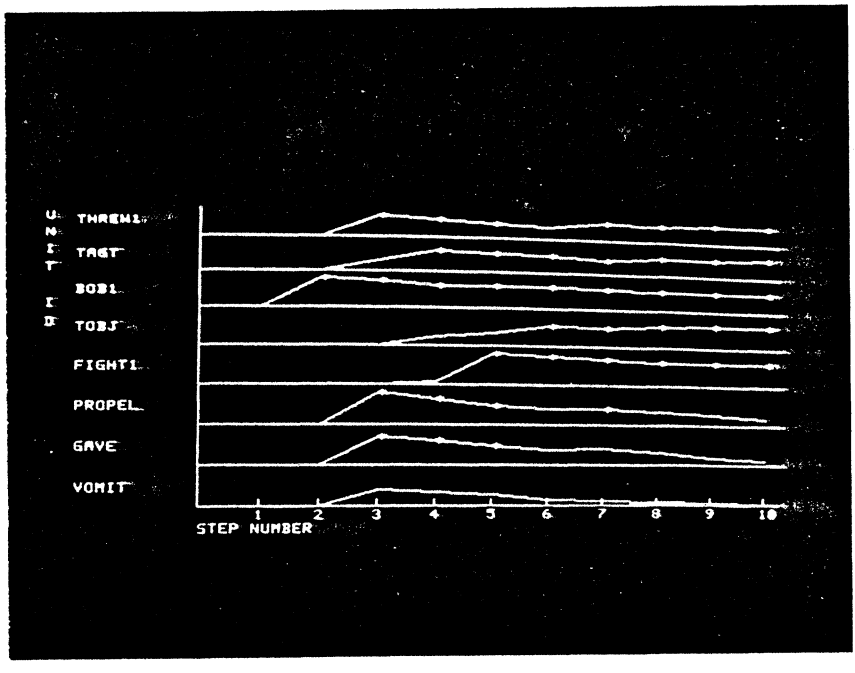



Figure 4.26. Unit potential over time for "bob threw the fight," illustrating frequency differences.

although harder to detect, in Figure 4.27, where we show the case frames for the senses of interest, PROPEL and GAVE (a poor lexicalization of the "host" sense). The POBJ and GOBJ cases are filled by the different senses of "ball" (SOMEBALL fills the POBJ case and SOMEDANCE fills the GOBJ case), so disambiguation has to result from the frequency effects.

We see this situation in iteration 7 (Figure 4.27). The two senses of ball (SOMEDANCE and SOMEBALL) are mutually inhibiting. The SOMEBALL sense, however, has an initial activation level higher than that of the "dance" sense. This, coupled with the lower initial activation level of the GAVE sense of "threw," enables the coalition involving PROPEL to "beat" down the activation of its competing coalition, so that by iteration 14, the SOMEDANCE unit is no longer firing. Now, with no support for the GOBJ case, it fades in the next iteration, causing GAVE to fade also, since GOBJ is an obligatory case for GAVE. In a domino fashion, the coalition for GAVE collapses,

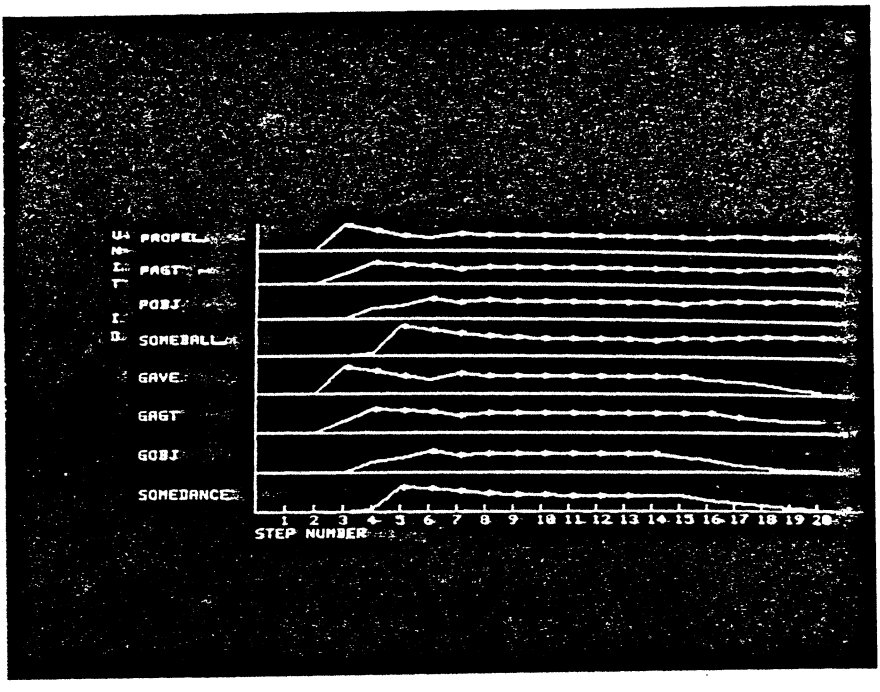



Figure 4.27. Output from the simulation of example 2. Lower overall initial activation of GAVE coalition results in PROPEL coalition winning.

resulting in the proper coalition (the four upper units in Figure 4.27, plus BOB1, not shown).

Now, if the sentence ends with "for charity," the beneficiary case of GAVE (GREC) is filled (see Figure 4.28). The result is that GAVE receives more feedback (from three cases instead of two). GAVE in turn gives more activation to GOBJ, enabling more feedback to SOMEDANCE, which is enough to help it overcome SOMEBALL, resulting in the correct interpretation.

These examples illustrated several processing aspects of our model:

(1) Parallel processing; all units are updated simultaneously, so processing occurs at all levels at the same time. Bindings are being established between the word-sense and case levels while new words are arriving at the lexical level and stimulating their senses at the word-sense level.

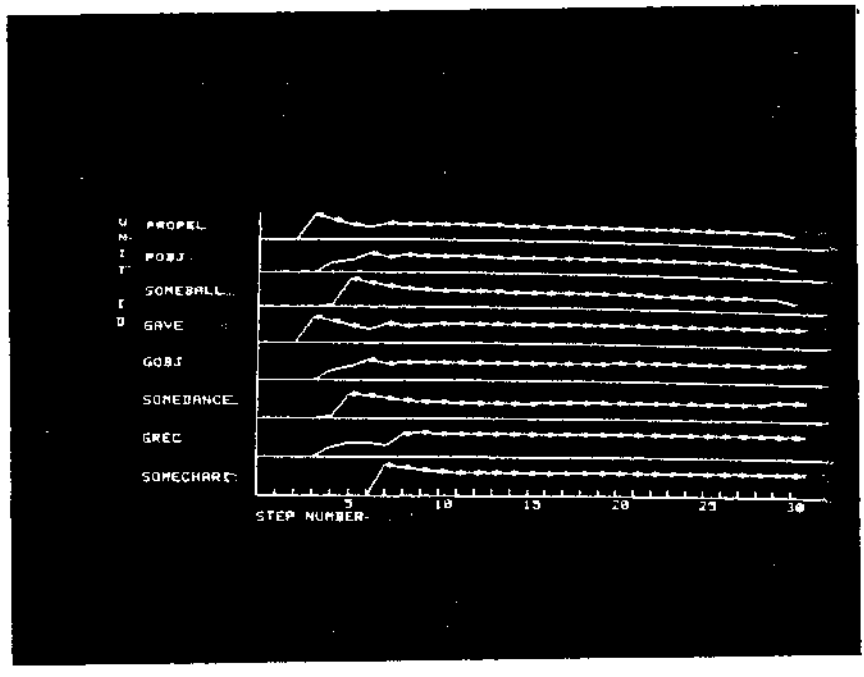



Figure 4.28. Output from the processing of "bob threw a ball for charity." The third filled case (GREC) helps the GAVE coalition win.

(2) Parallel activation of multiple hypotheses: the four senses of "threw" all fire simultaneously, while several different agent and object cases fire and compete over several iterations until the most active wins. Many NLP programs choose an alternative and backtrack if wrong. Note that in this paradigm, it is actually easier to activate all possibilities, with the one that fits best winning. We saw this happen in all of the examples with the case nodes moderating the interactions between the senses.

(3) Expectation-based parsing: cases which are primed fire easily. These are strongly typed case roles so that the expectations are specific to the verb which defined them.

Also, the examples illustrated some control aspects:

(1) Distributed decision making: decisions between competing coalitions are decided by inhibitory links between units of the same type. Case roles of the same type (e.g., all agents) are mutually inhibiting. This is a general

technique in connectionist models: incompatible value units are in a "Winner Take All" network (Feldman and Ballard, 1982) that always settle to only one unit highly active. Note that no homunculus appeared to force decisions.

(2) Word sense disambiguation: the structure does this in two ways: (a) Obligatory cases provide more feedback than optional ones, so that a predicate with all of its obligatory cases filled defeats any other predicates that do not. This is the way in which VOMIT defeated PROPEL in "bob threw up dinner". Even though both verbs had two cases filled, the missing POBJ case caused PROPEL to fade out. (b) When two verbs have their obligatory cases filled, the decision is made based on word sense frequency differences. The more frequent senses get more initial activation, and this edge is enough to defeat the competing coalition.

The problems with this design were discussed in the introduction. As we mentioned there, there is no way in this model to assign case roles in semantically reversible sentences, no clean way to interface with syntax, and not enough overall organization. The design presented in this chapter overcomes these objections, while preserving the positive aspects of this model. What is lacking is an implementation to verify that these claims are true. We hope, however, that this presentation has been sufficiently detailed to convince the reader that this is possible. Also, it should be clear that the technique employed in structuring the case nodes is general enough to overcome any problems that might arise. That is, when a particular behavior is required, we can specify the behavior desired in terms of logical predicates, apply the unit/value principle to implement those predicates, and use unit functions to reflect the control specified by the logic.

## 4.5. Conclusions

In this chapter we presented the design of a connectionist semantic interpreter. By virtue of it being connectionist, we get to add the modifiers:

massively parallel, completely distributed, and neurologically plausible. The processes of assigning semantic roles and disambiguating word senses are mutually constraining, parallel processes that communicate through activation and feedback channels in an *active* knowledge representation. The efficacy of the model was demonstrated in the results from a preliminary implementation. From a Cognitive Science point of view, our model provides a framework in which various theories of aphasic and psycholinguistic phenomena may be tested.

There is, however, much left to do. The "proof of the pudding" of the current design, an extensive implementation, is necessary before we can claim that all the problems associated with the earlier design have been overcome. This design also could use improvement: one would like to have more of the case grammar of Cook (1979) incorporated here, with covert case roles, verb types, subject preference orderings represented in some way. We plan to do this by making use of entries in the lexicon for verbs, as we did for nouns. The hierarchical relationships relationships between verbs would fall out of such a representation: Similar verbs would inherit cases from superordinate verb types, while allowing for exceptions to this inheritance. The connectionist implementation of inheritance hierarchies with exceptions in Chapter 6 is viewed as a step toward this improvement. Thus, the specification of the case frame of a verb would then be through the lexicon, rather than through direct connection, and idiosyncratic case marking could be represented.

Further along, there is the problem of higher order representations in this framework: frames, scripts, and plans. This involves structures beyond the clause, and interactions with the various memory types (short, medium, and long term; and episodic) that have been identified by psychologists. While some of the fundamental problems of memory have been addressed (Feldman, 1981, 1982; Shastri & Feldman, 1984), it appears to us that a clear characterization of memory which addresses the known data on short term vs. medium term vs. long term memory has yet to be determined. This is fruitful

ground for future research.

Finally, within the scope of the present work, the syntactic level sentence processing which interacts with the case assignment mechanism ar disambiguation machinery of this chapter needs to be specified. This is th subject of the next chapter.

# CHAPTER 5

# A CONNECTIONIST SYNTACTIC ANALYZER

## 5.1. Introduction

Our major emphasis in this work is on word sense disambiguation, rather than parsing *per se*. However, in order to provide disambiguating feedback for noun-verb ambiguities, it is necessary to have a parser. In this chapter we present the design of the syntactic processor, and the results of a simulation of the model. While this is a preliminary version, lacking many of the features one would want in a complete parsing system, it demonstrates the feasibility of the approach, and there are aspects of this model which are rather general and interesting as partial specifications of a parsing mechanism in their own right. These include the mechanism for assigning constituents to their roles which has a natural interface with our model of semantic role assignment, and the ability to represent syntactic attachment preferences. Also we might include here the fact that this parser uses the massive parallelism inherent in the connectionist approach, with the concommitant distributed decision making. And, unlike the parser of Pollack & Waltz (1982; 1985), there is no interpreter that builds a network based on the input sentence and then runs it in parallel; this parser uses a network that is fixed, yet responds flexibly to the input.

### Overview

In order to implement this parser, we developed a grammar formalism suitable to our needs, and wrote a LISP program that takes as input a dictionary and a set of grammar rules, and outputs commands to the ISCON simulator to build the network. This network implements a top-down parser.

Expectations are set up compatible with an S. As input comes in (words are activated at fixed intervals), the structures compatible with the input continue to be active, while productions that are incompatible with the input are turned off. After a settling period, a stable coalition in the network represents the parse tree. Ambiguous lexical items at the leaves are disambiguated through feedback from the parse tree as it develops. One important difference from the Pollack and Waltz (1985) spreading activation parser is that the assignment of constituents to their syntactic roles is explicitly represented by nodes in the tree called *binding nodes.* The combination of top-down expectations and bottom-up evidence comes about at the binding nodes, which combine the evidence and compete with one another through mutual inhibition. These nodes can be used to interface with case role assignment in the semantic analyzer.

Another important difference from Pollack and Waltz is the fact that we use a fixed network. This is accomplished by having a pool of copies of constituent recognizers. These may be activated as needed by constituents that form their parent constituents. All constituents are permanently connected to constituents that could compose them through a selection network. Whenever a constituent is expected by some other constituent, for example an NP by an S, the S can select the first available unused NP recognizer, thereby activating it, causing it, in turn, to expect its constituents. The selection rules are such that a constituent recognizer may be selected as long as it is not bound to a parent constituent. Therefore, more than one parent constituent can select the same (sub-) constituent recognizer and compete for it. This precludes the grammar from being left-recursive, since a constituent could end up picking itself with these rules.

**Current Limitations**

As a first cut, this parser is less elegant than one would hope, and less powerful than what is needed for a "real" parser. Although it can handle

recursive constructs, and thus has a mechanism for selecting constituent recognizers as needed, no attempt was made to do the same for buffer positions, so it accepts only sentences of a fixed maximum length. Also, the fact that the network is of fixed size means that there are only a fixed number of constituent recognizers, so the number of constituents of each type that can occur in a sentence is limited. Also, this parser has not been tested on an extensive grammar, therefore, take any claims with a grain of salt. Finally, we haven't done anything about: (1) optional repetition of constituents (2) feature marking of constituents (hence no feature agreement checking) (3) anaphora or (4) gapping. We would hasten to point out that this doesn't mean that mechanisms for handling these phenomena can't be implemented in this framework; simply that we haven't done it yet.

The rest of this chapter is organized as follows: We first review some of the psycholinguistic and neurolinguistic data on syntax (some of this will be quick: an extensive review of the neurolinguistic data is the subject of the next chapter). Then we review what the semantic analyzer needs from syntax, and describe our grammar formalism, leading into a description of our parser, and finally, we describe a run of the implementation on a sentence.

## 5.2. The Data

### 5.2.1. Introduction

Unlike the semantic analyzer, this parser was not written with a view toward trying to achieve psycholinguistic or neurolinguistic reality, nor was it intended to cover a wide range of phenomena. We were only interested in a parser that could handle the simple sentences of the Seidenberg et al. (1982) study, so we have not considered representations of gaps or lexical preferences, for example. However, the parser does have obvious methods for dealing with attachment ambiguities, and predictions relating to some psycholinguistic theories are possible. Hence we will only discuss a handful of studies that seemed particularly relevant, concentrating on relating our work to Frazier's

(1979) parsing strategies. In the service of making this section self-contained, we will try to point out the relevance to our model here, rather than in the description of the parser.

*5.22.* Psycholinguistic Data

We will start with lexical ambiguity results. The Seidenberg et al. (1982) study discussed in Chapter 3 found that regardless of the strength of the context, both readings of noun-verb ambiguous words are activated, followed by disambiguation within 200 milliseconds. We interpret this result to imply that the human sentence processor is at least in part "bottom-up", that is, all possibilities (at least for lexical constituents) are activated regardless of the developing syntactic representation. Our parser follows this scheme, activating all readings of a word in the word sense buffer, followed by rapid disambiguation through feedback from the developing tree.

On the other hand, for noun-noun ambiguous words in the first clause of a two clause sentence, where the disambiguating information is contained in the second clause (as in *the teacher looked at her* **pupils** *and noticed that they were dilated)* people appear to be able to maintain the activation of both meanings for as long as 500 milliseconds (Hudson & Tanenhaus, 1984). In the model of the word sense buffer we shall present here, (which is slightly different than the model of Chapter 3), an early (wrong) choice of the meaning of a semantically ambiguous lexical item does not wipe out the complete representation of the alternate definition; recovery is easier than if the wrong syntactic choice is made. Also, syntactic feedback will tend to support both definitions of the word (for a semantic ambiguity), which should aid in maintaining both readings longer. We next consider higher level syntactic aspects of sentence processing.

Frazier (1979) has hypothesized three strategies used by the human sentence processor, which she also presents data in support of:

(1) *Minimal Attachment.* Attach incoming material into the phrase marker being constructed using the fewest nodes consistent with the well-formedness rules of the language.

(2) *Late Closure.* Whenever possible, attach incoming material into the phrase or clause currently being parsed, except where this conflicts with Minimal Attachment.

(3) *Weak Semantic Principle.* Constituent assignment decisions are not made *in violation* of lexical semantic constraints on the possible relations between words of a sentence, unless no other analysis of the sentence is possible. Elsewhere, she restates this as: "The parser uses semantic constraints during its syntactic analysis but only to reject anomalous analyses." (Frazier, 1979, p. 73).

The Minimal Attachment Principle is simply a statement of what appears to be a sound strategy: use the fewest nodes possible to parse a sentence. One problem with it is that it depends on the grammar one uses; different grammars would lead to different predictions. On the other hand, one could *assume* the Minimal Attachment principle and then try to derive internal grammars from timing studies. In any case, the model we propose follows the Minimal Attachment Principle. Without having described the model yet, we can explain this intuitively as follows: The model works by combining top-down expectations and bottom-up input. Imagine a grammar representation which is *active*, in the sense that as parts of productions are recognized, activation spreads to the next part of the production. Different productions in the grammar *compete* for the attachments of constituents that are found in the input. The more nodes involved in a particular interpretation, the farther the activation has to spread, and the longer it takes to activate those nodes that the input actually attaches to. Meanwhile, if there is a representation that matches the input that involves fewer nodes, this will become activated faster and get a head start over the representations involving more nodes. Thus our model

explains Minimal Attachment as a *timing* phenomenon, involving the latency of activating simple versus complex representations through a spreading activation network.

As for Late Closure, the rule for deciding between alternate productions of the same constituent in our model is a simple voting scheme that prefers longer productions over shorter ones. On the surface, this looks like it supports Late Closure, but the interactions with enclosing constituents that could also take the incoming constituent makes the relationship slightly more complex. The possible attachments are represented by the binding nodes mentioned in the introduction. These compete with one another through mutual inhibition. One source of the evidence for a particular attachment (i.e., excitation for a binding node) comes from the production that is the target of the attachment. The production that has more constituents filled so far will give more evidence to the binding to itself. So the prediction is that late closure depends in some measure on how much of the "current" constituent has been satisfied so far versus the enclosing constituents. This prediction must be modified by consideration of Minimal Attachment; if one of the productions that could use the constituent has to do so through "calling" another production, this will take longer, and it may lose the attachment competition. All of this depends on particular parameters of the implementation: how long an attachment takes to "win" over others, how long it takes for activation to spread to subordinate productions, and of course, the grammar used.

The Weak Semantic Principle (WSP), according to Frazier, presupposes that syntactic and semantic analysis proceed in parallel, consistent with our model. In the restatement of the principle, Frazier's claim is that the syntactic analyzer uses semantics only to reject parses that would result in anomalous interpretations, unless that is the only interpretation. In our system, the syntactic analyzer's attachment decisions get feedback from corresponding semantic attachment decisions; semantically plausible attachments would

receive more support, and would thus win over attachments that had no support, due to anomaly. However, if there is no semantically plausible representation, then the syntactic analyzer would simply not get any feedback from the semantic analyzer; syntactically plausible attachments are always constructed, insofar as they match with what has been constructed so far. There is also the issue of timing here; since we have two independent systems running in parallel, one may work faster than the other. Since we have yet to actually connect our syntactic and semantic analyzers, we can only speculate, but our hypothesis is that syntactic analysis requires less computational effort than semantic analysis, and so the semantic analysis would generally lag behind the syntactic analysis. However, it should be possible to manipulate the syntactic and semantic complexities so that semantic analysis proceeds faster. Some evidence for this stems from a study by Blumenthal (1966) in which subjects interpreted sentences such as *the man the girl the boy met believed laughed* as a compound subject followed by a compound predicate. However, it is unclear whether such sentences prove anything; given that they probably only occur in psycholinguistic experiments and the halls of academic departments.

Because of these timing considerations, our model spans two hypotheses Frazier calls the "Intermediate semantic hypotheses", so-called because they attribute a larger role to semantics than the Weak Semantic Principle:

(1) Semantic constraints are used during the syntactic analysis of a sentence and these constraints may dominate the parser's syntactic conclusions.

(2) The parser uses semantic constraints during its syntactic analysis but only uses these constraints to select the most plausible of its competing syntactic analyses.

We hypothesize that the first possibility (semantics dominating syntactic analysis) occurs in cases where the semantic analysis is stereotypical, and the syntactic analysis is not, so that the semantic analysis finishes first This in fact.

is the explanation for some "garden path" sentences; in one such as *the old man the boats*, the semantic analysis has dominated the syntactic analysis, biasing it to an unrecoverable (without conscious intervention) state. The second possibility is the more typical case. Our model will pursue all syntactically plausible attachments that are not otherwise resolved by Minimal Attachment or the "longer productions are better" rule until semantic feedback resolves the ambiguity. It is blissfully unaware of semantic implausibility; this is only indicated by a *lack* of feedback, rather than *negative* feedback. Hence, contrary to the original statement of the WSP, our analyzer works on a principle of "innocent until proven guilty": it considers implausible attachments until they are beaten by more plausible ones. Thus the question of "what information is used when" reduces to "what information is *available* when."

Although it is not implemented here, another assumption we make is that the parser has initial biases for the competition between bindings (in the form of inhibition biased one way or another) based on frequency of attachment and lexical considerations. When the semantic selection is for the less frequent binding, the competition takes longer to resolve. Thus our system does not neatly fit into Frazier's breakdown of different degrees of use of semantic information. Rather than conforming strictly to (1) or (2) above, the use of semantic information depends on the interaction of the assumptions of independent parallel processing of the two analyzers and the varying relative processing speeds depending on the input.

Rayner, Carlson and Frazier (1983) have demonstrated that there are interactions of syntactic and semantic preferences for attachment. Minimal attachment predicts that in such sentences as *the cop saw the burglar with the binoculars*, the PP *with the binoculars* will be attached directly to the VP rather than to an NP with *the burglar* as a sister. When the semantically preferred attachment is different from the syntactically preferred one, as in *the cop saw the burglar with the gun*, reading time is increased, with longer fixations in the

region of conflict. This argues for an independent contribution of syntax and semantics to the attachment process. It does not necessarily mean that one precedes the other; as discussed above, one system may just operate more slowly than the other on this input. There is a tantalizing correlation between the fact that subjects spent more time scanning the syntactically ambiguous section of the sentence when the semantic preference was for non-minimal attachment, and the way that minimal attachment preferences fall out of our model. Perhaps the longer fixation times are necessary to spread activation through the longer production. This is an area for further study.

Rayner et al. conclude from their experiments that the results support "a model in which independent mechanisms are responsible for structural parsing preferences on the one hand, and lexical, semantic, and pragmatic preferences on the other." We therefore conclude that their study supports the overall model presented in this thesis.

### 5.2.3. Neurolinguistic Data

This data is discussed in the following chapter; here, we briefly mention the implications of some of this data for our model. The general picture emerging from studies of agrammatic and Wernicke's aphasics is that access to syntactic and lexical-semantic information can be independently disrupted, which we take as support for our overall model. Linebarger et al. (1983) have shown that patients who appear unable to use syntactic clues to comprehend sentences can nevertheless make relatively complex grammaticality judgements, which also supports an independent syntactic processor. This result implies that agrammatics can *compute* syntactic representations, but can't *use* them to form semantic interpretations. We assume that the interpretation this has in our model is that the syntactic processor and the semantic processor have become "unlinked", as far as the ability to map syntactic and semantic constituents to one another. The only pathway left is through the word sense buffer. Our model currently has no mechanism for the detection of

ungrammatically, so we can't make claims about grammatically judgements, only about what kinds of information are available to the parser. It is not clear how much of the problem in detecting certain types of malformed sentences for these subjects was due to their short term memory problems. Since we also don't have a model of short term memory, (unless one uses the word sense buffer for this purpose), we will have to leave for future work the exploration of the ability of our model to predict the types of malformed syntactic constructions that can be detected without semantic information or short term memory.

We can however now mention the lesions that would be appropriate in our model for a Wernicke's aphasic. These patients appear to have severe comprehension deficits. Their speech is usually meaningless, but fluent. It generally appears to be syntactically correct, while lacking content due to severe paraphasias (word and phoneme substitutions). We assume that the disruption necessary for this behavior in our model would be either the wholesale destruction of the lexicon, or more forgivingly, simply access between the word sense buffer and the lexicon. If similar functional units in production were affected, then all the Wernicke's aphasics could do is activate the syntactic generator, which would generate a syntactic representation without the "meaning" nodes filled in in the word sense buffer or semantic constraints on the function words used The result would be a string of syntactically correct elements (at a gross level; the *classes* should be correct) with the function words and other items that can be wholely syntactically specified in their proper positions, with random content words *of* the proper class in the open class item slots.

On the other hand, we would predict that Wernicke's aphasics could make grammaticality judgements as well, if they could be made to understand the task. Unfortunately, this isn't the case. However, patients with echolalia. who can only repeat what is said to them, have been known to spontaneously correct syntactic errors in the repeated sentence. This too argues for an

independent syntactic processor.

## 5.3. The Parser

### 5.3.1. Background

#### What Do We Need from Syntax?

There are basically two things we want from our syntactic analyzer. The first is the syntactic disambiguation of lexical items, to prevent spurious predicates and fillers from confusing the semantic system. This can be done through feedback to the syntactic features in the word sense buffer which form a grammatically correct sentence. The decision machinery in this buffer will then kill off the meaning features for the meanings corresponding to the wrong syntactic class, which then will stop sending input to the semantic analyzer. Secondly, for semantic role attachment, the semantic analyzer needs constituent role assignment information. For example, there is no way for a purely semantic analyzer to make the assignment of Agent and Object in *John loves Mary*, since both John and Mary are equally likely candidates for both roles. Of course, the semantic analyzer could have the weights between binder nodes set so that the first "=Agent" node activated always won over the "=Object" node. But then it would never interpret the Passive correctly. There must be information from the syntactic analysis corresponding to the Passive transformation (see Figure 4.16, reprinted here as 5.1). There are two components to this process, of which we we shall describe only one. The first is mapping syntactic entities to their corresponding conceptual entities (shown as direct links in the Figure); this we shall not implement here. This would be part of the process of getting all of these components we have described to work together. As a tactical measure, we have decided to leave that for future research. The mechanism needed is essentially a binding mechanism similar to the one described here for constituent-role assignment. The second is activating binding nodes that represent the assignment of the syntactic entities

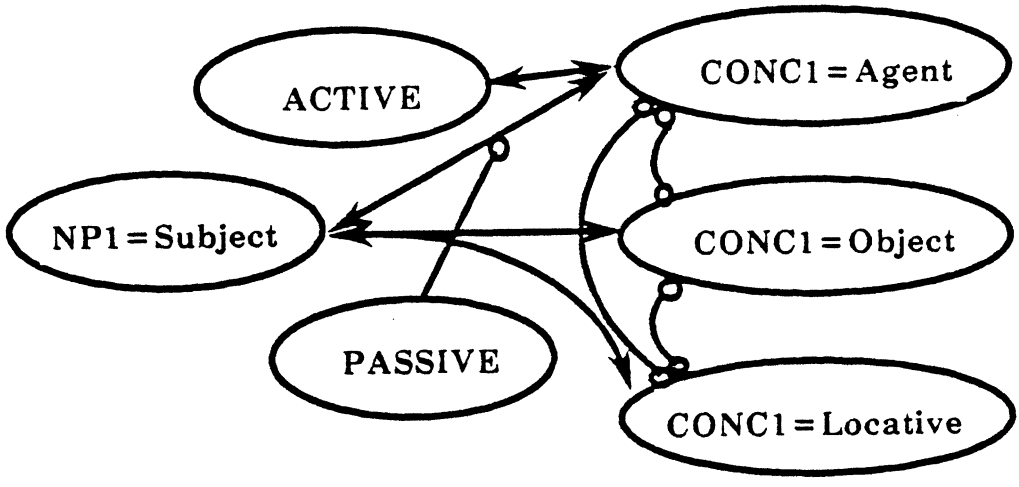



Figure 5.1. The Passive Transformation.

to their roles; this our parser accomplishes.

**The Role of Binding Nodes**

Binding nodes are really the heart of the whole system. In the sense that what we are implementing here is a constraint relaxation process, connections between binding nodes encode constraints between assignments. Within syntax or semantics, the local constraints are that the same constituent can't be assigned to more than one role, and one role can't have more than one constituent[1]. Thus connections locally are generally inhibitory. Between these two systems, there are constraints such as, if the verb is Passive, then the semantic constituent corresponding to the syntactic constituent filling the Subject role is not the Agent. Constraints go the other way too. In *the cop saw the burglar with the binoculars*, the attachment of the prepositional phrase is underdetermined syntactically. It can be attached to the enclosing VP or to an Nbar with the adjacent NP (apologies for the "mix and match" linguistic

[1]This constraint would have to be relaxed in the case of covert cases in semantics. For example, "John" is both the Agent and the Object in "John ran". The Agent case is overt, but the Object case, the thing being affected by the running, happens to be equal to the Agent here. See Cook (1979) for a discussion.

terminology). This is true semantically as well, but one would usually assume that the binoculars were used for the seeing; rather than that the burglar was carrying (perhaps stealing?) the binoculars. This is clearer in the case of *the cop saw the burglar with the gun* where it is less likely that the cop was using a gun to see the burglar (a flashgun? a gun with a telescopic sight?). This is a case where the semantic role assignments can constrain the syntactic ones (see Figure 5.2). In this figure, we are assuming the implementation of conceptual frames in semantics. The CONC4=CONC3MOD is intended to represent the assignment of "the gun" to a role in the "burglar" frame. CONC4=INSTRUMENT represents the assignment of "the gun" to an outer case of "see". Since the former is more likely semantically than the latter, and corresponds to a modification of an NP by a PP syntactically (among other constructions), this will lend support to the proper attachment.

At this stage the reader hopefully sees what we mean by the centrality of the binding nodes to our theory. Since they correspond to the assignment arrow in a programming language, what we have here is a parallel competition between assignments. The result gives a simple interface between the syntactic and semantic systems: the binding nodes in one constrain the binding nodes in the other. Thus, our design of the parser started "from the inside out", with

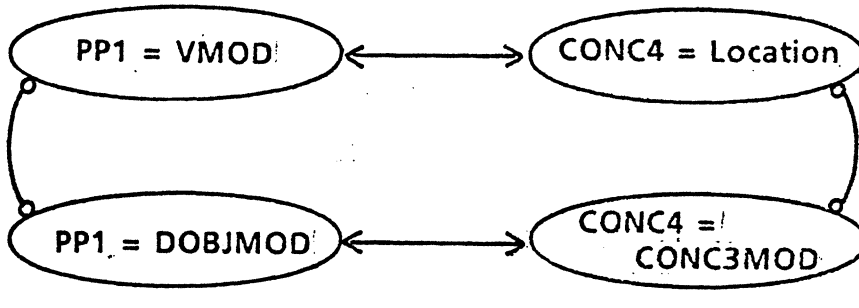



Figure 5.2 Mutual constraints.

the necessity of binding nodes.

**The Grammar Formalism**

The grammar that we have developed to enable us to automatically generate the parsing network explicitly represents roles and the constituents that can fill those roles. Hence we call it a *role-constituent grammar*. We are not aware of any formalism in linguistics that is similar to this although it appears to be weakly equivalent to a context free grammar. However, as has often been remarked, the notation one uses can affect the way one thinks about a problem. In this case, the notation is handy for generating a network which contains binders for constituents to their roles in parent constituents. An example grammar is shown in Figure 5.3. The left hand side of a production in this grammar is a constituent. The right hand side is a list of roles followed by a set of constituents that can fill those roles. In this example, the Head of a noun phrase can be filled by either a PRO a NOUN.

Since there are sometimes a number of alternative constituents that may fill a role, the dot between the role and the set of possible constituents corresponds to a set of binders that must compete for that role. On the other hand, every set that a constituent is in corresponds to a possible role for that constituent, so that binders for that constituent to these roles must compete as well. For example, any particular NP could be the Subject, Direct Object etc.,

```
S->   Subject.{NP} Predicate.{VP}
      Predicate.{VP}

VP->  Main.{VERB} DirObj.{NP}
      Main.{VERB} IndObj.{NP DirObj.{NP}
      Main.{VERB}

NP->  DetPhrase.{DET} Head{NOUNf
      Head.{NOUN}
```

Figure 5.3. A sample role-constituent grammar.

so binders to these roles compete. The network is set up so that unless a role is expected, then that binder doesn't become activated, so in practice competition does not involve all of the binders.

Another thing to notice about this grammar formalism is that the role a constituent can fill is context sensitive within a production. For example, in the first production for an NP, a NOUN can fill the role of the Head, but not a PRO. In the second production, the Head can either be a NOUN or a PRO. Thus there must be two "Head recognizers", one for each kind of Head. Of course, because we haven't implemented feature marking, this production is unnecessarily complicated, since the determiner would mark the NP as "determined", which would conflict with a PRO Head.

Now given this grammar, a dictionary containing the possible syntactic classes of the lexical items, and some "magic numbers" (how many copies of each kind of constituent recognizer there will be, and how long the word sense buffer will be), we can generate a network that will recognize sentences that match these grammar rules. After a tour of the word sense buffer, we shall describe this network in greater detail.

**The Word Sense Buffer**

We used a slightly different version of lexical access from the model of Chapter 3, basically because it was easier to implement with respect to the interface with the syntactic analyzer. However, it has some interesting properties in its own right, which we will describe here. For comparison purposes, we show the network for "deck" in Figure 5.4. The lexical node "deck" activates "grandmother cells" for each of its definitions. These are self-stimulating, to keep the definition around after the lexical node decays. (Thus the lexical node can be re-used later. Enablement of the "def" nodes in successive buffer positions is handled by control nodes that sequence through the buffer. This figure represents one buffer position.) The definition nodes, in turn, are connected to feature nodes representing syntactic class and

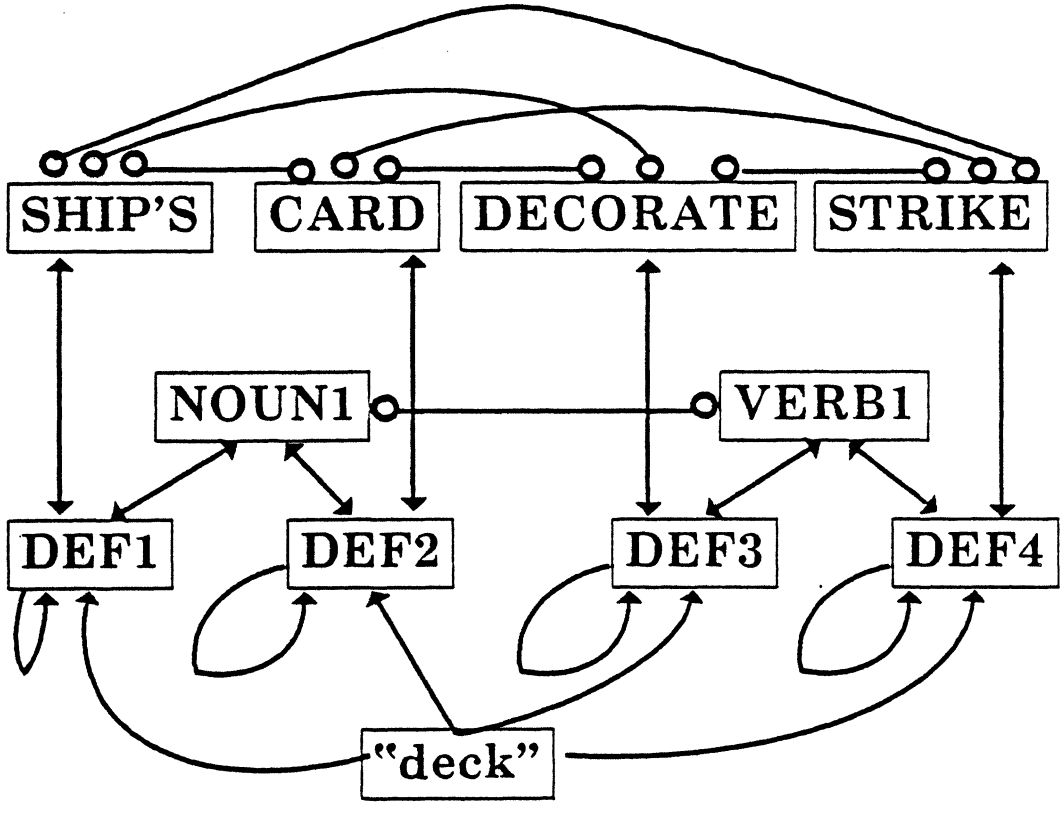



Figure 5.4. A word sense buffer position. Inhibitory links between DEF nodes left out.

meaning. The definition nodes are also mutually inhibitory, but the inhibition weights are such that they can't defeat one another until decisive feedback to the meaning nodes supports one "def" node over another. The "meaning" nodes are the word sense nodes of the previous chapter, and have no further connections to the syntax network. The syntactic class nodes are shared between meanings of the same class. Thus feedback to one of these from a role node above will support both definitions within a class, and kill off the out-of-class syntactic node, followed by the out-of-class meaning and definition nodes. (Of course, if there is only one meaning for a syntactic class, then feedback to that class is enough to cause that "def" and meaning node to win.)

Feedback to one of the meaning nodes, on the other hand, will kill off the other meaning nodes, but the supported definition node, say "def1", will increase the activation of NOUN1, which causes it to win over VERB1. However, "def2" will get both extra support and extra inhibition. It gets extra inhibition from "def1", but since there is a positive pathway between the two, through the NOUN1 node, it also gets extra support. With the current settings of weights we have used in our simulation, the result is that "def2" gets inhibited, but not below threshold. However, since "CARD" has been killed by "SHIP'S", "def2" is invisible to the rest of the network. Only the features compatible with the "def1" are visible. A prediction this network makes is that it would be easier to recover from a *within-class* mistaken meaning choice, since the "def2" node is still firing.

Thus, the major difference between this network and the one of Chapter 3 is that the out-of-class meanings go first, rather than the within-class meanings. This difference may be an artifact of connecting *all* of the meaning nodes inhibitorily. The meaning nodes corresponding to the inappropriate syntactic class definitions of the word are inhibited by the alternate meaning nodes. If we just connected the within-class meaning nodes inhibitorily, we would get the same prediction as in Chapter 3 modulo weight settings (again, there is a positive path between SHIP'S and CARD through DEF1-NOUN1-DEF2, so the behavior is not determined by the network structure as much as it is by the weights and unit functions). This is a problem with connectionist networks; the setting of weights often makes the networks "chameleons" of modelling behavior. This tends to put them in the class of descriptive, rather than explanatory, models. To the degree that the settings of weights can be independently motivated, the term "explanatory" becomes justifiable. As yet we have no independent motivation for the weight settings in this portion of the model.

Again, as in Chapter 3, the lexical access network is the decision machinery for disambiguation. Now, all that is needed is the feedback from

higher levels that drives this process.

This description of the word sense buffer leaves many unanswered questions. For example, in what sense is this a buffer? What is the mechanism for reading things in to the buffer? How are different buffer positions represented? These questions will be answered in the Implementation Details section. For the moment, the reader should simply know what the notation means: NOUN2, for example, represents "a Noun in the second buffer position", not "the second Noun to come along".

### 5.3.2. The Parser

The parsing network is generated by a LISP program that reads the grammar and a dictionary and outputs commands to the ISCON simulator to build the network. We will start by explaining at a high level what is generated by a production for one constituent with several productions. Then we will go into slightly more detail about how productions for a single constituent compete with each other, and how the binding nodes compete with each other. Following this, we will describe a sample run of the parser. The penultimate section covers implementation details.

#### Overview of the Network Generated by a Production

Figure 5.5 shows a production and a high level description of the network fragment generated by it. The nodes in Figure 5.5 represent networks in the implementation. For the purposes of exposition, however, this level of detail is more appropriate. It should be pointed out that the numbers on the constituents, e.g. NP1, are significant only as identifiers of which constituent recognizer copy is being used; copy 1 of the NP recognizers in this example.

The S recognizer is connected to two "production recognizers" which correspond to the two productions for an S constituent. These are not selected; they are a part of the constituent recognizer machinery for this copy of the S recognizer. These implement a simple voting scheme for comparing

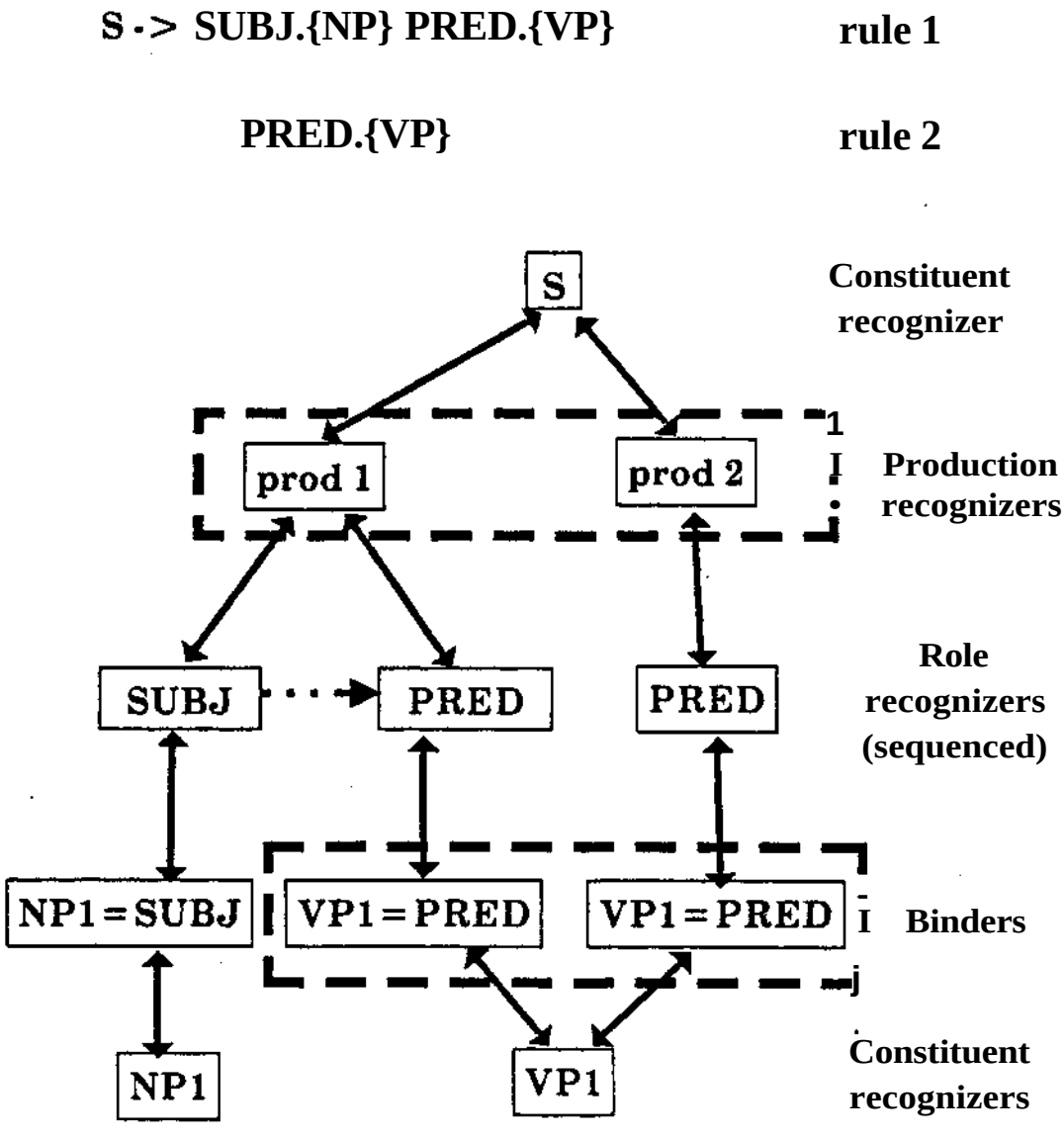



Figure 5.5. The network generated by a grammar rule.
Dashed rectangles indicate possible competitors.

evidence between the two productions. If the difference in the amount of evidence for the two productions is greater than a certain amount (called the *competition window)* the one with less evidence gives up.

The production recognizers are connected to "role recognizers", that sequence through the roles of the production. As evidence comes in for the Subject role for example, expectations are set up for the Predicate role, activating that recognizer. As mentioned above, since roles can be filled by different constituents depending on what production they are in, there have to be two Predicate recognizers, (n any case the control is different for the two; the Predicate recognizer in the first production has to wait for the Subject to be recognized.

The role recognizers operate by enabling the binder nodes for their role, and setting up expectations for the possible role-filling constituents. These constituents, in turn, are recognized by another set of constituent recognizers. These role-filling constituents are selected by the role recognizer from the set of inactive constituent recognizers of that class. As input comes in, it activates the constituents appropriate to it that are expected, and the binders for that constituent compete. If it was possible, for example, for both Predicate recognizers to be active (it isn't, given that the presence of a Subject would kill off production 2), the two VP1 = Predicate nodes would compete with one another. In practice, the competition would be between binders from a constituent to roles in different higher level constituents, such as ones with common prefixes. Binders get feedback from the production recognizers, so that a binding to a well satisfied production will tend to win over others.

This network is repeated for every production in the grammar. It "bottoms out" on buffered syntactic class nodes, for example NOUN2 or VERB3, which are directly connected "downwards" to the definition node for a lexical item, and "upwards" to binding nodes for roles such as Head in NP1 or Main in VP1.

Implicit in this discussion has been the fact that this parser starts out with top-down expectations, and merges them with bottom-up input. So, in the beginning, expectations are generated for everything that could start an S. This

means that the constituent recognizers are enabled and ready to go. Then the input that matches with those compete through the binders nodes. The binder nodes represent a merge of bottom-up input with top-down expectations. Whether or not we have a good algorithm for combining these types of evidence, we at least have a paradigm for testing different evidence combination functions.

The end result of a successful parse is that there is a unique binder that has won over its competition for every constituent, and one production that has won for every constituent, from the top level S on down. In the next sections we cover some of the aspects of this model in more detail.

**Production Competition**

The production competition is a simple voting scheme which favors longer productions over shorter ones, and as discussed earlier, roughly favors "late closure" (Frazier, 1979) of constituents. It works as follows: Every production gets two votes for every "filled" role. "Filled" here means that a binder for a constituent filling that role has won over all competing binders. This is communicated to the production competition network by a higher firing rate for that binder than is possible while it is in competition with other binders. Each production gets one vote for an unresolved role. "Unresolved" here means that there is evidence that a constituent for this role exists, that is, a binder for a role-filling constituent is firing, but it is still competing with other binders. The evidence rule for each production then is:

> if Max(votes for all productions) - MyVotes < Competition Window
>
> then continue competing;
>
> else lose;

In the current implementation, the competition window is 2 votes. Since the votes for this production are included in the Max of votes for all productions,

if this is the winning production, the left hand side of the inequality is zero, and the production continues. If a competing production is ahead by two votes, it kills this production. This evidence combination rule favors writing grammar productions with alternatives that are not greatly different in length; very long productions, if only partially satisfied, could win over shorter ones. However, this seems to be a natural restriction on grammars.

While particular rules like this are certainly arguable (this one is), the point is that we have a good framework for evaluating such rules; they are intrinsically interesting because they are completely *local* to the production competition network; there is no global interpreter making decisions about the "best fit" to the grammar. Thus this is a testbed for exploring decision algorithms for a parser that works in a completely distributed manner.

The careful reader will have noticed a problem with the rules as given. See Figure 5.6. Given that these two productions have a common prefix, and given input that matches the prefix, i.e. an unmodified NP, how does production 2 ever win? One answer is to add a "Closure" role that requires no filler (see Figure 5.7). Then, if no PP comes along, production 2 gets an extra 2 votes and wins. The problem is deciding when the Closure role should fire. Simply having it start firing after an arbitrary interval won't work, since different PP's will have different recognition latencies; they may arrive quickly, unfairly beating production 2 before the Closure role fires (the proper

```
Nbar->    Head.{NP} Mod.{PP}    (1)

          Head.{NP}             (2)
```

Figure 5.6 Production competition: the closure problem.

```
Nbar->    Head.{NP} Mod.{PP} Clos.{}    (1)

          Head.{NP} Clos.{}             (2)
```

Figure 5.7 The closure problem: A solution.

attachment isn't necessarily to this Nbar) or it may arrive too late to come to the rescue of production 1. One answer is to make the Closure role fire when either the PP is recognized, or input inconsistent with a PP is recognized. This can be computed directly from the grammar, using the "Follow set" (Aho & Ullman, 1977) (used in predictive parsers of computer languages) of the NP, in this example. Members of the Follow set inconsistent with the PP would cause the Closure role to fire. This solution is not currently implemented. By this mechanism, the shorter production will not win too soon. We can then depend on the semantic feedback to resolve the attachment of the PP, and the reader can check that our evidence rule will make the appropriate decision once this binding has been resolved (and not until then). While this appears plausible, we won't feel confident in it until we have tested it on an extensive grammar[2].

**Binders**

Binding nodes use a similar rule to the production competition one:

```
if (inhibition - support < window)
{ /* keep going */
  if (inhibition == 0 && support > criterion) then win;
  else continue;
}

else lose;
```

One minor difference from the production competition code is that the inhibition doesn't include what this node sends out. (Here we are using the absolute value of the inhibition, which is usually negative.) The competitors of a binder are determined from the rule that the same constituent can't be assigned to more than one role, and one role can't be filled by more than one constituent. The inhibition is the maximum of the input from these competitors. "Support" is the sum of bottom-up and top-down input. Top-down input comes from the production evidence network, and reflects how

[2]We would appreciate counterexamples, if the reader can come up with one.

well the **production** is doing. As more roles in a production get filled, the binders to those roles get more feedback. Thus if a binder for another role in the production wins, this is communicated indirectly to the other binders through increased feedback.

**Details**

There are several details that have been suppressed in this exposition; we return to them later in the "Implementation Details" section. Most of these stem from the use of a fixed network: Mechanisms had to be implemented to allow role recognizers to select constituent recognizers from a pool of them; the word sense buffer is of fixed length, although a similar selection mechanism might work here too; and just the existence of copies of constituent recognizers with the exact same control structures is not particularly palatable. However, recent advances in connectionist tools make the future look brighter. McClelland (1985) has developed a system called Connection Information Distribution (CID) which allows the storage of connection information in one central network (a *knowledge source*) to be "loaded" into a programmable buffer (like a Hearsay blackboard, Lesser & Erman, 1977) as needed. While the mapping of our system into the CID framework is not immediately obvious, the idea holds promise for avoiding many of the "fixed network" uglinesses.

### 5.4. An Example Run

This system was implemented on a VAX/750 running Franz Lisp and C; the network builder is coded in Lisp and feeds commands to the ISCON simulator which actually builds the network. This in turn is "compiled" into a representation suitable for a much faster network simulator written in C by Sumit Bandopadyay and Mark Fanty. The actual network for the simple example we will present contains over two hundred nodes and over a thousand connections. Hence, we will only give a high level description of the network's

behavior[3].

Given the sentence *he cut the roll* there are two cases of lexical syntactic ambiguity, *cut* and *roll* There are no interesting closure problems, as any difference between this and *he cufr* can be handled by using "period" as a lexical item. We simulate reading this sentence by activating each lexical item every 30 steps of the simulation. These are then "read in" to the definition buffer described earlier, and after 7 more steps, the syntactic class and meaning nodes are activated (these and the other nodes in the buffer accumulate activation slowly). About the same time that PRO is activated in the buffer, an NP is expected by the Subject role of the S production. Since "he" is unambiguous, PRO has no competitors and gets highly active quickly. It then rapidly becomes bound to the Head role in the first NP (by iteration 16) since there is no competition for it.

The network then expects a VP, and the two productions for a VP (with and without a Direct Object) set up expectations for a VERB, When "cut"'s features (NOUN2 and VERB2, the "2" indicating buffer position) become activated at clock step 38, they inhibit one another, driving each other below threshold Since this cuts off the inhibition, they rise up again, like flickering bits, but now feedback from the binder for VERB2 to the Main Verb role in the VP gives extra support to the VERB2 node, allowing it to remain above threshold while NOUN2 doesn't, resulting in a win for VERB2. Thus, "cut" has been disambiguated as a verb. By a few steps later, the node corresponding to the "meaning" of "cut" as a noun also loses.

After this is propagated up the network, expectations are set up for either a "period" or a Direct Object which can be filled by an NP. The Direct Object role recognizer selects NP2 (the next unused NP recognizer) which sets

[3]Eye witness reports that this really works can be obtained from my friends.

[4]Recall that we haven't implemented features. In particular, there are no verb subcategorization features, so if we wanted to represent that *cut* is different from *left* we would need to encode them as different syntactic classes in the current parser. Since we haven't, *he cut* is acceptable to our parser.

up expectations for either a DEL NOUN or PRO. After "the" has been processed, the NP2 recognizer is only expecting a NOUN, and so when "roll" comes in, it is quickly disambiguated. Eventually the production corresponding to a VP with a Direct Object wins, and the resulting stable coalition represents this parse with the appropriate binding nodes in a highly active state.

Thus our parser has disambiguated the two ambiguous lexical items on the basis of their "fit" into the developing parse tree. Anything incompatible with the expectations developed at the "frontier" of the developing tree was quickly extinguished. If there had been more structural ambiguity, lexical items compatible with either structure would have remained ambiguous until some production won over another. While this may seem implausible, it is compatible with the results of Hudson and Tanenhaus (1984).

## 5*5. Implementation Details

This section is intended for those die-hards that want to see how it *really* works. We first answer the previously posed questions about the word sense buffer, then detail the operation of the binding nodes, show the actual networks corresponding to constituent and role recognition, show how copies are selected from the recognizer pool, and finally detail the production competition network.

### 5-5.1. The Word Sense Buffer

The word sense buffer must be activated by lexical items and activate their "definitions" in sequential locations. The way this was done requires that only one lexical item be active at a time; while psychologically implausible, it was expedient In every buffer position, there is a copy of every word's definition units (the "def nodes of Figure 5.4) and syntactic and semantic feature units. All of the definition nodes are self-stimulating, so they keep themselves going once activated, until inhibited by alternate definitions with

greater feedback from the feature units. Syntactic feature nodes are shared within a buffer position, so every definition node in position 3 for example, that corresponds to a Noun, is connected to N0UN3. Thus the interface to the rest of the grammar is restricted to these few syntactic class nodes in every position.

All that is necessary to read a word into a buffer position is to activate the "def" nodes in that position corresponding to its various definitions. There are three nodes per buffer position that control the sequencing (see Figure 5.8). The "enable" unit enables all of the definition (and the feedback) nodes in a buffer; without input from the enable unit, the definition nodes will not respond to input from the lexical nodes. The feedback unit is of the same type as the definition nodes, and serves as an indicator that the definition node is

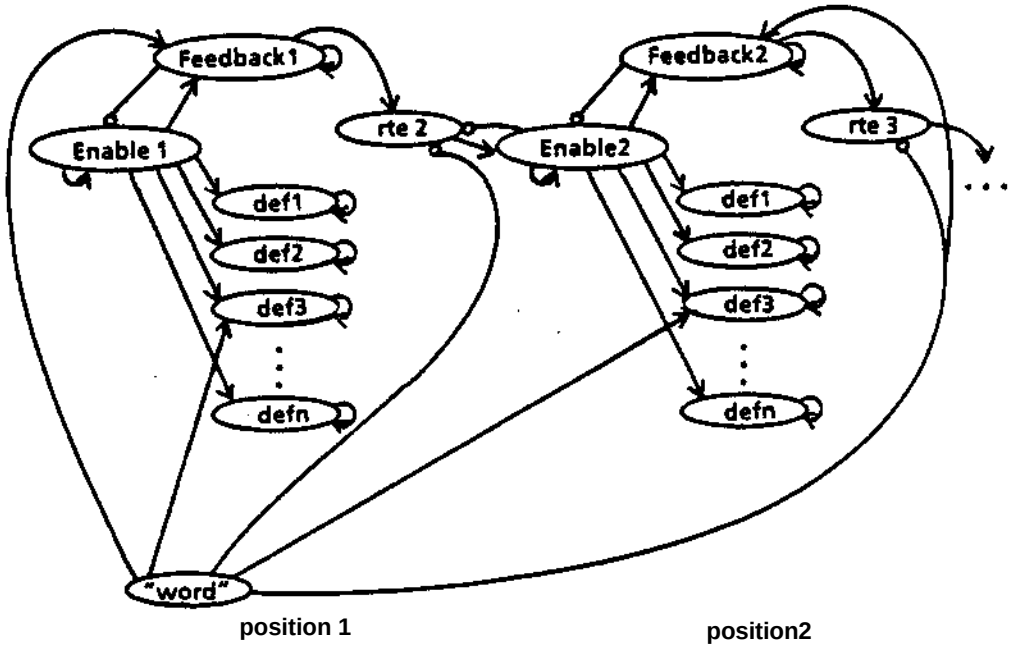



Figure 5.8. Sequence control of buffer positions.

firing. All of the lexical units are connected to each of the feedback nodes. This is logically unnecessary, as the definition nodes can serve the same purpose. The feedback node turns off the enable node, and gives input to the "ready-to enable" (rte) node for the next position. The rte node is inhibited by the lexical units. So the semantics of the rte unit is "not word." As long as a lexical item is firing, the rte unit isn't. When the lexical unit decays below threshold, the rte unit quickly begins firing (it has been getting input from the feedback node all this time), activating the enable unit for the next buffer position. The enable unit is self-stimulating, so it continues to fire until another word comes in and the feedback unit for that buffer position turns it off. It also "bites the hand that feeds it", turning off the rte unit that started it up in the first place. This is how the buffer positions are sequenced. It would be desirable to have a mechanism for this that doesn't require the implausible assumption that only one word unit at a time fires.

### 5.5.2. Binding Spaces

The "binding space" for a constituent is the set of roles that it can have. Since the roles are in many copies of constituent recognizers (see the next section), there has to be a controlled way in which the binding units for the various copies of the parent roles interact. An early problem with the implementation was that after a binding unit to a particular role had become active, when another constituent of the same type came along with the same role for the already bound constituent (for example, another NP copy for the Direct Object when, say "he" was already bound to the Head role in the Subject NP) the binder for the new role became active. Thus we connect all of the binders to copies of the same constituent in a left-to-right WTA, with a strong inhibitory weight, so that if a constituent becomes bound to a role in constituent copy i, it can't become bound to a role in constituent copy $i+1$. This is only a stop-gap measure, since the left-to-right WTA's are only for binders to constituents of the same type (NP, VP, etc.). If a constituent of

another type tried to bind the already-bound constituent, the same problem could arise. The problem stems from the function used for binder nodes, which has since been changed to use state transitions that prevent new competitors from activating once a binder has won. However, since we haven't tested it without the left-to-right WTA's, we don't want to make unwarranted claims. The new binder unit function causes any binder that is receiving inhibition and is not enabled (expected) to enter the "lost" state. Once it is in this state, it refrains from ever becoming active as long as there is still inhibitory input. This just implements the semantics that an "already bound" constituent should not be bound to anything else.

The fundamental problem is that there is no way in the current system of marking the span of the input that the constituent covers. The order is determined solely from who is active when. This could also lead to constituent recognizers "skipping over" portions of the input. The design (but not the implementation) thus now incorporates a special binder for every role that becomes activated by input incompatible with the role, which kills the production. Another solution suggested by McClelland (personal communication) is to use recognizer copies that only apply to specific portions of the buffer, similar to his Trace model of speech recognition (Elman & McClelland, 1984). It would be interesting to determine whether such a (relatively drastic in the number of copies) measure is necessary to parse most sentences, or whether a system that has less of a strict ordering such as the present design is sufficient. We intend to explore these questions in future research.

A second problem is setting the parameters in the binding node WTAs. There is a fine balance between allowing competition and squashing it. For competitors that *should* be allowed, the weights have to be low enough to allow each other to keep going when the evidence is not overwhelming for one binder, yet high enough to kill off the competition when the evidence for one binder is strong. "Weight twiddling" is an unsavory affair, and it appears that

what should replace it is a better theory of combining evidence for distributed decision making. The way decisions are made in this implementation is rather draconian; once a unit has lost, it has lost forever. We believe that using probabilistic units of the type used by Hinton and Sejnowski (1983) would allow a more forgiving decision mechanism; units that have low evidence, once "squashed", would have a chance to recover if better evidence arrived[5].

The function computed by the binding nodes is shown in Figure 5.9. The basic idea is that if the difference between the evidence for this binder and the evidence for its competitors is not larger than the user settable competition window, it continues to compete. Otherwise, it loses. If it has lost, and receives no inhibition, its competition has lost as well, and it returns to the initial state. The evidence is a weighted sum of the bottom-up and top-down support.

### 5.5.3. Constituent and Role Recognizers

The networks involved in recognizing the Subject of an S are shown in Figure 5.10. Expectations are started by activating the "Expectation" node for a constituent recognizer. This has to start somewhere, and so we give the "Expect-S" unit a non-zero resting potential, thus expectations originate with it[6]. We will first trace the flow of top-down expectations through one layer of the network, and then follow the effects of bottom-up activation from a recognized constituent.

Once the an "Expect" node is activated, it remains firing until either the companion feedback node is activated, which turns off the expectation node, or the role node that started the constituent recognizer is itself turned off because its production lost (there is none for the "Expect-S" node). The Expect-S node has two effects: It enables the Feedback-S node, and starts the recognition

[5]Wait! I've just been handed a bulletin! As this thesis was being placed in final form, I received a paper which takes this very approach. (Selman & Hirst, 1985). A few remarks on this paper were included in Chapter 2.

[6]In a grammar with embedded S's, it is useful to have a surrounding production in which the top-level S fills the

```
support = 0.6 * bottomup + 0.4 * topdown;
switch (state){
        case q0: if (enabled and there is bottom-up input and
                        (|inhibition| - support < competition_window)) then
                        {
                         state = competing;
                         return(support);
                        };
                        /* if we got here, then either: not enabled,        */
                        /* no bottomup input, or not enough support.        */
                        /* If anyone is competing, we should lose.          */
                        if (inhibition) then state = lost;
                        return(0.0);

        case competing: if (enabled and there is bottomup input and
                          (|inhibition| - support < competition_window)) then
                        {
                         if (no inhibition and
                                 bottomup input >= win_threshold) then
                         state = winner;
                         return(support);
                        }
                        /* if we got here, then either: not enabled,        */
                        /* no bottomup input, or not enough support.        */
                        state = lost;
                        return(0.0);

        case winner:    return(0.8);

        case lost:      if (no inhibition) then state = q0;
                        return(0.0);
}
```

Figure 5.9: The function computed by binding units.

role of "Matrix".

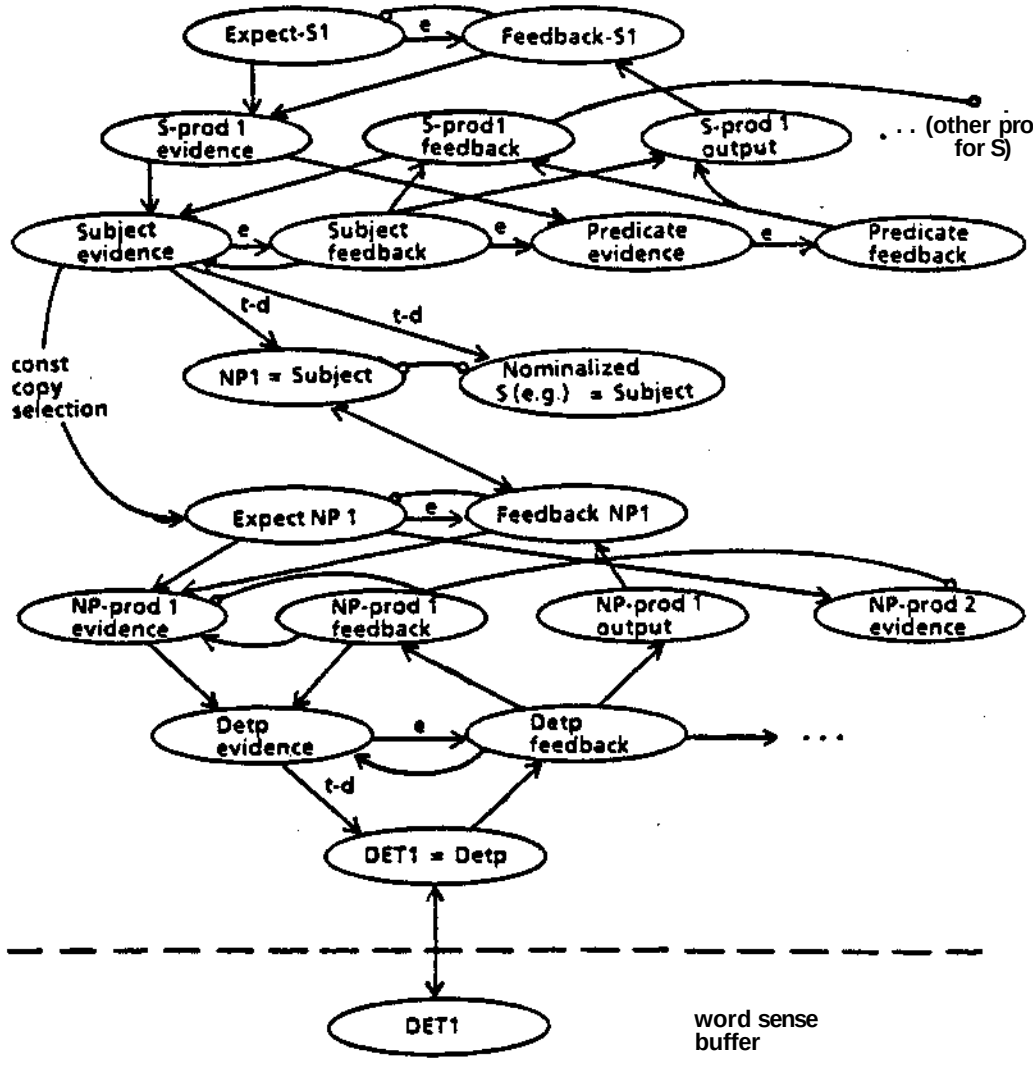



Figure 5.10. Recognition of an S.

networks for each of its productions. It does this by stimulating the production evidence nodes. We will go into this in detail in the Production Competition Network section. As long as the constituent is expected, the production evidence nodes get this input. They, in turn, enable the feedback and output nodes for their production and start the sequence of role recognizers for their

production. Starting the role recognizers consists of activating the first role evidence node, in this case the Subject evidence node of the first production for an S. This has several effects: 1) a constituent recognizer for each constituent that can fill this role (e.g., NP1 for an NP to fill the Subject role) is selected by the mechanism described in the next section, in effect activating its expectation node (Expect-NPl), 2) binders for those constituents to this role are enabled, and 3) the feedback node for this role is enabled (Feedback-Subject). Thus expectations cascade down the left side of Figure 5.10, until a DET is expected This means that binders from all DETs in the buffer to the DetPhrase role are enabled.

Now, we will suppose that a "the" comes in, activating DET1, and follow the effects of bottom-up activation. (Since all of this machinery is enabled top-down, nothing happens unless the constituent is expected.) First, all of the binding nodes for the DET1 that are enabled become active. The one for the first role in production 1 of NP1 (DET1 = DetPhrase) sends activation to the feedback node for the DetPhrase. This becomes active, and has two effects. It sends its activation up to the feedback and output nodes for this production (production 1 of NP1) and enables the next role expectation node for this production (Head in NP1) which begins firing immediately since it is getting top-down input from the production evidence node. Thus the recognition process for the next constituent in the production is begun. The second effect of the role feedback node is to give evidence to the production feedback and output nodes (they collect the bottom-up "votes" for the production).

The production feedback node then sends the evidence to the evidence node for production 1 of NP1, which then decides this production should continue competing, and to the evidence node for production 2, which turns itself off. Also, the feedback node for production 1 of NP1 sends activation back down to the DetPhrase role expectation node. This is the path for top-down feedback to the binders. The circulation is: binders -> role feedback -> production feedback -> role expectation -> binders. Since all of the role

feedback nodes for this production (DetPhrase and Head) feed the production feedback node, this is a way of having the contextual effect of a winning production increase the probability of attachments to itself.

The output node for production 1 of NP1 sends its evidence up to the feedback node for NPL The NP1 feedback node takes the maximum of the evidence from the various productions for NP1, and sends this up to its binders. Currently, the feedback node is thresholded and starts sending output as soon as about a third of a production is recognized (that is, one third of the roles are filled, counting the Closure role). Another possibility would be to gate the output of the constituent feedback node by the last role of the production, so that it would not start sending output until an entire production is recognized- We would like to explore the effects of different schemes here further. This one was chosen because it speeds the spread of activation, and allows higher levels in the grammar to work with partial results. Another effect of the constituent feedback node firing is the inhibition of its partner "expect" node. The feedback node must then take over some of the function of the expect node, and so is connected to the production evidence nodes. The expect node must be inhibited as part of the constituent copy selection process, described next.

### 5-5.4- Constituent Copy Selection

When a role recognizer needs a constituent recognizer, it has to select one from the pool of available ones. It does this through the network pictured in Figure 5.11. The role evidence node activates all of the selection nodes. The selection nodes are arranged in a strict, locking WTA, similar to the one described in (Feldman & Ballard, 1982). Once a selection node wins, it stays on forever (unless the parent role dies due to its production losing). The selection nodes, in turn, activate the "Expect" nodes for the constituent copies. Any constituents that have been recognized will have their "Expect" node inhibited by their "Feedback" node as described above. Thus, the selection

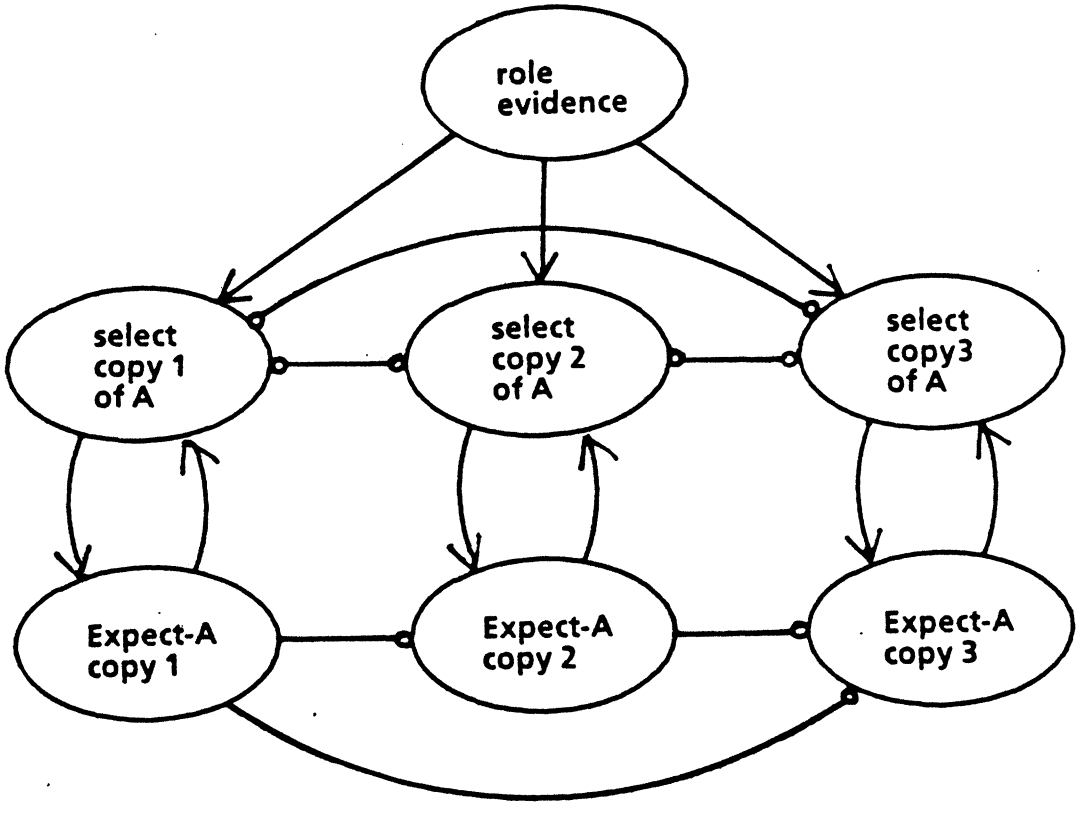



Figure 5.11. Constituent Copy selection.

node for this copy will not get feedback from its "Expect" node, and will lose the competition forthwith. The "Expect" nodes are arranged in a left-to-right WTA, so the "leftmost" constituent copy that has not been recognized yet will inhibit the ones to its right. This will result in more feedback to its selection node, which will then kill the others, and lock on as a permanent pathway between the role expectation node and the constituent expectation node. One thing to notice about this is that any other role that needs this particular constituent can select the same recognizer, as long as the constituent hasn't been recognized yet.

### 5-5,5. The Production Competition Network

Figure 5.12 shows the network which implements the production competition and its links to the outside world. As discussed above, this network becomes activated by a constituent expectation node, which activates all of the production evidence nodes. The functions computed by the nodes in this network are shown in Figure 5.13. The feedback nodes collect the bottom-up votes for this production from the binder nodes. Based on the input from each binder, they decide whether it has won or not (the binder nodes put out a certain maximum amount if and only if they have won), convert that to two votes for a winner and one vote for a binder still

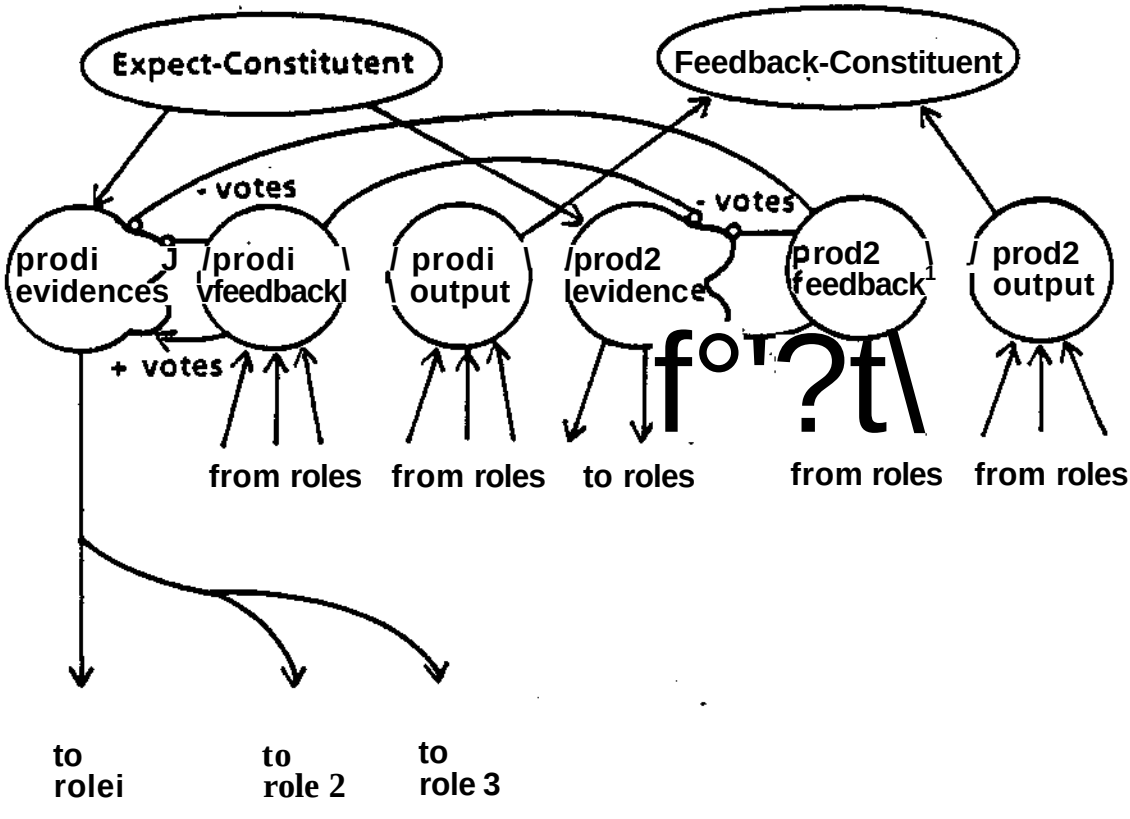



Figure 5.12. The production competition network.

```
/* Rule Evidence node: Outputs sum of site input */
{
 input = topdown + bottomup + inhibition;
 if (there is topdown input) then return(input)
 else return (0);
}

/* Rule Feedback node: Outputs sum of votes */
if (enabled) then
{
 votes = 0;
 foreach(input)
        if(input > 0) then
                if (input > binder_win_threshold) then
                        votes = votes + .2;
                else votes = votes + .1;
 return (votes)
}
else return (0);

/* Rule Output node: Outputs normalized average of votes */
if (enabled) then
{
 votes = 0;
 foreach(input)
        if(input > 0) then
                if (input > binder_win_threshold) then
                        votes = votes + .2;
                else votes = votes + .1;
 return ((votes/number_of_inputs)*5);    /* Average votes, */
}                                        /* Normalized to [0,1] */
 else return (0);
```

Figure 5.13. Pseudocode for Rule Competition Network nodes.

competing, and add it all up. It then sends this to every production evidence

node in the network as inhibition, and to its own evidence node as excitation. Thus it cancels out its inhibition on its own production. This is not a no-op, since the evidence node takes the maximum of the inhibition from all productions. The production evidence nodes add the input from the feedback node, a fixed value for top-down input (a user settable parameter, the "competition window") and subtract the maximum of the inhibition. The result is always a value between 0 and the competition window (the potential has a floor of 0). When the potential goes to 0, everything enabled by the evidence node goes off, killing the production.

The production output node computes the same function as the feedback node, except it scales the votes according to the length of the production. Thus the competition between productions is based on an absolute scale of the number of roles filled but the output to the constituent feedback node is normalized. The rationale for this is that if the competition were scaled as well, productions of length two (three, counting closure) quickly get an unfair advantage over productions of length three, for example. The absolute scale is also easier to control, since with normalized production competition, it is hard to determine a useful "competition window", since the difference between the amount of weight accorded to any votes for a role varies with the length of the productions[7]. On the other hand, we decided that the evidence that goes to the constituent feedback node should be scaled, so that constituents of different lengths could compete on an equal footing for attachments.

## 5.6. Conclusions

We have presented a parsing model which has the following features:

(1) **Completely distributed.** The decisions between alternatives are made solely on the basis of information local to each processor. There is no global interpreter that can view the whole tree to make attachment decisions.

[7]Selman and Hirst (1985) use the scaled approach. Early advantages for shorter productions are apparently less of a problem for them because of the flexibility of the probabilistically updated units.

Global coherence is maintained by mutual constraints between competing and cooperating hypotheses at neighboring levels of the system.

(2) **Massively parallel.** All units run in parallel, and there are a lot of them! The parallelism is at a much finer granularity than most previous systems.

(3) **Automatically generated from a grammar.** The only other system of this type that is generated from a grammar is that of Pollack and Waltz (1985). However, in their system, the network is generated *in response to the input*, and is tailored specifically to the input. The system reported here is generated once from the grammar, and is then ready to accept whatever input is dictated by the grammar.

(4) **Does not use symbol passing.** Most AI systems build large symbol structures and pass them around. This appears to be neurologically implausible (at least, improbable) while the spreading activation used by the connectionist paradigm is more in line with what appears to be the mechanisms used by the brain.

(5) **Uses a fixed network.** Again, the use of a fixed network lends neurological plausibility to the model. Furthermore, an often heard argument against connectionist networks is that since they are fixed, they are incapable of "X", for some X. We have shown that parsing, at least, is not equal to X.

(6) **Minimal Attachment "falls out" of the mechanism's operation.** Many systems "explain" Minimal Attachment as some parameter setting in their model. In this model, Minimal Attachment is a property of the way computation is carried out, i.e., by spreading activation through a grammar system. This appears to be a more fundamental explanation.

# CHAPTER 6

# IMPLICATIONS FOR APHASIA

## 6.1. Introduction

One test of the validity of the model presented in the previous chapters, if it is truly neurologically plausible, is to evaluate its adequacy at accounting for neurolinguistic data. This is the goal of this chapter. After reviewing some evidence regarding breakdowns in the language system, we try to account for some of the data in terms of "lesions" to our model. The model is shown to be adequate for explaining some of the overall effects.

The major finding that is consistent with our model is the (controversial) unintuitive result that there exist patients who appear to be able to make grammaticality judgements without the ability to use grammatical information in understanding. This is explained in the model as a loss of the system that maps the constraints between syntactic and semantic attachments. The model then makes some predictions based on the remaining pathway between the two systems in the word sense buffer.

Lexical disorders are also considered as they relate to our model of the lexicon. Implications are drawn for our model and extensions to it.

It is interesting to note that models such as the one presented in this thesis are "lesionable" without reprogramming. This makes them interesting testbeds for theories of aphasia. Most AI models require considerable programming effort in order to "remove" routines that correspond to hypothesized deficits. The fact that connectionist models still "run" when parts of them are missing considerably reduces this effort.

## 6.2. Neurolinguistic Evidence

### 6.2.1. Introduction

Saffran (1982) has argued that much can be learned about the language processor by studying the language behavior of aphasics. "The selectional impairment of language functions can help to reveal the componential structure of the language system." (Saffran, 1982, p.317). In addition, the error data used by many psycholinguists to analyze the system (Garrett, 1980) are plentiful in aphasic patients. Finally, the findings relating to aphasic patients are sometimes so unexpected as to require alterations of existing theories of the structure of the language system.

A brief characterization of three types of aphasia are in order. It should be pointed out, however, that (as Saffran advocates) a much finer grained classification of aphasic disorders based on detailed linguistic analyses is needed. The data from patients within these classes can vary wildly, simply because the classifications are much too broad to be useful.

(1) *Broca's Aphasia.* These patients generally display an inability to produce fluent speech, characterized by the omission of function words and bound morphemes, labored speech, "serial naming" in severe cases, and often omissions of the verbal element. In the last ten years, studies have shown that this deficit is paralleled in comprehension, notably by an inability to use syntactic cues to comprehension.

(2) *Wernicke's Aphasia.* This syndrome is characterized by the fluent production of speech, which is, however, devoid of content and error prone. Often these patients will display *paraphasias*, in which semantically related words are substituted for the target word (called *semantic paraphasias*), or phonemic substitutions which result in gibberish which nevertheless follows the phonemic rules of their native language. In contrast to Broca's aphasics, Wernicke's aphasics will often produce function words and correct syntactic frames, but the content words will be

scrambled.

(3) *Anomia.* Anomia means "word-finding difficulty". Anomics have not lost the words, they just have trouble accessing them. Often these patients will be able to describe the word they are seeking, and find it without realizing it. (For example, in searching for the word *comb*, they may say, "It's something you use to comb your hair, but I can't think of it...").

In this discussion, we consider disorders of two aspects of the linguistic system as they relate to our model: Lexical disorders and agrammatism. Rather than doing a wide ranging review of the literature as in the previous section, we will base our discussion on review papers and a small number of recent studies that seem particularly relevant to our work.

### 6.2.2. Lexical Disorders

We will follow the review of Buckingham (1981) on lexical and semantic aspects of aphasia. This data is particularly relevant to models of the lexicon. It is necessary to begin with a brief review of semantic feature systems, since they have clinical correlates. *Selectional restrictions* are rules that subcategorize verbs on the basis of features for their subjects and objects. For example, a subject that is human would be marked [+Human]. This is then picked up by the verb as +[+Human_], indicating the subject is human. In some cases, this will select one sense of the verb over another, especially when combined with restrictions from other verb arguments. For example, in *The slug operated the vending machine* (Hirst 1984), the sense of "operate" is selected by the restrictions on the cases of the different meanings of "operate".

Two ways of organizing features (which appear to have psychological and neurological correlates (Keil, 1984; Goodglass and Baker, 1976) are along *paradigmatic*, or hierarchical relationships, and *syntagmatic* or "horizontal" relationships. Paradigmatic relationships are the basis of the familiar IS-A hierarchies of Artificial Intelligence. They are usually divided into semantic *fields*, such as color, kinship, spatial relationships, temporal relationships, body

parts, etc. An important concept here is *minimal contrastive sets.* This refers to elements that are immediate children of the same superordinate type (minimal), and yet are mutually exclusive (contrastive), i.e., one and only one word in the set can apply to any given particular. For example, *chair* and *sofa* are both elements of the set of furniture, but a chair is not a sofa and vice versa. These are also sometimes called *contrast coordinates.*

Paradigmatic features can be further divided into so called *defining* and *characteristic* features. Defining features are more abstract and general, such as "Human", "Male", "Artifact" and so on. Characteristic (or *residual*) features are more ephemeral, such as "married", "ferocious", etc. A linguistic test which separates these is whether they survive under negation. For example, a "husband" has features +[Human], +[Male], and +[married] The presuppositions about the referent of "husband" in "that person is not a husband" are that the person is +[Male], while the "married" feature switches to -[married]. Basically, characteristic features are more malleable. Kiel (1984) did some interesting experiments in which he told children a story about something in which all of its characteristic features changed until it resembled in every way a member of a different category (for example, a raccoon that was changed to look like a wind-up toy), and the children still maintained that it was a raccoon (if they were old enough).

Syntagmatic relationships are based on contiguity of three types[1]: *Predication*, such as adjective noun relationships ("red ball") copulative sentences ("Joe is a lousy lover"), object-location ("the boat is in the water"), or functional relationships ("watches are for telling time"); *Coexistence*, such as "peanut butter and jelly", "bread and butter", etc. and *synecdoche* which refers to using a part to stand for a whole, as in "wheels" for "car", or "tube" for "TV".

---

[1]According to Buckingham. The classification of these three types of relationships under one root seems strained.

The reason for laboring through these definitions is that they have clinical correlates. Loosely speaking, Broca's aphasics become insensitive to syntagmatic relationships, while Wernicke's aphasics lose sensitivity to paradigmatic relationships. It is interesting to consider these phenomena in terms of their implications for models of the lexicon. The major implication to be drawn is that these distinctions between paradigmatic and syntagmatic must have cognitive correlates, and that there are separate anatomical structures and systems in which they are represented.

Fortunately, there are more specific conclusions to be drawn. An interesting fact about semantic paraphasias, where a semantically related word is substituted for the intended word, is that the substitution is often a contrast coordinate. If we assume that the lexicon is shared between production and comprehension (more on this later), then contrast coordinates must be stored "closely" to one another. We take this as evidence that the natural hierarchical representation of paradigmatic relationships is reflected to some degree in the actual cognitive representation of the lexicon. In accessing the word to be spoken, these aphasics appear to be accessing the lowest level category of the word they want, but then the selection machinery breaks down. We hypothesize that (as mentioned w.r.t. synonyms in the semantic priming section) there is a Winner Take All network for each coordinate set in the hierarchy of concepts that is enabled by the production system[2]. It is the control of these WTA's that appears to be disrupted in semantic paraphasias.

Often the exchanges that are seen are very similar to the results of word association tests. Besides contrast coordinates, antonyms are also substituted. Like contrast coordinates, antonyms often share a superordinate category (for example, "hot" and "cold" are both temperatures). Finally, it is not only coordinates that are exchanged. Superordinates are also substituted.

---

[2]A type of Winner Take All network that can be controlled in this way is described in Shastri & Feldman (1984). Basically, a separate node is connected to all the units in the WTA and computes their maximum output, which it sends back to all units in the WTA as inhibition. If this node is controlled, then the WTA can be turned on or off.

Further evidence for the hierarchical nature of the representation is the existence of category-specific anomia. These patients display word-finding problems for only certain semantic categories. For example, category anomias have been found for colors and body-parts. This suggests that the route to a word is through initially accessing the category (as hypothesized above), and second, that categories are accessed as a unit, or at least through some specific pathway that has been disrupted Finally, there are also errors that substitute totally unrelated words. This leads to the question as to whether these patients have disrupted their access mechanism at the highest levels of the hierarchy, and are putting out whatever word becomes most activated (It should be pointed out that these patients with paraphasias, mostly Wernicke's aphasics, often don't realize anything is wrong with their speech. Therefore they can't be monitoring it for errors.)

All of these data derive necessarily from the production system. On the comprehension side, perhaps more relevant to a model of comprehension, there are parallel deficits. Returning to an earlier point, Saffran (1982) argued that if we find that one kind of deficit is *always* accompanied by another kind, then we can almost safely infer that they share some common component. The "almost" caveat follows from the observation that the reason two deficits always co-occur may be due to the neurological systems they use being simply close together physically, rather than being identical. With that in mind, we will blithely accept for now the hypothesis that the data support a common subsystem, the lexicon, for use both in production and comprehension.

The parallel deficits in comprehension are best described (still following Buckingham's discussion here) in a study by Goodglass & Baker (1976). They were following up some earlier work by Zurif et al. (1974) who found significant differences between Wernicke's and Broca's aphasics (and normal controls) w.r.t. the way they would group words they considered "similar" together. Wernicke's tended to group things along syntagmatic lines (given *mother*, *husband*, and *cook*, one patient uttered "My mother is a good cook" as

he grouped *mother* and *cook* together) while Broca's showed sensitivity to the +[Human] feature. The Broca's aphasics, though, did tend to group things outside of the +[Human] class along residual feature lines rather than paradigmatic features. Goodglass & Baker tested aphasic and normal sensitivity to paradigmatic and syntagmatic relationships between words, including superordinate, contrast coordinate, and functional associates of the words. First, they pre-tested the group on a set of 16 pictures of objects that corresponded to 8 high frequency and 8 low frequency words, to see if they could name the objects. Then, they showed the pictures and played a tape of related and unrelated words, asking the subjects to respond if the word reminded them of the pictured object. First of all, they found that everyone did well when the word was the name of the picture (identity). However, normals and Broca's aphasics did very poorly with contrast coordinates, while Wernicke's did fairly well (compared to their performance overall, which was poor). We suggest that, following the discussion above, in the normals' and Broca's aphasics' mental lexicons there is mutual inhibition between contrast coordinates that the Wernicke's aphasics appear to have lost.

Contrary to expectation, the Wernicke's aphasics did not show sensitivity to functional associates, a syntagmatic relationship. However, they basically did poorly on everything. One significant result was their 50% higher error rate on pictures they couldn't name, suggesting a correlation between naming problems and the associative structure of the lexicon. It should be noted that Broca's aphasics did as well on pictures they could name as ones they could not. Other research has indicated that their mental lexicons are intact, so their problem appears to be more one of access rather than disorganization.

In conclusion, Buckingham delineates three explanations for word finding problems. He obviously favors the first, which says that word finding difficulty is a result of lexical associative disruptions (i.e., the structure of the lexicon has been disrupted). The other two are not as popular; the *disconnection* explanation is that the connection in the brain between the lexicon and visual

associates is broken. This has little to recommend it; it does not explain the apparent disruption *within* the lexicon, nor the ability of aphasics to recognize the name of a picture when they hear it. The last is the "neurodynamic" explanation of Luria (1974). It is similar to the one we have alluded to in the above discussion; he posits that a general inhibition mechanism has been disrupted. In his theory, lexical (production) processing is comprised of two phases, a general excitation phase, in which a multidimensional matrix of associated lexical items are stimulated, followed by an inhibitory phase in which attention is directed to the proper word and all others are inhibited. It is the second, inhibitory phase which has been disrupted, according to Luria. The subject has not only the target word, but all associated words activated, and is unable to select from them. This would explain why associated words are often exchanged with the desired word. However, according to Buckingham, this explanation does not appear to fit with the results of Goodglass & Baker, where it is apparent that the associated words are not activated, since the subject does not respond to them. However, it should be said in defense of Luria that if patients often substitute associated words for the target word, then Goodglass & Baker's results are not assessing the associations available during production. It is dangerous to generalize in this way, though, since we don't know the specific production behavior of the patients they used. It is important, as Saffran (1982) points out, to be more specific linguistically about the type of deficit each patient has.

Our explanation is similar to Luria's, but takes the structure of the lexicon into account We hypothesize that the normal subject accesses the semantic section (or field) of the lexicon for the word to produce, and moves down the hierarchy, inhibiting alternatives not wanted until reaching a single word. We hypothesize that at least *some* patients with semantic paraphasia are often accessing the section of the lexicon that they want, but can't inhibit alternatives. If these patients correspond to the ones in Goodglass and Baker's study who did not respond well to any associates, then we would argue as

above, that their method is not assessing the associations engaged by production. Of course, the other explanation cannot be ruled out; that is, in severe cases, the structure itself may be disrupted.

A final item in Buckingham's review appears to support our explanation over his. Weigel-Crump & Koenigsknecht (1973) (hereafter, W-C& K) report on the results of naming therapy on anomic patients. They chose 40 words the patients could not name initially such that they were equally divided among five superordinate semantic categories. The patients were then drilled on half of the words, but words from one category (foods) were left out. After therapy, the patients made significant improvements in naming both drilled and undrilled items, including those from the undrilled food category. Buckingham claims that this does not disprove the lexical structure disruption account. His explanation is that their therapy strengthened the associative *bonds* for the words in the drilled category, which "by fiat", delineated the undrilled words in that category as well. To account for the improvement in the undrilled category, he generalizes this explanation to say that since the therapy concentrated on four of the categories, then "by fiat" the fifth was drawn into sharper distinction from the rest, making it easier to label. One wonders whether, if W-C&K had tested other categories, they might not have found that naming had improved on them as well. If not, why would "food" in particular be improved? This would weaken Buckingham's argument. It is one thing to say that the therapy brought into sharper focus the elements of the category that was drilled. It is another to say that drilling on four categories brings *all* other categories into sharper distinction. Rather, it appears that the control of a general *access* mechanism, such as the successive inhibitory selection of more specific levels, has been improved. Further research should be done to distinguish these two explanations.

### 6.2.3. Syntactic Disorders

This section concentrates on the results of a small group of researchers whose work seems particularly relevant to our model. We will follow to some extent the review of this work in Saffran (1982). As mentioned earlier, it is possible with aphasics to view subsystems in relative isolation. In particular, one could interpret agrammatic aphasics' behavior as reflecting the operation of the semantic interpreter without the aid of a syntactic parser. While this is perhaps an overgeneralization, it is a useful one. Within the group of aphasics identified diagnostically as Broca's aphasics there is considerable variation in syntactic deficits. We will concentrate on a subset that displays specific grammatic impairments termed *agrammatic* aphasics. The *explanation* of those difficulties is still hotly debated, but the resultant behavior is generally agreed upon (except where it does not agree with someone's explanatory theory). It is this behavior and its interpretation as a reflection of the semantic system in isolation that we are interested in. (Of course, we will not be able to avoid discussion of explanations, since we want to throw in some of our own!)

We will give a brief review of the three major explanations of agrammatic deficits in production first, as a vehicle for discussing previous work. While our model is not a production model, this data is of interest because there appear to be parallel deficits in comprehension, and this will prime the reader for that discussion. These explanations each concentrate on implicating a particular level of the system (see Figure 4.1 for reference). The first is that the *sound* level is impaired (Lenneberg 1973). The argument here concentrates on the observation that Broca's aphasics appear to have trouble initiating sentences with unstressed words; thus, since function words are most often unstressed, they are omitted most often at sentence-initial positions (cf. Gleason et al., 1975). This explanation says that there is a more complex representation than what we observe, because the patient simply cannot vocalize the unstressed elements well. Vocalization problems notwithstanding, the data of Saffran et al. (1980a), discussed below, indicates that the underlying

structure is *not* correct.

A second level that has been implicated is the *positional* level (Kean, 1979; Berndt and Caramazza 1980). According to some theories (Garrett, 1980), at this level the syntactic frame has been specified, i.e., the grammatical morphemes in order with slots for content words. The deficit at this level is thought to be either one of disrupted access to the grammatical morphemes, or more generally a disruption of the syntactic processes that specify these frames.

However, Saffran and her colleagues (Saffran et al. 1980a) have identified a more serious problem. Most theories assume that word order, at least, is preserved in these syndromes. Saffran et al. found that agrammatics have serious difficulty in producing words in the right order when the major roles in the situation described are reversible. For example, in a situation such as a man running to a woman, where either person could potentially fill either role, runner or runnee, the subjects would often reverse subject/object order.

> "These results suggest that the capacity to map relational roles onto Noun-Verb-Noun sequences is seriously impaired in agrammatic aphasics. In lieu of normal mapping procedures, they seem to rely on pragmatic strategies such as the production of the animate (or more generally, the more potent or more salient) noun first." (Saffran, 1980, p.321).

Saffran defends this view over the positional level view by noting that if the positional level itself were the problem, then the ordering problems should be *random.* They are not. They correlate with parameters involving case role mappings onto syntactic roles. If the positional level alone were the difficulty, then the relative animacy of the role fillers should not make a significant difference.

Saffran points out that there is no reason to suppose that only one level of the system is affected. Researchers generally try to implicate only one for parsimony reasons. Her point is that what she terms the functional level,

which is responsible for mapping case roles to syntactic roles, is definitely implicated by these results.

Turning to comprehension in agrammatics, we find a similar set of deficits and explanations. Broca's aphasics have a remarkable ability to understand utterances when the semantic constraints of the lexical items are enough to determine the relationships among them. For example, they have little difficulty on sentence-picture matching tasks with complicated-looking sentences such as *The apple that the boy is eating is red* However, similar to their production behavior, if the sentence is reversible, such as *The cat the dog is chasing is black*, their performance falls to chance levels (Caramazza & Zurif, 1976). Also, on sentences where the position of a grammatical element is **crucial, such as *The woman is showing her baby the pictures* vs. *The woman is showing her the baby pictures*,** their performance indicates that they are not sensitive to the position of *the* (Heilman & Scholes, 1976). Again, the prevailing theory here is that the deficit is at the positional, or syntactic level[3]. However, Schwartz et al. (1980) found that some agrammatics performed poorly in a sentence-picture matching task even on simple sentences such as *The dog chased the cat*, where the nouns are of equal animacy, while performing well on sentences with animate subjects and inanimate objects. Again, this implicates the functional level. Saffran has pointed out that it is not a good idea to average over aphasic populations, and that this is one discipline where single case studies are valuable. In this regard, it should be pointed out that in the Schwartz et al. (1980) study, there was a variety of behavior among their five subjects. Two out of the five performed reliably on active voice sentences when the syntactic subjects and objects were familiar, usually human, fillers for these roles, such as "clown", "dancer", "man", "dog", etc. These two performed at chance levels on passive sentences using the same nouns and verbs. When the active sentences were altered to only use

[3]**It should be noted that even though this is termed the positional level, proponents of this account generally assume the deficit is a structure-building one. rather than an inability to appreciate the order of the words.**

"square" and "circle" (with suitable stick figure pictures), one of the two was no longer able to decode them properly, while the other continued to perform reliably. A third subject (B.L.) appeared to be generalizing S-V-O[4] to passives in the study with normal Subjects and Objects, but performed at chance with the squares and circles. A fourth subject performed at chance on all tasks relating to word order. Such results led Schwartz et al. to conclude that these subjects had a syntactic mapping deficit which they were compensating for by applying various heuristics inconsistently across the tasks, using different sets of rules at different sessions.

Linebarger, Schwartz and Saffran (1983) found that aphasics who performed poorly on the above active/passive discrimination test (one of the subjects was in the Schwartz et al. study) could reliably discriminate between syntactically correct and incorrect sentences for a wide range of types of grammatical errors, including problems involving verb subcategorization, particle movement, subject-aux inversion, phrase structure, etc. They did poorly on some types of errors, such as tag questions (**John is very tall, doesn't he?*). Overall, however, their performance was surprisingly good, considering the prevailing theories stated that either agrammatics had no access to functor information, or that their syntactic processor was in general, disrupted (cf. Berndt & Caramazza, 1980). Of particular interest here is that they did very well on sentences such as **She went the stairs up in a hurry* vs. *She went up the stairs in a hurry*, where it appears to require an appreciation of order information to discriminate between them.

Thus agrammatic aphasics can perhaps *compute* syntactic representations, but cannot *use* that structure for interpretation. This suggests that the overall picture of the system given in Chapter 1 is not that far off the mark; that is, there are independent access routes from the lexicon to the syntactic and semantic processors, and there is another path between the two of them that

[4]Subject-Verb-Object: the canonical structure of English sentences.

has been disrupted in agrammatic aphasics. After we have explicated our model in more detail, we will return to these results with an eye towards explaining them in terms of our model.

### 63. Implications for Aphasia

Let us return now to the results concerning agrammatic aphasics. Does our model have anything to say about them? Suppose that, following the Linebarger et al. (1983) results, that we assume that the *constraints* between syntax and semantics are gone. That is, the main pathway between syntax and semantics in our model, the connections between the binding nodes in syntax and the binding nodes in semantics (cf. Figure 4.16 and the next chapter) is disrupted[5]. The point of these connections in the model is to transmit constraints back and forth between the assignments of constituents to their roles in one system to the same assignments in the other.

Suppose that these constraints are gone; what could one compute in the semantic side? The example runs we just saw show that we can interpret a sentence correctly if semantic constraints are enough, conforming to the results of Caramazza & Zurif (1976). The definition of "semantic constraints" is somewhat clearer now; we mean that among the major lexical items in the sentence, there is a "best **fit"** for the case structure of the verbal item[6].

As we mentioned before, Saffran believes that in this area of research, single case studies are a valid form of data. We would extend this to Cognitive Science modelling of these patients; it makes little sense to try to "lesion" our model in one way to describe the behavior of patients who among themselves, differ wildly. Therefore, as an illustrative example of how our model may used

[5]**As noted in the next chapter, for this "connection" to be operative, a correspondence must be computed between syntactic constituents and semantic ones. Thus we can't assume a direct connection. Rather, since this correspondence must be *computed,* we assume that it is the neural assemblies performing this computation which have been destroyed.**

[6]**We would have to extend this to mean a conceptual frame best fit as well, to account for the Caramazza and Zurif results. That is, we assume a frame system for concepts: in *the apple that the boy is eating is red. red* would fill a slot in the frame for *apple* more readily than it would in the frame for *boy.***

as a framework for studying accounts of particular behaviors, we will concentrate on the study by Schwartz et al. (1980). We will only use the results of the Active/Passive with normal subjects and objects *(the clown applauds the dancer)*[1]. Table 6.1 (adapted from Schwartz et al. 1980) summarizes their results, A "+" means that the subject did significantly better than chance on that test a "--" means the subject did significantly worse than chance (there is only one instance of this, B.L., in which he consistently interpreted passive sentences as if they were actives), and a "0" means chance performance.

To model these deficits in our framework, it is necessary first to recall the results of Goodglass & Baker (1976) that the associative structure of the lexicons of the Broca's aphasics they studied appeared intact with respect to members of the +[Human] class, but impaired outside of that class. It is not unreasonable to assume that, given the wide range of behaviors within this class of aphasics, some patients will be more or less impaired on +[Human]. To model these deficits in our framework, we consider how this might affect the binding nodes for Agent and Object.

In English, the first +[Human] in a sentence is frequently the Agent. In the model, this information is reflected in activation from 4-[Human] in the

Table 6.1 Summary of Schwartz et al. (1980) Results (Experiment 1)

| Subject | Active Voice | Passive Voice | Proposed Deficit |
|---|---|---|---|
| B.L. | + | - | +Human-is-Agent, -"by" |
| H.R. | + | 0 | +Human-is-Agent, +"by" |
| J.R. | 0 | + | -Human-is-Agent, +"by" |
| V.S. | + | 0 | +Human-is-Agent, +"by" |
| H.T. | 0 | 0 | -Human-is-Agent, -"by" |

[1] The other pan of the study involved active sentences with "inanimate" subjects and objects *(the square applauds the circle)*, and prepositional verbal elements, e.g. *(he square is* above *the circle.* There were interesting results here, but we don't know how to account for them.

lexicon to the Agent binders: the Agent binder receives a stronger signal. This would have two results: The "CONC1 = Agent" binder would begin to suppress the other binders for C0NC1, as well as other $^{M}$ = Agent" binders, preventing them from activating. Notice there is no need for the *order* of the concepts coming in to be recorded; this is reflected in the Agent binder for the first concept being activated first, winning over the other C0NC1 binders, and suppressing other Agent binders. Without this information, the first concept could fill either the Agent or the Object case equally, so these would compete until some other information came along, or until a choice was forced, in which case the assignment would be random. Secondly, although we did not discuss it here (see Cottrell & Small, 1983), we assume that prepositions such as "by" have a representation in syntax as PREP, but a separate "meaning" representation in the semantic side as case signalers. That is, "by" would prime binders for the Agent, Location, and Method cases. We will assume that this *type* of information (case signaling by prepositions), which requires a different set of connections compared to those for content words, can be independently disrupted.

We explain the behavior of the five aphasics in the Schwartz et al. study as a combination of the loss of syntactic constraints on the bindings (the connections from syntax in Figure 4.16 are gone) and various combinations of loss of the extra weight for + [Human] to " = Agent" binders, and the extra input from "by" to " = Agent" nodes for concepts following the "by". Looking at Table 6.2, we assume that every subject that got active sentences right has the information that + [Human] is usually the Agent, so the binder for the first + [Human] that comes along wins. It is the interaction of this with the "by" information that is interesting. If they have the "by" information, then when the second candidate for the Agent comes along, even though the binder for that to Agent is being suppressed, it gets an extra boost from "by" (besides being a + [Human]), which activates it enough to cause it to compete on an equal basis with the " = Agent" binder for the first concept. Then the choice

becomes random, as in the cases of H.R. and V.S.

If they don't have the "by" information, then the first +[Human] still wins, as in B.L.'s case. We assume that if their choices on the Active sentences are random, then they are missing the +[Human]-is-Agent information. If they then do have the concept-following-"by"-is-Agent information, they will get the Passives right, as in the case of J.R. If they have neither of these abilities, they will choose randomly, as in H.T.'s case. Of course, if they had the constraints from syntax and both of these other information sources, then they would make the correct interpretations, as normals do.

If it were only possible to make up *post-hoc* stories, the model would be an interesting intellectual exercise, but of little scientific use. Fortunately, given how the model behaved in the sample runs in Chapter 4, we can make some predictions. We assume that the behavior of the model given in the that section corresponds to a "lesioned" complete model, since there are no syntactic constraints. Thus, we would have to predict that agrammatic aphasics could *disambiguate* lexical items based on the case structure. Thus if we gave them a sentence with an ambiguous word disambiguable in this fashion, such as *Fred hosted a* ball *for his friends*, and then asked, "What did Fred host?", they ought to be able to pick out a picture of a party from a picture of a baseball. This appears, however, to be a "grandmother result"; one my grandmother could have told me.

A more interesting claim is based on the apparent disassociation between the syntactic and semantic interpretation systems. Suppose agrammatics were given sentences containing noun-verb ambiguities, that were semantically anomalous given the syntactically correct interpretation, yet contained a coherent interpretation if the lexical items were interpreted agrammatically. The model would be confirmed if they formed the "correct" interpretation. For example, given the sentence, *The saw bobbed the rose* (or some such) they should interpret it as *Bob saw the rose.*

Unfortunately, another aspect of our model makes this prediction suspect. The model posits independent access to syntax and semantics from the word sense buffer. There is also feedback to the word sense buffer from syntax and semantics. Thus, there *is* a remaining pathway between the two in the model. This fact should be reflected by the word sense buffer's disambiguation function. If information from one system or another selects a particular sense of the word, this should be reflected in the representation formed by the other system after the information has had a chance to propagate through the buffer. Thus if a lexical item is syntactically disambiguable, then that disambiguation should be reflected in the semantic processing and vice-versa. If the two systems make opposing selections, then there should be a conflict in the buffer between the opposing definitions. If the model is correct, we would have a way of testing the strength of the two systems' contributions to the disambiguation process. One way which may give the semantic system a chance to beat this is to use words where the syntactically biased meanings are of low frequency, and the meanings that "mesh" are of high frequency. This is a characteristic of "semantic" garden path sentences.

This observation gives rise to a possible test of the existence of this pathway between the two systems. In a "semantic" garden path sentence, such as *the old man the boats*, the conflicting information that "man" is a noun comes from semantics[8]. If the connection through the word sense buffer exists, then either (a) semantics will win and they will judge the sentence ungrammatical, as normals often do, which would be evidence that there *are* still semantic constraints operating or (b) syntax will win, they will judge the sentence as grammatical, and their behavior on a picture matching task should reflect the syntactically biased interpretation, showing that at this level, they *are* using syntactic constraints in semantic interpretation. Appropriate controls would be something like *the old woman the boats* and *the sailors man the boats.*

[8]Actually, this is arguable. "Man" is probably most often used syntactically as a noun. There are probably better examples.

If this connection does not exist or has no effect, the agrammatics should judge this sentence as grammatical, yet still pick the semantically biased picture of an old man and some boats (rather than the old *manning* the boats). This would controvert the model's claim of the existence of a path through the buffer, but would give rise to an interesting picture of processing in the degraded system: it implies the existence, "side by side" of two incompatible representations of the sentence, each of which is acceptable to its particular processor. (It should be pointed out that it is probably a good idea to use synthesized speech for this, to avoid contour cues.)

Given that the proposed deficit is between attachment constraints, another test of the model is to see if these have effects in the agrammatics or not. Normals can be biased to make implausible attachments by priming them with sentences in which ambiguous attachments are all one way, for example where a final prepositional phrase is always attached to the VP rather than an adjacent NP. This attachment bias will perseverate when they are given a sentence that is semantically biased for the other attachment. Agrammatics, however, according to our model, should show no such effects. They should always make the semantically most plausible attachment, even if syntactically primed for another.

### 6.4. Conclusions

The major finding of this section is that the model can account for recent unintuitive results in the aphasia literature. The apparent ability of agrammatic aphasics to compute syntactic information while being unable to use it has a natural interpretation in our model. Further, the model makes nontrivial predictions that are direct consequences of the remaining "link" between syntactic and semantic representations through the representation of word definitions in the word sense buffer.

While any of these predictions are empirically testable, and particular aspects of our model may be proven or disproven, the claim remains that this

type of model is a suitable framework for testing theories about aphasia. We may suppose certain information sources are there or not, run the model and derive predictions. This argues for the continued use of such models for fruitful interaction between the various disciplines of Cognitive Science.

# CHAPTER 7

# A FORMAL BASIS FOR CONNECTIONIST INHERITANCE HIERARCHIES

## 7.1. Introduction

In Chapter 4, the model of the lexicon and the case system use a hierarchical structure to organize the information, and make use of the property of *inheritance* that such hierarchies enjoy. One problem with such hierarchies is that they can contain *exceptions* to inheritance; often a more specific concept will have a value of some role that is different from its superordinates. In this chapter, an attempt is made to put such hierarchies on a firm formal foundation. Reiter's (1980) Default Logic is used as a basis for the semantics of such hierarchies. The resulting implementation appears successful, and suggests further uses of Default Logic as a specification language for connectionist networks.

## 7.2. Background

In a recent paper, Etherington & Reiter (1983) (hereafter E&R) formalized a simple version of semantic networks (known as *inheritance hierarchies*) with exceptions in terms of Reiter's (1980) Default Logic. With this approach they were able to formally characterize the correctness of an inference algorithm in terms of Default Logic, and exhibited an algorithm that was correct in this sense. Finally, they concluded that massively parallel architectures for semantic networks, such as NETL (Fahlman, 1979), apparently cannot implement this algorithm. In this chapter, we show that a connectionist implementation of the simplified semantic networks outlined in their paper avoids the objections to NETL. We also present some results of

simulations in this framework of the examples presented in E&R.

### 7.2.1. The Problem

Semantic networks have been found to be an efficient and useful representation of knowledge by AI researchers for many years. One principal advantage is the ability to store information about objects at appropriate levels of abstraction in the IS-A hierarchy, so that the fact that dogs, elephants, and people nurse their young, for example, can be stored once at the MAMMAL node. Retrieving all of the properties associated with an instance of some class is done by an inference procedure that is particularly simple in these systems, known as *inheritance.*

As Hayes (1977) points out, there is an obvious correspondence between IS-A hierarchies and simple collections of FOPC formulas. For example, "Clyde is an instance of an Elephant" corresponds to the assertion Elephant(Clyde). Statements about classes, such as "Elephants are Gray", correspond to first-order formulae, in this case, (x).Elephant(x)→Gray(x). Inheritance can then be seen as a repeated application of modus ponens. One nice property of inheritance hierarchies is that, since they are acyclic, modus ponens can only be applied a finite number of times, no more than the depth of the hierarchy. Also, as pointed out by E&R, the node labels in such hierarchies are unary predicates, e.g. MAMMAL(x). Finally, no exceptions are permitted to inheritance. A dog is a mammal, no matter what.

Unfortunately, the real world is not as simple as a taxonomic hierarchy. Often it is useful to abandon the tree structure in favor of multiple inheritance hierarchies, and to allow *exceptions* to inheritance relations. This introduces non-monotonicity into the representation, as well as ambiguity. An common example of a non-monotonic rule is: "assume a particular Elephant is Gray unless proven otherwise." This is often known as *default reasoning* and has been formalized by Reiter (1980). When combined with multiple inheritance, default reasoning can lead to ambiguity. A well-known example is:

(1) Nixon is a Quaker.

(2) Nixon is a Republican.

(3) Republicans are normally non-pacifists.

(4) Quakers are normally pacifists.

Reiter's formalization of the above facts would be (assuming, for convenience, that Nixon is a type):

(1) (x).Nixon(x)→Quaker(x)

(2) (x).Nixon(x)→Republican(x)

(3) $\frac{\text{Republican(x):}\tilde{}\text{Pacifist(x)}}{\tilde{}\text{Pacifist(x)}}$

(4) $\frac{\text{Quaker(x):Pacifist(x)}}{\text{Pacifist(x)}}$

(1) and (2) are just the first order rules corresponding to (1) and (2) above. (3) is an example of a default rule. The formula to the left of the colon is called the *prerequisite* of the default. If this is known, and the part to the right of the colon, (the *justification*) can be consistently assumed (i.e., its negation isn't provable from what we know), then we can infer ~Pacifist(x), the *consequent.* The above rules, where the justification is the same as the consequent, are called *normal* defaults. Often, the justification contains all of the exceptions to the rule we know about. In this case, we might add "~NRAmember(x)" to the justification of (4). When such exceptions are included, the rule is called a *semi-normal* default. E&R point out that their correspondence between inheritance hierarchies with exceptions and default logic require semi-normal defaults (due to the identification of exceptions with clauses in the justification.)

Is an individual b for which Nixon(b) holds a pacifist or not? In Reiter's terminology, there are two *extensions* consistent with our knowledge. An extension contains the first order facts and is closed under the default rules as well as first order theorem-hood. One contains Pacifist(b), the other

~Pacifist(b). In general, the problem we want to solve is: Given an individual **b**, and a predicate **P** known to be true of **b**, we want to compute $P_1(b), \cdots, P_n(b)$ such that the $P_i$'s all lie within a single extension. As noted by E&R, we can ignore the predicate arguments here, and the default theory is purely propositional.

### 7.2.2. Correctness, Defaults, and Hierarchies

Before we discuss E&R's algorithm and our implementation of it, we need to provide a wider context than the system described in their paper to motivate some of the later discussion of our implementation. It has been an open problem for several years to define a correct inference algorithm for semantic network languages that allow multiple inheritance and exceptions. For example, Fahlman et al (1981) showed that their "shortest path" heuristic for NETL gave anomalous results. There now exist at least two formal systems which claim to provide the "correct" semantics for inheritance hierarchies. The problem is that they don't agree as to what constitutes "correctness". Following Hayes' example, E&R provided a semantics for such networks in terms of Reiter's Default Logic. Touretzky (1984) provides a mathematical semantics based on predicate lattices of a 3-valued logic. Both of them produced a correct inference algorithm for their theory.

Without going too much into the formal details, I will try to point out the major differences between their systems, and mention a third strategy. First, E&R's algorithm randomly chooses an extension when there are multiple ones. If there is only one, it will find it. Touretzky's system reports an ambiguity when it finds more than one extension. A more important difference, however, is that built into the heart of Touretzky's semantics is the inferential distance principle, which says: if A inherits P from B, and ~P from C, then "if A has an inheritance path via B to C and not vice versa, then conclude P; if A has an inheritance path via C to B and not vice versa, then conclude ~P; otherwise report an ambiguity." (Touretzky (1984) p.204) This captures our intuitions

about inheritance hierarchies, if we believe that subclasses' properties should override superclasses' properties. NETL tried to capture this preference by the "shortest path" heuristic to try to get a single preferred extension. Such heuristics have been shown to be incorrect, in that they can lead to "facts" that are not in any extension (Etherington, 1982). Touretzky's work was motivated by this problem, and when applied to NETL, shows that it is possible to have NETL operate correctly with respect to Touretzky's semantics, but only after the network has been "conditioned" in advance, which is an expensive operation.

An example that illustrates the inferential distance rule is shown in Figure 7.1, using the E&R network notation. The corresponding default theory is:

$$\{(x).Clyde(x) \rightarrow Elephant(x)\ (x).Clyde(x) \rightarrow Albino(x)\ (x).Albino(x) \rightarrow Elephant($$

$$\frac{Albino(x):\tilde{}Gray(x)}{\tilde{}Gray(x)}\ \frac{Elephant(x):Gray(x)}{Gray(x)}\}$$

E&R's system allows two possible extensions, one of which claims Clyde is gray. In other words, their formalism doesn't make use of the hierarchical

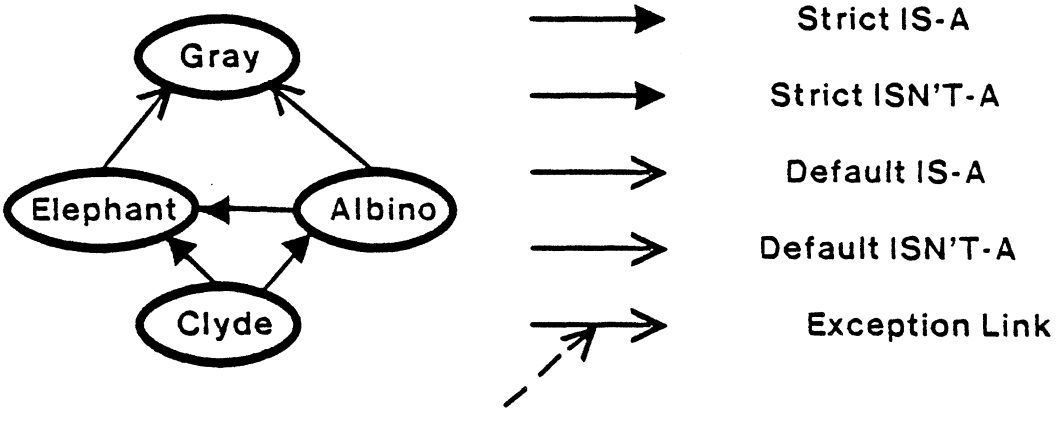



Figure 7.1. Clyde the Elephant, in E&R's notation. Gray or not?

nature of inheritance hierarchies. This counter-intuitive result is blocked in Touretzky's system by the inferential distance rule, which finds only one consistent extension: Clyde is not gray. To capture this preference in default logic, exceptions must be made explicit. An exception must be added to the justification of the last default rule:

$$\frac{\text{Elephant}(x):\ \text{Gray}(x)\ \&\ \sim\text{Albino}(x)}{\text{Gray}(x)}$$

This has the disadvantage that whenever a new exception is discovered, (for example, Albino Elephants) the default *rules* must change (see McCarthy, 1984) for a system that could be viewed as a different style of using default logic that only requires new rules to be added). The E&R formulation can lead to other counterintuitive results discussed in the simulation section.

A third approach which is enabled by this rare brush between the interests of formal logic and cognitive modelling, is to use an inference procedure which chooses an extension as part of the inferencing, based on some representation of belief strengths, in an attempt to model what people do. This could involve relaxing constraints that ensure consistency or completeness or both, and would require empirical verification. This approach may lead to less formal inference procedures, probably ones which are formally incorrect, yet may lead to important insights into everyday commonsense reasoning. One motivation for such an approach is based on the observation that one rarely "reports an ambiguity", as in Touretzky's system, when queried about ones' beliefs. One usually professes a belief. It also seems unlikely that on different occasions one would report "Nixon is a pacifist" and on another "Nixon is a hawk", as in E&R's, unless one's beliefs had changed in the interval. Also people may come up with inferences that aren't in any extension, analogously to the way they can make speech errors by blending parts of words, resulting in words that aren't in any lexicon. Considerations tike these are the motivation behind systems such as that reported in Shastri & Feldman (1984) and Feldman &

Shastri (1984), where weights on arcs in an active network encode belief strengths, and help choose extensions (not their terminology, however).

### 7.2.3. Etherington and Reiter's Algorithm

With these caveats in mind, we briefly review E&R's inference algorithm in intuitive terms. Those interested in the formal details may refer to their paper. The purpose of the algorithm is to "derive conclusions all of which lie within a single extension of the underlying default theory." When faced with multiple extensions, the algorithm randomly chooses one. The algorithm operates by successive approximations to an extension. Starting with the first order facts as a first approximation to an extension, it successively chooses (randomly) default rules which are not blocked by the current approximation *or the previous approximation*, and adds their consequents to the current approximation, until all of them are used. The constraints derived in previous approximations thus propagate to the current approximation. It iterates on this, starting with the first order facts again, until two successive approximations are the same (convergence). Etherington (1983) has proved that this algorithm will always converge on an extension. The randomness is essential to the algorithm's ability to derive any possible extension, if it is run "enough" times. A deterministic algorithm could be devised that simply considered all possible orderings of the applications of the defaults, if one wanted all extensions, and were willing to wait long enough! An important point about the algorithm as given is that it can be viewed as a relaxation-style constraint propagation technique.

Unfortunately, NETL is unable to capture such algorithms due to the "one-shot" nature of marker-passing. Markers are propagated through the network to find properties. The very existence of cancellation links in the version of NETL discussed in E&R (equivalent to exceptions; these were discarded in Touretzky's system) defeats marker passing because a link can be crossed before it is cancelled from a longer path. See Figures 7.2(a) and 7.2(b),

reproduced from E&R. In Figure 7.2(a), F must be reached before B in order to generate the extension properly, and vice-versa in 7.2(b). It is clear from this that the problem with NETL is not that it is a parallel machine. Rather, the problem is that is a *single pass* marker passing machine.

## 7.3. An Alternate Parallel Approach

### 7.3.1. Introduction

An obvious answer to these objections is to relax the "one-shot" nature of the parallel network. Connectionist networks, being iterative, have no such restriction. The obvious direction to take is to use a NETL-like network, with connectionist units and activation-passing instead of marker passing. This is the basic idea of the implementation described below. An "extra" that is derived from this model is a small step in the direction of answering the

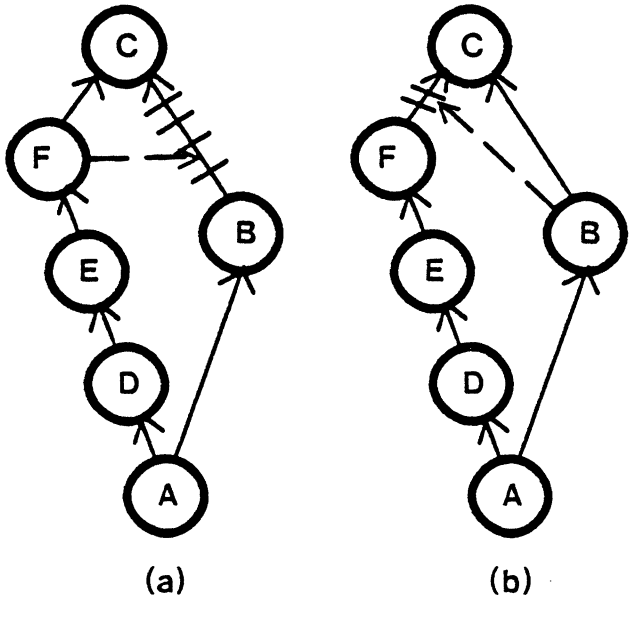



Figure 7.2. Networks which defeat the shortest path heuristic.

following question: What constraints on the functions used in connectionist networks can be reasonably .assumed without losing computational ability? In the following model, we show that even with strong constraints on the functions computed, we can still get fairly powerful results. Also, the specification of the network is extremely simple, the network is generated by a LISP program from simple commands such as (isa 'Clyde 'Elephant), The bit of network generated from this is specified by a simple mapping from Default Logic to units and links. This and other considerations lead to the idea of using Default Logic as a specification of more complex networks, if the mapping can be "raised" from the propositional to the FOPC level. This will be discussed in more detail at the end of this chapter.

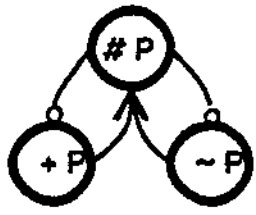
In the definition of connectionist networks in Chapter 2, there is no mention of time. That is, in simulating such networks, the units could be scheduled for updating in various ways: They could be kept in lock step (synchronous) or they could be updated in random order, with some units perhaps being updated several times before another gets a chance to be updated (simulating asynchrony). In all of the other simulations described in this thesis, synchronous updating has been used For technical reasons, we use asynchronous updating in the following model Whether the other networks described in the thesis could be run asynchronously is an open question.

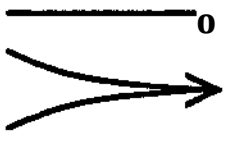
In the following, we present a connectionist model of semantic networks of the kind discussed in E&R. It should be kept in mind that these have a particularly simple form. Properties are not distinguished from type nodes, and there are no two place predicates. For a different formulation of semantic networks in connectionist terms which overcomes these objections, see (Feldman & Shastri, 1984).

### 7,3.2. A Connectionist Inheritance Model

In E&R, a correspondence was made between the five link types of a semantic network (Strict IS-A and ISNT-A, Default IS-A and ISNT-A, and

exception links) and formulae in Default Logic. Since our purpose here is to show that a connectionist network can mimic their inference algorithm, we start with formulae from Default Logic that correspond to inheritance axioms and display the corresponding bits of network. The first step, however, is to choose a representation of the predicates. Following the unit/value principle, we will start with two units for every predicate P, called +P and "P, representing the two different possible assignments of truth values to those predicates. When computing an extension, a node that is firing (after convergence) represents that it is part of the extension. There is an immediate consistency constraint between these two nodes, i.e., they should not both be on in any stable state. Thus we should make them mutually inhibitory. An immediate observation we can make at this point is that if the two nodes were to both remain on at a low level without a clear winner, due to equal evidence, then we have either an ambiguity (if they are on due to competing default inferences) or an inconsistency, if they are on due to first order inferences. A method for detecting such conflicts is given in Shastri & Feldman (1984). A problem with this formulation of predicates is: How should the inhibitory evidence be weighed by the evidence function? If we make one unit being on prevent the other from ever coming on ("eager" inhibition), we would duplicate the shortest path heuristic (whoever got inferred first would stay on), which we know to be incorrect. Thus we have to make it "fair" in some sense. A unit that has evidence should be allowed to propagate that evidence before being inhibited. This is essential if we are to consider all possibilities in parallel. Thus we introduce a third unit, #P, (to use Touretzky[1]s notation, if not his semantics), which represents "inconsistency." This node inhibits +P and "P if *both* of them are firing. Thus this introduces a delay in the inhibition between +P and "P. See Figure 7.3. There is a conjunctive connection from +P and "P to #P to encode the semantics that both have to be on for #P to come on. Then it outputs the maximum of the two, inhibiting both +P and ^P[1]. We can now, if we wish, monitor the #P unit to detect inconsistencies or



**Inhibitory connection**

**Conjunctive connection**

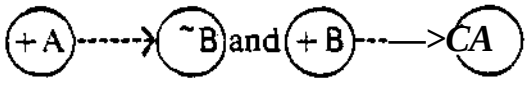
Figure 7.3. The Spock representation of the predicate P.

ambiguities.

We make the following correspondence between the four formulae defined in E&R, and networks in the connectionist framework. All links shown have weight 1.

(1) (x).A(x) -* B(x): This corresponds to two links:


(2) (x).A(x) -* ^B(x): This corresponds to two links:

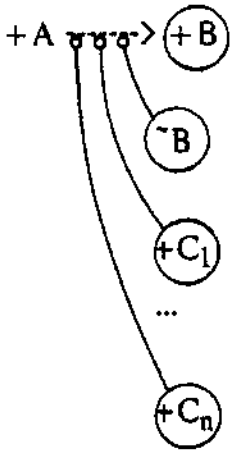



[L]An alternate formulation would have been to use two of these units so that the stronger unit would not receive its own inhibition, but the opposing unit's. We intend to explore this option in the future.

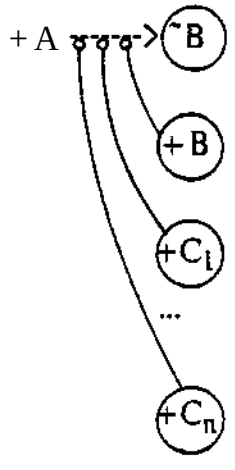
A(x): B(x) & *Ci(x) &...& ~$C_n$(x)

(3) —-L-Ll——-1ŁJ!———21_i (where n is possibly 0):

b(x)




**(4)** $\frac{A(x): \sim B(x) \& \sim C_1(x) \&...\& X_n(x)}{\sim B(x)}$ (where n is possibly 0):




Our encoding of an inference rule is thus just to put a positive link from the antecedent to the consequent. This is the first link in the encodings of (1) and (2), and it guarantees that if +A comes on, eventually +B (or "B, as appropriate) will come on as well. Since these correspond to first order facts, we don't allow any exceptions to these (i.e., modifier links - see below). In an attempt at some semblance of completeness, we encode the contrapositive as well (remember we have no interpreter to deduce these consequences). This is

the motivation for the second link in the encodings of (1) and (2). We are explicitly adding the rule 1 -> "A (or B -^ "A, in (2)). In E&R's formulation, extensions are closed under first order inference. This has some odd consequences, which we will see in the Cephalopod example.

In the encodings of (3) and (4), there are no "backwards" links like these, because the contrapositive doesn't hold for default inferences. However, "B in (3) (as well as any of the -fQ's) for example, should block the inference of + B if it is on, so we use a modifier link from "B to the link between + A and + B. If there is any evidence at all for "B, it should not have to compete with + B, if +B was inferred by default The modifier link blocks activation from crossing this link if "B has any output at all. Thus it seems a good choice for encoding the semantics.

An inference, then, corresponds to, e.g., +A activating +B. This inference may be retracted later if "B is inferred by some (necessarily first order) rule. The retraction is accomplished by using activation functions that cause a unit's potential to go to 0 if its evidence goes to 0. An extension will then correspond to a stable state of the network.

Due to the limitation that modifier links have not been implemented in our simulator for connectionist networks, ISCON (Small et al, 1982), we have taken the liberty of introducing an extra node on all links (an "isa" node) to provide a site for the modification. Thus a modifier link is realized as an inhibitory link to the link node, and the link node has a special activation function that sets the activation to 0 if there is any inhibitory input, to duplicate the semantics of a modifier link. This encoding has the effect of delaying propagation uniformly throughout the network. The IS-A units are useful sites for control, however, and are necessary in a more fully specified model (see Shastri & Feldman, 1984). Unfortunately, it also has the effect that if, for example, + B is inferred by a default rule, and ~B is inferred in some way, then + B and "B end up competing more than they "should", in pan due

to the asynchrony. +B may get updated several times before the "isa" feeding it gets updated and turned off. This causes longer settling times for the network.

Finally, we specify what the units compute. An evidence function which appears reasonable for our purposes is to take the maximum of all positive input (from subtypes) and add the minimum of all inhibitory inputs. The motivation for this rule is that we will use one "source" for the network's activation, namely the predicate whose extension we are seeking, and so it does not seem appropriate to use more evidence than the maximum from that source. However, there are arguments against this. In a non-demonstration system, one would want an alternate way to combine evidence, for example Dempster-Shafer rules (see Ginsberg, 1984), for some extensions of Dempster-Shafer rules for semantic networks). This is especially important for default rules. If someone is a Republican *and* an NRA member *and* a veteran then we would be more inclined to assume they are not a pacifist, even if they are a Quaker. See Feldman and Shastri (1984) for more discussion on this. In any event, for the examples we will be discussing, the max and min rule appears sufficient.

The result of the evidence function is passed to the activation function. We have implemented two activation functions in this system. One uses table lookup and looks almost like iterative marker passing (it only uses three values), and the other is more continuous-valued. The first function appears in Table 7.1. The basic idea is to move towards the value of your input. This version of the system we call Spock because it doesn't allow intermediate interpretations. A unit is on, or it isn't. The second uses the activation function in Table 7.2 (due to McClelland & Rumelhart, 1981). In this function, E is the result of the evidence function, p is the potential and d is a decay constant (we used .2 in the simulations). We call this version Dr. Spock because it is more permissive, allowing intermediate values. If one had no decay, then starting the system with a 1 on the assumed unit, and only using

Table 7.1. The Spock activation Function

| Evidence | Current Pot. | New Pot. |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | -1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | -1 | 0 |
| -1 | 0 | -1 |
| -1 | 1 | 0 |
| -1 | -1 | -1 |

Table 7.2: The Dr. Spock activation function

| p <- p + [if E < 0 then E*p else E*(1 - p)] - d*p |
|---|

weights of 1 or -1 on links, then the result is that units only take on the values 0 or 1. The problem with this is that if a unit is at 1, then it will stay there if it is not inhibited, causing false inferences to stay around. The desire to try to have a system with no decay led to the table lookup function above.

The goal is to solve the same problem as E&R, that is, given an individual b, for which P(b) holds, determine all predicates $P_1 \cdots P_n$ such that P(b), $P_1(b), \cdots P_n(b)$ belong to a common extension. We do this inference by clamping on node +P. After convergence the nodes that are "on" are in an extension.

We should state at this point that there is a major difference between E&R's algorithm and the following implementation. We claim here, without proof, that if the network only encodes a consistent set of first order (inheritance) rules, we can allow all first order rules to fire in parallel and the network converges. An interesting question is whether we can allow *all* inferences to proceed in parallel, including the default inferences, while relying on our predicate networks to guarantee consistency. While at this stage we have no proof of convergence or correctness, experimental results with the system support the conjecture. E&R's algorithm stipulates introducing one

default rule at a time, generating all inferences, and then trying another, so we depart from their algorithm in this. The difference is that we don't wait for first order consequences to propagate before we try another default. We use a random update order (simulating asynchrony) to allow one default to run before another, as in E&R's algorithm. If they are "competing" defaults (as in the Nixon example), this ensures one will block the other, so this is basically a tie-breaking strategy. In a synchronous network, one could use noise to break ties.

Here is a rough sketch of how to mimic their algorithm exactly. The control issues in the following have not been worked out, but there is nothing inherently hard about them. The following would be done synchronously, that is, the units are updated in lock step. The entire network would have to be duplicated once. An approximation to an extension is generated in one network as follows: Clamp on +P, using enable links (control links) to the strict IS-A nodes to allow only first order inferences. As claimed above, if the first order knowledge is consistent, there is no problem in doing all first order inferences in parallel. It would take two times the depth of the network iterations for all inferences to propagate (besides upward inferences, there are the contrapositive inferences that come back down). Then, randomly enable one default IS-A link (along with the strict IS-A's), and wait the same length of time. Repeat this process until all default IS-A's have been enabled once. Then send a signal to all nodes to "clamp on" their state. The next approximation to an extension is generated in the other network, using the knowledge inferred in the first network to block default inferences (modifier links are to the default IS-A's in both networks). Then the original network is reset and the operation continues until two successive states are identical. Again, the difference between this and our simulation is that we try to see how far we can get by just using one network and letting *all* inferences proceed in parallel, including the default inferences, while relying on our predicate networks to guarantee consistency.

## 7.4. Simulation Results

We present the results of simulating several of the networks from E&R. In the following, the results are from the Spock version except where noted. This is because in almost all cases, the results from the two systems were similar. We use asynchronous updating of units to simulate the randomness of E&R's algorithm[2]. That is, units are all equally likely to be simulated at any point. An "iteration" consists of doing as many updates as there are units in the network. Note that this doesn't mean that all units have been updated; some may have been updated more than once, others not at all.

We begin with the Cephalopod example from E&R, shown in Figure 7.4. The source of this example is (Fahlman et al, 1981). In English, it's:

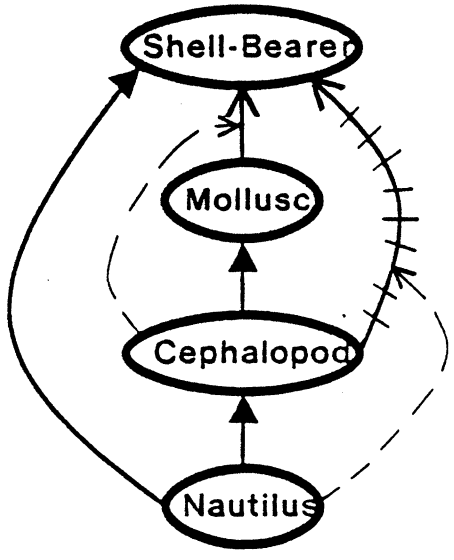



Figure 7.4. E&R's network representation of Cephalopod facts.

[2]We implemented a synchronous version as well. Based on simple examples, it appeared that if there was a unique extension it would find it, and oscillate on examples with more than one, allowing us to report an ambiguity. We discarded it upon finding it didn't converge on the "hard" example discussed at the end of this section, even though it had only one extension. The fixes for the asynchronous version discussed at the end of the chapter would probably help here as well.

Molluscs are normally shell-bearers.

Cephalopods must be Molluscs but normally are *not* shell-bearers.

Nautili must be Cephalopods and must be shell-bearers.

The default theory corresponding to this (as given in E&R) is:

$$\{\frac{M(x):\ Sb(x)\ \&\ \tilde{}C(x)}{Sb(x)},\ (x).C(x)\rightarrow M(x),\ (x).N(x)\rightarrow C(x),$$

$$\frac{C(x):\ \tilde{}Sb(x)\ \&\ \tilde{}N(x)}{\tilde{}Sb(x)},\ (x).N(x)\rightarrow Sb(x)\}.$$

The connectionist implementation of these rules is given in Figure 7.5.

For this example we spread activation from the +Cephalopod node in order to find the extension of Cephalopod. To shorten the tables, only

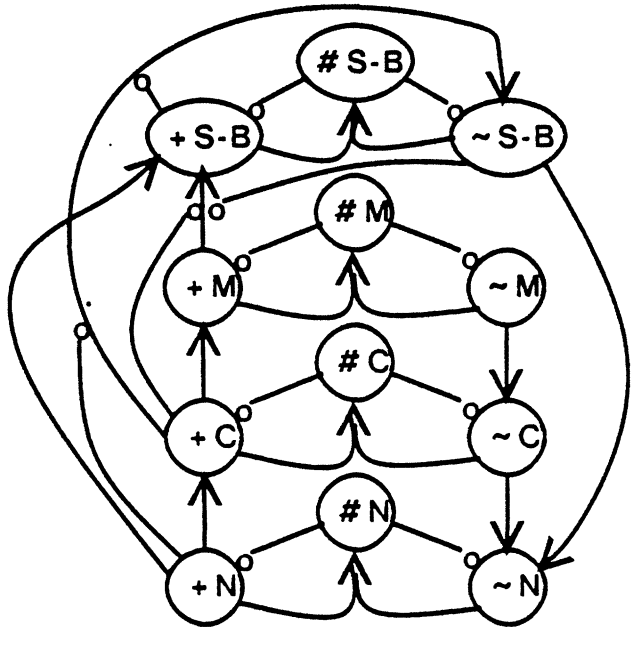



Figure 7.5. The Spock implementation of the knowledge in Figure 7.4.

iterations where one of the units in the table changed are shown. This is a "typical" execution, although of course they are rarely the same. We simulated this network about ten times, and it never took longer than 15 iterations to converge. As the results in Table 7.3 show, activation spreads from +Cephalopod and activates ˜Shell-Bearer, which blocks the IS-A link from +Mollusc to +Shell-Bearer.

An interesting result here is that ˜Nautilus is inferred! Since E&R require that "an extension ... is closed under the Defaults of D as well as first order theoremhood", then they will have to live with this. The inference came about because ˜Nautilus was consistent, allowing us to infer ˜Shell-Bearer from +Cephalopod. However, since Nautilus→Shell-Bearer, first order, then ˜Shell-Bearer→˜Nautilus, and we get the result that we prove ˜Nautilus from assuming it to be consistent. This is not unintuitive, since if Cephalopods are usually not Shell-Bearers, then they are usually not Nautili. This is just not what we expect from an inheritance hierarchy. Not including such "downward" inferences would eliminate completeness, but would also eliminate a problem with the final example.

If we activate +Nautilus instead, activation spreads to +Cephalopod and +Shell-Bearer, and +Nautilus cancels the IS-A from +Cephalopod to ˜Shell-Bearer, resulting in the correct extension.

Table 7.3. Trace of Unit Outputs from Example 1

| | Activating Cephalopod | | | | Activating Nautilus | | | |
|---|---|---|---|---|---|---|---|---|
| Iteration | 2 | 3 | 4 | 5 | 3 | 6 | 8 | 9 |
| +Nautilus | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| ˜Nautilus | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| +Cephalopod | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| +Mollusc | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| +Shell-Bearer | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| ˜Shell-Bearer | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

If we use the default theory relating to the NETL version of this hierarchy given in E&R (see Figure 7.6), we get an ambiguity with respect to whether a given Cephalopod has a shell or not, since the default IS-A from +Mollusc to +Shell-Bearer is not cancelled. Also, in this version all inferences are defaults, so ˜Nautilus is never inferred. Our network shows a marked preference for shortest paths in this case. In 20 runs, we got ˜Shell-Bearer 18 times and +Shell-Bearer only twice. This is not surprising, given that the activation is most likely to follow the shorter path first. (In simulations of the Nixon example described earlier, where the paths are of equal length, the the results were 50-50 between the two extensions.) In one run, the end of which is shown in Table 7.4, both +Shell-Bearer and ˜Shell-Bearer were inferred at the same time. This is an implementation artifact, since a true modifier link would have prevented this from happening. However, it illustrates the robustness of the system. The ambiguity resolution of the #Shell-Bearer node saves us from the

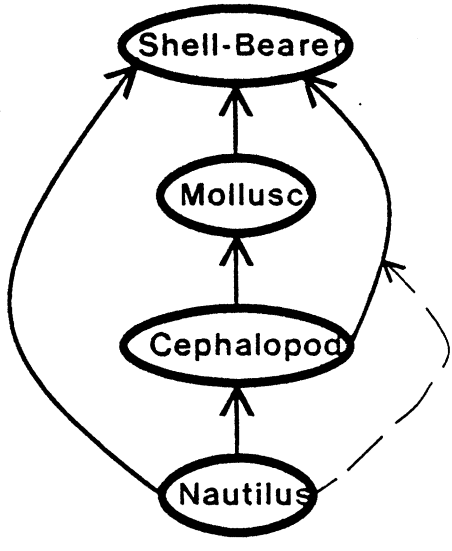



Figure 7.6. The NETL version of the Cephalopod example.
All inferences are defaults.

implementation. (In all other cases, the network converged in 10 iterations). It comes on only when both of its + and ˜ nodes do, and inhibits them. Because of the asynchrony, eventually one of the nodes comes on without the other.

In example 3 (see Figure 7.7) we duplicate E&R's network that illustrated their inference algorithm. As Table 7.5 shows, at iteration 12 unit +D gets activated by unit +B. Because +C has blocked the IS-A link from +B to ˜D, it doesn't have to compete with ˜D. Sometimes we get the behavior their algorithm exhibited. In Table 7.5, we also show a run where ˜D gets inferred first. The fact that it gets inferred even though +C has already been inferred is again, an artifact of our implementation: The IS-A node from +B to ˜D was already on before +C was inferred, and ˜D ran before the IS-A node which supported it was able to run again and be blocked.

Table 7.4. Ambiguity Resolution at Work in Example 2

| Iteration | 10 | 12 | 14 | 16 |
|---|---|---|---|---|
| +Shell-Bearer | 1 | 0 | 0 | 0 |
| ˜Shell-Bearer | 1 | 1 | 0 | 1 |
| #Shell-Bearer | 0 | 1 | 0 | 0 |

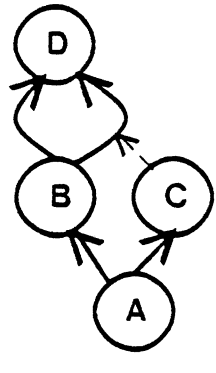



Figure 7.7. From E&R's example of their procedure behavior.

Table 7.5. Trace of Unit Outputs from Example 3

| Iteration | Activating A | | | | A run where ˜D gets inferred | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 4 | 9 | 12 | 5 | 6 | 7 | 9 | 12 |
| +A | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| +B | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| +C | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| +D | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| ˜D | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

In example 4 we show that we overcome the fixed radius problems of NETL. The networks in Figure 7.2 defeat any shortest path heuristic. As Table 7.6 shows (left hand side), since activation is computed continually, the effects of node +F are felt when it gets activated, and the IS-A from +B to ˜C is cancelled, allowing +C to become active. Note that, because this is all default inferences, +C can't be inferred until ˜C goes off, because ˜C blocks the inference of +C. Table 7.6 shows that the network of Figure 7.2(b) works as well. In this case, it is practically impossible for ˜C to come on, since +B is almost always inferred early, blocking the inference of ˜C. If it ˜C did come on, the network would still converge as it did for the network of Figure 7.2(a). In this case, the Dr. Spock version was a little slower, because ˜C took several

Table 7.6. Trace of Unit Outputs from Example 4

| Iteration | Figure 7.2(a) | | | | | | | Figure 7.2(b) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | 5 | 7 | 11 | 14 | 16 | 25 | 2 | 4 | 5 | 6 | 8 | 11 |
| +A | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| +B | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| +C | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| ˜C | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| +D | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| +E | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| +F | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

iterations to decay to 0. The modifier links are strict, so that any output from X stops default inference of +C.

Example 5 illustrates that our networks, like the algorithm they imitate, don't subscribe to Touretzky's inferential distance rule. The network in Figure 7.1 has two extensions according to default logic, one in which Clyde is Gray, and another in which he is not This is isomorphic to the Nixon example, except that there is an IS-A link between the two intermediate nodes. This does not cause the network to behave any differently, however. A sample run is given in Table 7.7. It is interesting to note that since there is no criterion to choose between the two extensions in this case (both "Gray and +Gray block the inference of the other, and are equidistant from the source of activation), the ambiguity resolution takes considerably longer. The network has to find a stable bit pattern of activation, and the search appears random. The pattern involves more than the nodes shown, due to the extra link nodes between each node shown. However, because of the randomness of execution, we can be assured that it will find it. (Also, if some of the units' behavior seems anomalous, it is helpful to recall that we only see the result of what may have been several changes to each unit in a single iteration.)

It appears that our networks behave rather well. The final example shows that this is not always the case. The example is given in Figure 7.8. It is nearly identical to the example in Figure 7.2(a), except that the inferences are

Table 7.7. Trace of Unit Outputs from Albino Elephant Example

| Iteration | 11 | 12 | 13 | 14 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 24 | 25 | 27 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Clyde | 1 | 1 | 1 | "l | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Elephant | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Albino-El. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| +Gray | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | 0 |
| "Gray | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| #Gray | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

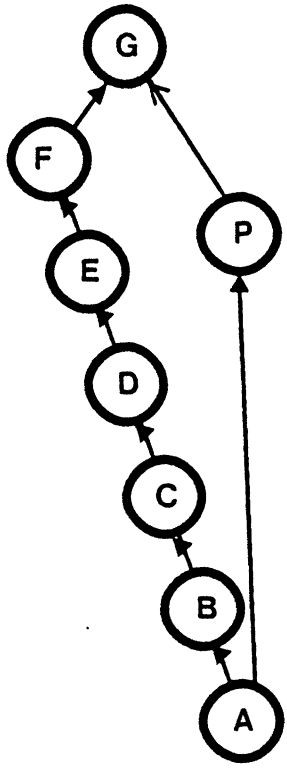



Figure 7.8. This one causes Spock some trouble (but not the Dr.).

all first order in the left hand chain. This example looks harmless because it has a unique extension. Unfortunately, it takes Spock over 1000 iterations to converge on that extension (but Dr. Spock was an order of magnitude faster: see below). To see why, recall that first order IS-A's allow downward inference of ˜P's. (Both A→B and A→˜B result in contrapositives with negative consequents). Secondly, the delay in inhibition between a "+" node and a "˜" node induced by the "#" node allows inference chains to "pass" each other. What happens in this case is that the default inference of ˜G starts a downward chain of "not" inferences. This meets an upward chain of positive inferences. They pass each other, then the consistency constraints begin

turning off both + and ~ predicate nodes in the middle of the chain. The inference chains then meet rather incorrigible resistance at both ends. +G has a hard time blocking the inference of ~G because by the time it gets there, its support is falling apart behind it. If it fails, ~G is inferred again straightaway. +A is clamped on, and +P follows from that. Then the process starts over again. The ambiguity resolution search of example 5 is multiplied by the fading in and out of the bottom up support.

The problem is, of course, that we didn't wait for the results of our first order knowledge before applying a default rule, and it took a while getting to the point of disagreement. In all of the other examples, the ambiguity was very localized. Local ambiguities are not hard for connectionist networks to handle. It is ones that depend on global properties of the network that are hard to deal with and still maintain a small radius of communication (an unspoken assumption in most connectionist models). There are several ways this problem could be avoided. One is to introduce a more continuous activation function. This is one place where there is a big difference between the two activation functions. The Dr. Spock version converges in somewhat over one hundred iterations, an order of magnitude better, and more in line with the time taken in the other examples. The reason appears to be that the units are less "all or none". Because +G decays slowly, it is more likely to be on when +P tries to infer ~G, since any nonzero output from +G blocks the inference. The search for a stable bit pattern, on the other hand, can take a lot longer.

A second way which could augment the first is to make default inferences literally less strong than first order ones. If a default inference link is weighted by .5 instead of 1, then a first order inference could more easily overcome it. In this case, we need a activation function that converges to its evidence, to propagate the .5 value. (unfortunately, this is not a property the McClelland & Rumelhart function (called Dr. Spock here), enjoys). The function:

p <- p*(1-d) + E*d

achieves this[3]. Experiments with this scheme have been encouraging. The table lookup function could also be altered to reflect this other value. First order inferences which had a default at their source would then reflect that fact in their potential. In this way information that was non-local could be encoded in the signal. This still doesn't avoid the problem altogether. There is no reason why a default inference could not be at the base of each chain. It appears that a different consistency "gadget" may be required, perhaps one that randomly selects one of +P or ~P to inhibit and waits until the depth of the network updates to stop inhibiting, allowing first order consequences to propagate.

A third way which avoids the pitfalls of the others is to disallow downward inferences altogether, by going the way of NETL, and assuming that everything is a default inference. (Or by foregoing our attempt at completeness; hence not encoding the contrapositive.) This does avoid the problem of two default inferences being at the bottom of competing chains. The competition at the top is a local one, since either outcome is consistent. Adding weights reflecting belief strengths, as in Shastri & Feldman (1984), and advocated by Rich (1983), might make such a system a possible cognitive model.

## 7.5. Future Work

### 7.5.1. Correctness

The examples are an informal (engineer's) argument for correctness. What is missing from this presentation is a formal proof of correctness. A first cut would be to show that if the network is in a stable state, then the predicate units that are firing represent an extension, and leave the problem of convergence for another time. One point that is clear that the subnetwork of

[3]It turns out (unbeknownst to us when we derived it) that this is the function used by McClelland in his (1979)

three units representing a predicate has only three stable states:

(1) They are all off.

(2) +P is on.

(3) ˜P is on.

Any other configuration of activation will cause changes in some unit's activation. If only #P is on, it will go off. If only one of the others are on as well, then #P will also go off. If both are on, then #P will inhibit them until one goes off. If #P is off and both of the others are on, #P will come on. Using this we hope to come up with an inductive argument that if the network is in a stable state then it is consistent.

### 7.5.2. A Specification Language for Connectionist Networks?

An interesting observation about this implementation of Default Logic is that the consequents of default rules, even if inconsistent, are often "entertained" at the same time. There is an obvious correspondence between this and the usual method of "search" in a connectionist network: let all possible hypotheses activate, and then let the network "relax" to a consistent interpretation. If enough constraints are encoded, unique solutions are often found. If a mapping from general default rules, not just the propositional ones relating to inheritance axioms used here, could be found, then specifying major portions of a connectionist network could be reduced to writing axioms in Default Logic. Examples relevant to this thesis are given below. Of course, there are many control problems that may not be amenable to this treatment. Also, assuming this was used for cognitive modelling, weights on links are assuredly necessary; this would have to be expressed as annotations to the default rules.

The main technical problem with this proposal is keeping track of variable bindings. For example, in a rule such as,

---

Cascade model.

$$\frac{(x):\text{Quaker}(x)\ \&\ \tilde{}\text{NRA-member}(x)\ \&\ \tilde{}\text{Veteran}(x):\ \text{Pacifist}(x)}{\text{Pacifist}(x)}$$

unlike the propositional case, the network generated has to somehow enforce that the same "x" that is a Quaker is the one that is not an NRA member, etc. One scheme that may work for this is given in (Feldman & Shastri, 1984), where arguments to schemata are bound through Feldman's (1982) dynamic binding mechanism. More investigation is necessary.

Given such a specification language, networks implementing a parser such as the one described in this thesis could be specified using default rules. For example, noun-verb ambiguous could be specified by such rules as:

$$\{(x).\text{Noun}(x) \rightarrow \tilde{}\text{Verb}(x),\ \frac{\text{rose}(x):\ \text{Noun}(x)}{\text{FLOWER}(x)},\ \frac{\text{rose}(x):\ \text{Verb}(x)}{\text{STAND}(x)}\}$$

which would generate something like the network in Figure 7.9. Notice that disambiguation "falls out" of this representation; if the Verb node gets feedback, it supports ~Noun, which blocks the input to FLOWER. An

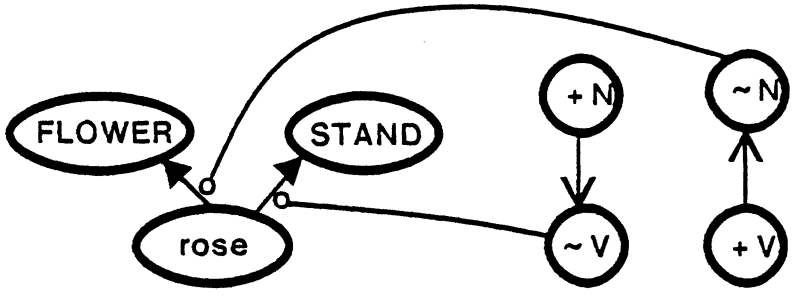



Figure 7.9. Network representing the meanings of "flower".

interesting consequence of the mapping is the explicit representation of such concepts as ""Noun", which then are used to block the inference of meanings which are not Nouns. Parsing rules can also be represented. A bottom-up rule is:

{Det(x) & Noun(y) & Adjacent(x,y) & cons(x,y,z) -* NP(z)}

Technical problems here involve not only variable binding, but creating instances of concepts, which crept in as "cons(x,y,z)"; a mechanism for labeling the pair "x" and "y" as "z" is necessary. The first orderhood of this rule makes feedback possible, but only negative feedback if a competitor of the NP wins. Then ""NP(z) would imply "Det(x) or "Noun(y), etc. It is left as an exercise to the reader to devise a network which will enforce this in the propositional case (for example, if cons(x,y,z), Adjacent (x,y) and Noun(y) are all true, then the network should force "Det(x)).

Another possibility is to view the default logic specification as a schema for the network, and to use machinery outside the logic for handling the bookkeeping problems of instantiation, labeling and coordination. For example, rules such as these could be the knowledge stored in a connection information source (McClelland, 1985), and the programmable buffer mechanism could keep track of the variable binding, instance generation, and coordination of bindings.

If the technical problems with this approach turn out to be tractable, this would be a great step forward in the ability to specify connectionist networks for language processing. In vision, the problems are often expressible analytically, allowing mathematical specification of the networks (Ballard, 1984). The non-analytic nature of natural language makes a mathematical approach problematic at best. Logical specifications are at least in the right ballpark, thus this seems a promising avenue for future research.

## *1.6.* Conclusion

We have seen how a connectionist model of inheritance mimics E&R's inference algorithm, avoiding the problems of NETL So, a massively parallel inheritance scheme *can* work. Two caveats should be mentioned. First, this is within the context of a very simple characterization of semantic networks. Second, our algorithm takes an inordinately long time to converge using the Spock activation function in the final example. This raises serious questions about its effectiveness in real domains, even on a parallel machine. However, the Dr. Spock version did converge in a reasonable amount of time, and by foregoing completeness, either system would converge reasonably. There is obviously much left to be done. For example, a correctness proof. Also, an interesting avenue for exploration now is using weights on the links to encode default strengths. This could have a speed-up effect on convergence, and could possibly be an interesting cognitive model. Also, specifying an evidence function that would better reflect the contribution of multiple sources of evidence is left for future research. Finally, this approach, if augmented to handle more than propositions, holds promise as a way of specifying connectionist networks in a formal language.

# CHAPTER 8

# CONCLUSION

## 8.1. Conclusion Introduction

Cognitive Science is a discipline that attempts to apply the insights of Computer Science, Psychology, Neurophysiology, Philosophy and Linguistics to the problem of human cognition. It is a field that will stand or fall on the basis of whether this interdisciplinary approach continues to generate new ideas that did not grow out of any of the various disciplines alone. To be successful, it will have to prove its worthiness by converging on one or more shared paradigms, that focus research in the parent fields on shared questions that flesh out a coherent picture of cognitive processing.

The model presented in this thesis has many implications for sentence processing within the various sister disciplines of Cognitive Science, and as a result, exemplifies the possibilities of Cognitive Science itself as a discipline. This work also adds to a growing body of literature that argues for the use of connectionist models as the paradigm for Cognitive Science. By virtue of the roots of the approach as an abstraction of the information processing capabilities of neural networks, it focuses on "the brain" as the central object of interest in forming cognitive models. The fundamental question becomes: How could one possibly walk, talk, see, hear, and plan, given that all one has to direct all this activity is a mass of highly connected, simple processing units? This thesis has been an attempt at answering part of this question.

A major premise underlying this work is that it is fruitful to apply constraints currently available from the various disciplines when designing

models. Many of the design decisions were directly motivated by empirical data. The end result is a model which feeds back to the constraining disciplines, generating new hypotheses that are empirically testable. As in the usual scientific cycle, data generated by these tests will no doubt cause refinements in future models and continue the interaction. The primary contribution of this work is the initialization of the cycle, illustrating by example the possibility of a connectionist model of sentence processing. The next section summarizes the major implications of the model.

## 8.2. Summary of the Implications of the Model

### Lexical Access

Two models of lexical access were presented here. The one in Chapter 3 satisfied constraints from recent data on lexical access and explained that data. Given four-way ambiguous words such as "deck" (two noun and two verb meanings) in a lexically biasing context, the model predicts that the inappropriate meaning that is within the class of the biased meaning will be deactivated first, followed by the other two meanings. Using this as a timing argument, it claims the selective access results of Seidenberg et al. (1982) were the result of testing the activation of the inappropriate meaning after it had been resolved. The model waffles on this claim, providing an alternate explanation that does produce selective access based on different model parameters.

The lexical access mechanism used in the syntactic processor of Chapter 5 makes the opposite claim. Given that model, the inappropriate class meanings would fade first. This makes it an unrealistic model, given the Seidenberg at al. results. Again, a shift in model parameters could maintain the original prediction. It may be desirable to try to "save" the mechanism this way, since it has the appealing feature of allowing alternate within-class definitions to remain viable, but "hidden" from the rest of the network. This predicts that recovery from within class inappropriate meaning selection is easier than

recovering from selecting the wrong class altogether for an ambiguous word. Of course, this prediction is dependent on other factors in higher level syntactic processing as well.

**Semantic Priming**

Chapter four unifies disparate results from the semantic priming literature regarding the two tasks most commonly used (lexical decision and naming) in terms of the timing of activation spread through the levels of the language processor. The explanation hinges on the levels assumed to be accessed by the two tasks, and claims that the results seen can be entirely explained by the time course of activation spread between those levels inherent in the structure of the system. Levels farther away from the level accessed by the task have no effect at short delays, but have increasing effect as the duration of the prime and the delay between the prime and target increase.

**Agrammatic Aphasia**

The model explains the observed behavior of agrammatic aphasics in terms of a loss of the constraints between syntax and semantics. This explanation is similar to the one given by Linebarger et al. (1983), where the deficit is explained as one of the loss of the ability to use the syntactic information to make functional role assignments. However, the explanation given here is more specific, stated in terms of a parsing system where the interaction of syntax and semantics is limited to (a) interactions at the lexical level in the word sense buffer, and (b) constraints between the bindings of constituents to their roles in the two systems. It is the loss of the latter that is the proposed deficit. Specifically, the lesion is to the binding system which dynamically forms the correspondences between constituents in the two systems, enabling the constraint information to be communicated.

The explanation gives rise to specific predictions about aphasic behavior. These are that patients will not be responsive to syntactic attachment biases in

their semantic representation of sentences such as *the cop saw the burglar with the gun*. If they have been biased towards attaching the PP to the VP, the prediction is that they will still make the semantically more plausible attachment, unlike normals. Also, as a test of the theory's prediction that there is still a connection through the word sense buffer, agrammatics should either pick the syntactically biased interpretation of such sentences as *cast iron quickly* choosing the syntactically biased interpretation on a picture matching task (someone throwing an iron) or judge it as ungrammatical and choose a picture of a cast iron pan. If not, then the model is wrong about the word sense connection. In this case, a revised model would predict that such sentences as *the old man the boats* would be interpreted by the agrammatics as syntactically correct, while their semantic interpretation would be the semantically biased one that the sentence refers to an "old man$^{M}$ and "boats".

Finally, this model is a useful testbed for predictions about agrammatism in general. Sources of knowledge about attachment information and lexical information can be deleted from the model, and the model will then produce testable predictions, without the reprogramming that would be necessary in other AI approaches to language understanding.

## Computer Models of Parsing

One of the obvious implications of this work for parsing models is that parallelism can be at a much finer grain than has been the case in most models. This is especially pertinent at a time when massively parallel machines are being developed for AI purposes (Fahlman, 1980; Hillis 1981). Whether or not the methods used here for combining evidence for grammar rules and word meanings prove to be useful in a larger grammar, the model at least provides a framework for studying highly distributed control algorithms for parsing and semantic interpretation.

Second the model provides a new explanation of the Minimal Attachment Principle (Frazier, 1979) in terms of the timing of the spread of activation

through rules. This seems a much more natural explanation than the one given by Frazier and Fodor (1978) where the parser is broken up into a preprocessor with a small window on the input, and a more powerful second stage that can view the whole tree. The explanation given there depends on this separate first stage being limited in its view of the tree. Here, it falls out of the mechanism of spreading activation through rules[1].

Third, the work here proposes a new way of representing the interaction of syntax and semantics as an interaction between attachment preferences, combined with interaction through the representation of the lexical items themselves in the word sense buffer. In spirit this is similar to the work in Lexical Functional Grammar (Bresnan, 1982). Future work will investigate this relationship more closely.

Finally, the model adds to the growing body of literature on lexical disambiguation. The sources of disambiguating information used here are not new, being similar to those used by Hirst (1984). Also, spreading activation and lateral inhibition have also been proposed before (Pollack & Waltz, 1982; Small et al. 1982). However, this system has a more modular design than previous attempts at spreading activation parsing, and doesn't require an interpreter to build the network. By taking a cognitive modelling approach, this parser can claim to have achieved a degree of psychological reality unattained by Hirst, and by using a fixed network, attains more neurological plausibility than Pollack & Waltz.

### Inheritance in Semantic Networks

Contrary to the position taken by Etherington and Reiter (1983), we can tentatively conclude from the model described in Chapter 7 that inheritance hierarchies with exceptions can be dealt with by a massively parallel architecture. We avoided the problems uncovered by their formalization of

[1]Non-minimal attachment preferences appear to be possible based on lexical preferences associated with different verbs (Fodor, 1978). Although not elaborated here, there is a natural way to represent lexical preferences in our

NETL by using iterative activation spread rather than a single pass marking strategy suggested by the NETL architecture. Improvements to NETL suggested by Touretzky's thesis (1984) also overcome the problems, but require conditioning of the NETL network first, which can be as expensive as solving the original problem. However, more work is necessary to determine what the expected convergence time of our networks is before we can be certain that we have a useful approach. As was seen in Chapter 7, the choice of activation function can have dramatic effects on convergence time.

**Connectionist Models**

The contribution of this work to connectionist modelling has been somewhat hidden throughout the exposition. It could be characterized as a case study of the use of the unit/value principle for solving control problems. In Chapter 4, we showed that by specifying the behavior desired from the case hierarchy in terms of logical predicates on the cases, and encoding those predicates as units, The problem can be reduced to one of encoding the control between those units. This is trivial given the logical specification, since connectionist units can compute any logical combination of their inputs. This suggests circuit design would be a good prerequisite for a neophyte connectionist. However, the future of connectionist modelling critically depends on *not* having to specify networks at this level of detail; see below. Also, we can point to the parallel discrimination network in Chapter 3 as a useful tool for future connectionist models requiring standardized decisions among alternatives.

Perhaps more important is the possibility of a specification language for connectionist networks where the problem can be defined in terms of default rules (Reiter, 1980). While the mapping from arbitrary default rules to connectionist networks has not been developed, it is suggested by the work in Chapter 7, where inheritance axioms expressed in default logic were mapped

model. This is a debt to be paid in future work

into network fragments. Given a hierarchy expressed as default and first order inheritance rules, a network can be automatically generated which embodies those rules. Future work will have to determine whether such an approach can be generalized to default rules with quantified variables.

## 8.3. Future Work

Although various references to "future work" have already been made, there are still many unanswered questions that have not been mentioned. As usual with "future work" sections, many of these are weaknesses in the current work.

### Numbers

First, the number of units involved in the model presented here is cause for some concern. There are standard techniques for reducing these numbers in connectionist models, given in Feldman & Ballard (1982). It has yet to be shown that these techniques can be gracefully applied to the model presented here. One of the sources of the large number of units is the duplication of the word sense buffer for every position. This also introduces the implausibility of duplication of connection information at every point in the buffer. This is one area where the programmable buffers of McClelland (1985) have a ready application. They are designed for storing the connection information for the mapping of input features to their aggregates at the next higher level of representation. McClelland's application is letter to word mappings. The model's word sense buffer corresponds to a dictionary lookup, and as such, is a perfect candidate for McClelland's system. The connection information is represented only once in such a system, making it considerably more plausible.

A second approach to reducing the numbers is to switch to a distributed representation as is often used in neural network learning models (Hinton & Sejnowski, 1984; Kawamoto & Anderson, 1984). In this approach, rather than using a unit for every value, each unit is part of many representations.

Through the concerted action of many units, patterns of activation at one level of the system representing a word, for example, activate patterns at the next level representing its meanings. There is a limit to the number of representations that can be stored this way without crosstalk arising between them, but it appears that more information can be stored for a given number of units using this approach. One question that is yet to be answered is how control can be represented in these systems.

**Control and Evidence**

For localist connectionist networks, the "central[1]" activity that units engage in is computing the evidence for their value. This is the essential problem: an evidence theory that relies on purely local information needs to be developed. Such theories exist for some domains (Shastri, forthcoming), but the most general evidence theories (Shafer, 1976) use global normalizations that are unsuitable for a connectionist network. Once this is pinned down, the next problem is constructing a mathematical model of the conceited action of units using this evidence theory that specifies conditions for convergence. So far, the best results known for networks with very simple mathematical characterizations do not guarantee convergence to a globally coherent state (Hinton & Sejnowski, 1983b). However, this may not be necessary for a cognitive model, since it is doubtful that humans ever achieve global coherence. On the other hand, they do achieve a certain amount of coherence concerning the interpretation of day-to-day perceptual inputs.

**Cooperative Computation**

A final hurdle that has not been tackled in this system is getting the syntactic and semantic components to work together. This involves building the binding space between them that will form the bridge between syntactic and semantic components, and considering timing issues. On the first point, the work of Hirst (1984) is of interest because his use of the constraint from Vtontague semantics of a one-to-one correspondence between syntactic and

semantic objects. This should simplify the problem of computing correspondences. On the second point, it is probably unnecessary to synchronize the two systems (in fact one would expect they operate at different rates in different situations), however, their relative speeds must be somewhat in line with one another, else information will arrive too late to be useful in one system or the other. Adjusting them to one another will be undoubtedly a non-trivial task.

## 8.4. Conclusion Conclusion

Clearly, much work remains to be done. However, as a first step towards a neural network model of language comprehension, the model presented in this thesis represents a goal that seemed perhaps premature when the research began three years ago. The effort has been rewarded by a model of sentence comprehension that has many implications within the various fields that form Cognitive Science.

# BIBLIOGRAPHY

Addanki, Sanjaya. A connectionist approach to motor control. Ph.D. thesis, Computer Science Dept., U. of Rochester, 1983.

Aho, A. & Ullman, J. *Principles of Compiler Design*, Addison-Wesley, Reading, Mass., 1978.

Arbib, Michael. From AI to Neurolinguistics. In *Neural Models of Language Processes*, Michael Arbib (Ed.), COINS TR 80-09, University of Massachusetts at Amherst, 1980.

Ballard, D.H. Parameter networks. *Artificial Intelligence, 22*, 235-267, 1984. Barto, Andrew R., Anderson, C.W. and Sutton, Richard S. Synthesis of nonlinear control surfaces by a layered associative search network. *Biological Cybernetics, 43*, 1982, 175-185.

Becker, C.A. Semantic context effects in visual word recognition: An analysis of semantic strategies. *Memory and Cognition, 8*, 1980, 493-512.

Berndt, S. and Caramazza, A. A redefinition of the syndrome of Broca's aphasia: Implications for a neuropsychological theory of language. *Applied Psycholinguistics*, 1980, *1*, 225-278.

Blumenthal, A.L. Observations with self-embedded sentences. *Psychonomic Science, 6*, 453-454, 1966.

Bresnan, J. *The Mental Representation of Grammatical Relations.* Cambridge, Mass.: MIT Press, 1982.

Bruce, B. C., Case systems for natural language. *Artificial Intelligence*, December 1975, 327-360.

Buckingham, H. Lexical and semantic aspects of aphasia. In *Acquired Aphasia*, Martha Sarno (Ed.), Academic Press, New York. 1981.

Caramazza, A. & Zurif E. Dissociation of algorithmic and heuristic processes in language comprehension: Evidence from aphasia. *Brain and Language*, 1976, *3*, 572-582.

Charniak, E. A common representation for problem-solving and language-comprehension information. *Artificial Intelligence, 16*, 1981, 225-255.

Chomsky, N. *Aspects of the Theory of Syntax.* Cambridge, Mass.: MIT Press, 1965.

Collins, A. M., and Loftus, E. F. A spreading activation theory of semantic processing. *Psychological Review*, 1975, *82*, 407-428.

Conrad, C. Context effects in sentence comprehension: A study of the subjective lexicon. *Memory and Cognition*, 1974, *2*, 130-138.

Cook, W.A. Case grammar: The development of the matrix model (1970-1978). Georgetown University Press, Washington, D.C., 1979.

Corkhill, D. & Lesser, V.R. A goal directed Hearsay-II architecture: unifying data directed and goal directed control. U. Mass. T.R.

Cottrell, G. W. A model of lexical access of ambiguous words. In *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, August, 1984. (a)

Cottrell, G.W. Re: On inheritance hierarchies with exceptions. in *Proceedings of the Workshop on Non-Monotonic Reasoning*, New Paltz, N.Y., October 1984. (b)

Cottrell, G. W. and Small S. A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*, 1983, *6*, 89-120.

Crick, Francis. Thinking about the brain. *Scientific American*, September, 1979.

DeGroot, A.M.B. The range of automatic spreading activation in word priming. *Journal of Verbal Learning and Verbal Behavior*, 1983, *22*, 417-436.

Dell, G. S. Phonological and lexical encoding in speech production. Ph.D. dissertation, Dept. of Psychology, U. Toronto, 1980.

Dell, G.S. A spreading activation theory of retrieval in sentence production. University of Rochester Cognitive Science TR 21, 1984.

Dowty, D.R., Wall, R.E. and Peters, S. *Introduction to Montague Semantics*, Dordrecht: D. Reidel, 1981.

Elman, J.L. & McClelland, J.L. Speech perception as a cognitive process: The interactive activation model. In N. Lass (Ed.), *Speech and Language: Vol X*. Orlando, Florida: Academic Press, 1984.

Etherington, D. Finite default theories. M. Sc. Thesis, Dept. of C.S., University of British Columbia, 1982.

Etherington, D. Formalizing non-monotonic reasoning systems. TR 83-1, Department of Computer Science, University of British Columbia, 1983.

Etherington D. and R. Reiter On inheritance hierarchies with exceptions. in *Proceedings of the National Conference on Artificial Intelligence*, Washington, D.C., August 1983.

Fahlman, S. E. *NETL: A system for representing and using real-world knowledge*. MIT Press, Cambridge, Mass, 1979.

Fahlman, S. E. The Hashnet interconnection scheme. Technical Report, Computer Science Department, Carnegie-Mellon University, June 1980.

Fahlman. S.E., Hinton, G.E., and Sejnowski, T.J. Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. In *Proceedings of the National Conference on Artificial Intelligence*, Washington, D.C., August 1983.

Fahlman, S.E., Touretzky, D.S., and W. van Roggen. Cancellation in a parallel semantic network. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., 1981.

Feldman, Jerome A. Memory and change in connection networks. TR96, Department of Computer Science, University of Rochester, 1981(a).

Feldman, J. A. A connectionist model of visual memory. In G. E. Hinton, & J. A. Anderson (Eds.), *Parallel models of associative memory,* Hillsdale, N.J.: Lawrence Erlbaum Associates, 1981(b).

Feldman, Jerome A. Dynamic connections in neural networks. *Biological Cybernetics,* 1982, *46,* 27-39.

Feldman, Jerome A. and Dana Ballard Connectionist models and their properties. *Cognitive Science,* 1982, *6* 205-254.

Feldman, J.A. and L Shastri. Evidential Reasoning in Activation Networks. In *Proceedings of the Cognitive Science Society Conference,* Boulder, Colo., June 1984.

Fillmore, C.J. The case for case. In Bach and Harms (Eds.), *Universals in Linguistic Theory.* Holt, Rinehart and Winston, 1968.

Finin, T.W. The semantic interpretation of compound nominals. TR 96, Coordinated Science Laboratory, University of Illinois-Urbana, 1980.

Fischler, I. Associative facilitation without expectancy in a lexical decision task. *Journal of Experimental Psychology: Human Perception and Performance, 3,* 1977(a), 18-26.

Fischler, I. Semantic facilitation without association in a lexical decision task. *Memory and Cognition, 5,* 1977(b), 335-339.

Fodor, Janet Dean. Parsing strategies and constraints on transformations. *Linguistic Inquiry, 9,* 1978, 427-473.

Forster, K. I. Levels of processing and the structure of the language processor. In W.E. Cooper & E.C.T. Walker (Eds.), *Sentence Processing.* Hillsdale, NJ: Erlbaum, 1979.

Foss, D. and Jenkins, C. Some effects of context on the comprehension of ambiguous sentences. *Journal of Verbal Learning and Verbal Behavior,* 1973, *12,* 577-589.

Frazier, Lyn. On comprehending sentences: Syntactic parsing strategies. Ph.D. Thesis, University of Connecticut, 1978.

Frazier, Lyn and Fodor, Janet Dean. The sausage machine: A new two-stage parsing model. *Cognition,* 6, 1978, 291-325.

Garrett M.F. Word and sentence perception. In R. Held, H.W. Leibowitz, and H-L Teuber (Eds.), *Handbook of Sensory Physiology. Vol. Vlll: Perception.* Berlin: Springer-Verlag, 1978.

Garrett, M.F. Levels of processing in sentence processing. In B. Butterworth (Ed.) *Language Production. I,* New York: Academic Press, 1980.

Gentner, D. Some interesting differences between nouns and verbs. *Cognition and Brain Theory,* 1982, *4* (2).

Gigiey, H. M. Neurolinguistically constrained simulation of sentence comprehension: Integrating Artificial Intelligence and Brain Theory. Ph.D. Thesis, Department of Computer and information Science, University of Massachusetts, September 1982.

Ginsberg, Matthew W. Non-monotonic reasoning using Dempster's rule.. In *Proceedings of the National Conference on Artificial Intelligence,* Austin, Texas, August, 1984.

Gleason, J.B., Goodglass, H., Green, E., Ackerman, N., and Hyde, M.R. The retrieval of syntax in Broca's aphasia. *Brain and Language,* 1975, *2,* 451-471.

Goodglass, H. & Baker, E. Semantic field, naming, and auditory comprehension in aphasia. *Brain and Language,* 1976, *3,* 359-374.

Hayes, P. J. In defense of logic. In *Proceedings of the Fifth Annual International Joint Conference on Artificial Intelligence,* Cambridge, Mass., 1977.

Heilman, K.M. and Scholes, RJ. The nature of comprehension errors in Broca's, conduuction and Wernicke's aphasics. *Cortex, 12,* 1976, 258-265.

**Hillinger,** M.L. Priming effects with phonmically similar words: The encoding-bias hypothesis reconsidered *Memory and Cognition, 8,* 1980, 115-125.

Hillis, W. D. The connection machine (computer architecture for the New Wave). Memo #646, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1981.

Hinton, G. E. Shape representation in parallel systems. In *Proceedings of the Seventh Annual International Joint Conference on Artificial Intelligence,* Vancouver, B. C, August 1981.

Hinton, G. E, and Anderson, J. A. (Eds.), *Parallel models of associative memory,* Hillsdale, NJ: Lawrence Erlbaum Associates, 1981.

Hinton, G.E. and T. Sejnowski. Analyzing cooperative computation. In *Proceedings of the Fifth Annual Cognitive Science Society Conference,* Rochester, N.Y., May 1983 (a).

Hinton, G.E. and T. Sejnowski. Optimal perceptual inference. *Proceedings of the IFEE Computer Society Conference on Computer Vision and Patttern Recognition.* Washington, D.C. 1983 (b).

Hinton, G.E. and T. Sejnowski. Learning semantic features. In *Proceedings of the Sixth Annual Cognitive Science Society Conference,* Boulder, Colorado, June, 1984.

Hirst, G. Semantic interpretation against ambiguity. Ph.D. dissertation, Brown University, 1983.

Holmes, V.M. Prior context and the perception of lexically ambiguous sentences. *Memory and Cognition,* 1977, 5, 103-110.

Hrechanyk, Lydia M. and Ballard, D. H. A connectionist model of form perception. In *Proceedings of the IEEE Workshop on Computer Vision*, Ringde, New Hampshire, August 23-25, 1982.

Hudson, S. & Tanenhaus, M. Ambiguity resolution in the absence of contextual bias. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, June, 1984.

Kawamoto, Alan and Anderson, James A. Lexical access using a neural network. In *Proceedings of the Sixth Annual Cognitive Science Society Conference*, Boulder, Colorado, June, 1984.

Kean, M.L. Three perspectives for the analysis of aphasic syndromes. In M. Arbib (Ed.) *Neural Models of Language Processes*, proceedings of a conference held at U. of Massachusetts, at Amherst, November, 1979.

Keil, F. "Learning word meanings." Talk presented at the University of Rochester, 1984.

Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P. Optimization by simulated annealing. *Science, 220*, 1983, 671-680.

Kolk, H. Judgement of sentence structure in Broca's aphasia. *Neuropsychologia, 16*, 1978, pp. 617-625.

Koriat, A. Semantic facilitation in lexical decisions as a function of prime-target association. *Memory and Cognition*, 1981, *9*, 587-598.

Kucera, H. and Francis W.N. *Computational Analysis of Present-Day American English.* Providence, R.I.: Brown University Press, 1967.

Lackner and Garret. Resolving ambiguity: Effects of biasing context in the unattended ear. *Cognition*, 1972, *1*, 359-372.

Lenneberg, E.H. The neurology of language. *Daedalus*, 1973, *102*, 115-133.

Lesser, V.R., and Erman, L. D. A retrospective view of the Hearsay-II architecture. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 1977.

Levy, William B. Associate encoding at synapses. In *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, Ann Arbor, Michigan, August 4-6, 1982.

Linebarger, M.C., Schwartz, M.C., and Saffran, E.M. Sensitivity to grammatical structure in so-called agrammatic aphasics. *Cognition*, 1983, *13*, 361-392.

Lucas, Margery. <<>> Ph.D. Thesis, Department of Psychology, University of Rochester, 1984.

Luria, A.R. Language and brain. *Brain and Language*, 1974, *1*, 1-14.

Marcus, Mitchell P. An overview of a theory of syntactic recognition for natural language. Memo #531, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1979.

Marslen-Wilson W.D. and Tyler, L.K. The temporal structure of spoken language understanding. *Cognition*, 1980, *8*, 1-71.

McCarthy, John. Applications of circumscription to formalizing common sense knowledge. To appear (1984).

McClelland, J.L. On the time relations of mental processes: An examination of systems of processes in cascade. *Psych. Review*, *86*,

McClelland, J. L. Putting knowledge in its place: A scheme for programming parallel processing structures on the fly. *Cognitive Science*, *9*, 113-146, 1985.

McClelland, James L. and David E. Rumelhart. An interactive activation model of the effect of context in perception: Part I, An account of basic findings. *Psychological Review*, *88*,

Meyer, D. E., and Schvaneveldt R. W. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 1971, *90*,

Meyer, D. E., Schvaneveldt R. W. and Ruddy, M.G. Loci of contextual effects on visual word recognition. In P.M.A. Rabbitt & S. Dornic (Eds.), *Attention and Performance V*, New York: Academic Press, 1975.

Milne, Robert. Predicting garden path sentences. *Cognitive Science*, *6*, 1982.

Milne, Robert. Resolving lexical ambiguity in a deterministic parser. Ph.D. thesis. University of Edinburgh, 1983.

Minsky, M. A framework for representing knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill, 1975.

Minsky, M., & Papert S. *Perceptrons*. Cambridge, Mass.: MIT Press, 1972.

Morton, J. Interaction of information in word recognition. *Psychological Review*, 1969, *76*,

Newman, J. E. and G. S. Dell. The phonological nature of phoneme monitoring: A critique of some ambiguity studies. *Journal of Verbal Learning and Verbal Behavior*, 1978, *6*, 364-371.

Neely, James H. Semantic priming and retrieval from memory: Roles of inhibitionless spreading activation and limited-capacity attention. *Journal of Experimental Psychology: General*, 1977, *106*, 226-254.

Onifer, William and Swinney, David A. Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias. *Memory and Cognition*, 1981, *9*, 225-236.

Pavio, A. *Imagery and Verbal Processes*. New York: Holt, 1971.

Pollack, Jordan and Waltz David. Natural language processing using spreading activation and lateral inhibition. In *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, Ann Arbor, Michigan, August, 1982.

Pollack, Jordan and Waltz David. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science. 9,* 51-74, 1985.

Posner, M.I. & Snyder, C.R.R. Attention and cognitive control. In R.L Solso (Ed.), *Information Processing and Cognition: The Loyola symposium.* Hillsdale, N. J.: Erlbaum, 1975.

Quillian, M. Ross. Semantic memory. Unpublished Ph.D. Thesis, Carnegie Institute of Technology, 1966.

Quillian, M. Ross. The teachable language comprehended A simulation program and theory of language. *Communications of the ACMFp, 1969, 12, 459-476.*

Rayner, K. Carlson, M. and Frazier, L. The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior, 22,* 358-374, 1983.

Reiter, Ray. A logic for default reasoning. *Artificial Intelligence 13,* 1980, pp 81-132.

Rich, E. Default reasoning as likelihood reasoning. In *Proceedings of the National Conference on Artificial Intelligence,* Washington, D.C., August 1983.

Riesbeck, Christopher K. Computational understanding: Analysis of sentences and context. Memo 238, Artificial Intelligence Laboratory, Stanford University, 1974.

Riesbeck, Christopher K., and Roger C. Schank. Comprehension by computer: Expectation-based analysis of sentences in context. Research Report #78, Department of Computer Science, Yale University, 1976.

Rosch, E. On the internal structure of perceptual and semantic categories. In T.E. Moore (Ed.), *Cognitive development and acquisition of language.* New York, Academic Press, 1973.

Rosch, E. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General,* 1975, *104,* 192-233.

Rosenfeld, Azriel, R. A. Hummel, and Steven W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics, 6,* 1976.

Rumelhart, D. E., & McClelland, J. L An interactive activation model of context effects in letter perception, Part 2, The contextual enhancement effect and some tests and extensions of the model *Psychological Review, 89,* 1982, 60-94.

Rumelhart, D. E., & Norman, D. A. Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science, 6,* 1982, 1-36.

Sabbah, Daniel. A connectionist approach to visual recognition. TR 107 and Ph.D. thesis. Computer Science Dept., U. of Rochester, April 1982.

Sabbah, Daniel. Computing with connections in visual recognition of Origami objects. *Cognitive Science,* 9, 1985.

Saffran, Eleanor M. Neuropsychological approaches to the study of language. *British Journal of Psychology,* 1982, *73,* 317-337.

Saffran, E.iM., Schwartz, M.F. & Mann, O.S.M. (1980a) The word order problem in agrammatism II. Production. *Brain and Language, 10,* 263-280.

Schank, R, Conceptual Dependency: A theory of natural language understanding. *Cognitive Psychology,* i, 1972, 552-631.

Schreuder, $R_M$ Flores d'Arcais, G.B. and Glazenborg, G. Effects of perceptual and conceptual similarity in semantic priming. *Psychological Research, 45,* 1984, 339-354.

Schwartz, M.F., Saffran, E.M., and Marin O.S.M. The word order problem in agrammatism I. Comprehension. *Brain and Language,* 1980, *10,* 249-262.

Seidenberg, M. S., Tanenhaus M., Leiman, J. and Bienkowski, M. Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Cognitive Psychology,* 1982, *14,* 489-537.

Seidenberg, M.S., Waters, G.S., Sanders, M. and Langer, P. Pre- and postlexical loci of contextuual effects on word recognition. *Memory and Cognition, 12,* 1984, 315-328.

Seidenberg, M.S., Waters, G.S., Barnes, M.A. and Tanenhaus, M.K. When does irregular spelling or pronounciation influence word recognition? *Journal of Verbal Learning and Verbal Behavior, 23, 1984, 383-404.*

Selman, Bart and Hirst, Graeme. A rule-based connectionist parsing system. Submitted to the 1985 Cognitive Science Conference.

Shafer, G. *A Mathematical Theory of Evidence.* Princeton: Princeton University Press, 1976.

Shastri, L and Feldman, J.A. Semantic networks and neural nets. T.R. 131, Dept. of Computer Science, University of Rochester, May 1984.

Simon, H.A. *The Sciences of the Artificial.* Cambridge, Mass.: MIT Press. 1969.

Small, S. Conceptual language analysis for story comprehension. TR 663, U. of Maryland, 1978.

Small, S. L. Word expert parsing: A theory of distributed word-based natural language understanding. Ph.D. dissertation and TR 954, Dept. of Computer Science, U. Maryland, 1980.

Small, Steven L. Exploded connections: Unchunking schematic knowledge. In *Proceedings of the Fourth Annual Conference of the Cognitive Science Society,* Ann Arbor, Michigan, August 4-6, 1982.

Small, Steven L. and Chuck Rieger. Parsing and comprehending with Word Experts (A theory and its realization). In *Strategies for Natural Language Processing,* Lehnert and Ringie (Eds.), Lawrence Erlbaum Associates. 1982.

Small, Steven L. and Margery Lucas. A computer model of sentence comprehension. Cognitve Science TR #1, University of Rochester, 1984.

Small, S. L., Shastri L., Brucks M., Kaufman S., Cottrell, G., and Addanki, S. ISCON: An interactive simulator for connectionist networks. Technical Report 109, Department of Computer Science, University of Rochester, Dec. 1982.

Smith, E. E., Shoben, E. J., and Rips, L. J. Structure and process in semantic memory: A featural model for semantic decision. *Psychological Review*, 1974, *81*, 214-241.

Stanovich, K. and West, R.F. On priming by a sentence context. *Journal of Experimental Psychology: General, 112*, 1983, 1-36.

Sternberg, S. The discovery of processing stages: Extensions of Donder's method. In W.G. Koster (Ed.), *Attention and Performance II*. Amsterdam: North-Holland, 1969.

Swinney, David A. (1979), Lexical access during sentence comprehension: (Re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, 1979, *18*, 645-660.

Swinney, David A. The information and time-course of information interaction during speech comprehension, lexical segmentation, access, and interpretation. In J. Mehler, E.C.T. Walker, and M. Garrett (Eds.) *Perspectives on Mental Representation*. Hillsdale, N.J.: LEA 1982.

Swinney, D. A., and Hakes, D.T. Effects of prior context upon lexical access during sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, 1976, *15*, 681-689.

Swinney, David A., William Onifer, Penny Prather, and Max Hirshkowitz. Semantic facilitation across sensory modalities in the processing of individual words and sentences. *Memory and Cognition*, 1979, *7*, 159-165.

Tanenhaus, M., Leiman, J., and Seidenberg, M. S. Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior*, 1979, *18*, 427-440.

Touretzky, David S. The mathematics of inheritance systems. Ph.D. Thesis, Carnegie Mellon University, 1984. Available as T.R. CMU-CS-84-136.

Walker, Donald E. (Ed.) *Understanding Spoken Language*. North-Holland: New York, 1978.

Warren, R.E. Stimulus encoding and memory. *Journal of Experimental Psychology, 94*, 1972, 90-100.

Warren, R.E. Time and the spread of activation in memory. *Journal of Experimental Psychology: Human Learning and Memory, 3*, 1977, 458-466.

Weigel-Crump, C. and Koenigsknecht, R.A. Tapping the store of the adult aphasic: Analysis of the improvement made in word retrieval skills. *Cortex*, 9, 1973, 410-417.

West, R.F. and Stanovich, K.E. Source of inhibition in experiments on the effect of sentence context on word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8*, 1982, 385-399.

Wilks, Y., Parsing English II. In Charniak and Wilks (Eds.), *Computational Semantics.* North-Holland, 1976, pp. 155-184.

Winograd, Terry. Procedures as a representation for data in a computer program for understanding natural language. T.R. 84, MIT Project MAC.

Winograd, Terry. *Language as a Cognitive Process. Volume I: Syntax.* Addison-Wesley: Reading, Massachusetts, 1983.

Yates, J. Priming dominant and unusual senses of ambiguous words. *Memory and Cognition*, 1978, *6*, 636-643.

Zurif, E.B., and Caramazza A. Psycholinguistic structures in aphasia: studies in syntax and semantics. In *Perspectives in Neurolinguistics and Psycholinguistics. Vol. I*, H. Whitaker and H. A. Whitaker (eds.), Academic Press, 1976, pp. 261-303.

Zurif, E., Caramazza, A. Myerson, R., and Galvin, J. Semantic feature representations for normal and aphasic language. *Brain and Language*, 1974, *1*, 167-187.

# APPENDIX 1

## Unit Rules for Example in Chapter 4

### A.1.1 Notation

The following is pseudocode for each node types' output. For simplicity, we use integer inputs and outputs. Most of the variable names are self-explanatory, but here aree a few hints:

window == a "competition window", as in Chapter 5's rule competition networks. Assumed to be 2 in the example.

bottom_up input == maximum of bottom-up input (from buffer nodes for s.n. leaves). When from outside the example network, as in the case of buffer nodes, assumed to be 4.

inhibition == the maximum of the evidence for competitiors (not their output, as in the rule evidence networks of Ch. 5)

my_evidence == this nodes' supportive input (sum of evidence)

### A.1.2 Noun Concept Buffer Nodes

These are the nodes in the word sense buffer that are connected into the semantic network and to the binding nodes. MALE-HUMAN/CONC1 is one of these, for example. They threshold their feedback from the semantic network and from the binder nodes.

```
if (activated by bottom-up input) then
 { /* Threshold feedback from semantic net      */
   sem_net_feedback = sem_net_feedback - 4;
   /* Ditto for binders                 */
   binder_feedback = binder_feedback - 4;
   my_evidence = bottom_up + sem_net_feedback + binder_feedback;

   return (decision (inhibition,my_evidence);
  }
return(0);
```

The following decision function is used by several nodes.

```
integer function decision(inhibition, support)

 diff = inhibition - support;

                    /* The difference between my      */
                    /* evidence and my competitors'  */

 if (diff < 0) then return (support);

                    /* Ignore competitors' evidence   */
                    /* unless it is > mine            */

 if (inhibition < window) then return (support - diff);

                    /* Still within competition */
                    /* window? keep going, but        */
                    /* lower output by competitor's advantage */

 return(0);         /* All other cases: give up */
```

### A.1.3 Verb Concept Buffer Nodes

These are the word sense buffer nodes corresponding to a verb. Their support depends crucially on the presence of an Object filler.

```
if activated by bottom-up input then
 {my_evidence = bottom_up input;
  if (feedback from the Agent hier.) then /* Add one for Agent      */
       my_evidence = my_evidence + 1;
  if (feedback from an Object) then              /* Add one for Object
  else my_evidence = my_evidence - 1;  /* Penalize by one if no Object

  return (decision(my_evidence, inhibition));
 }
return(0);
```

### A.1.4 Semantic Network Nodes

```
if (activated by bottom-up input) then
 {superordinate = superordinate - 4;      /• Threshold feedback from sup. */
  my_evidence = superordinate + bottom-up input;
  if (feedback from case hierarchy) then my_evidence = my_evidence + 1;
     /* This means a case was satisfied */

  return (my_evidence);
 }
return(O);              /* All other cases    */
```

A. 1.5 Case Nodes

These are as in the description in the text. The $^{ff}pr^{M}$ in the table just means that a predicate (verb) attached to this case is firing. This does not cause any output from the case nodes (by itself).

A.i.6 Binder Nodes

```
if (activated by buffer node) then

 {
 bottom_up = max(0, buffer_node_output - 5)
                    /* if buffer_node_puput > 5, must be      */
                    /* due to semantic network feedback       V
                    /* to buffer node, rather than from me    */

 my_evidence = 4 + bottomjjp; /* Arbitrary working amount */

 if (the root of the attached case hierarchy is satisfied + ) then
       my_evidence = my_evidence + 1:

 return (decision (my_evidence, inhibition));

 }
return(O);              /* All other cases: give up  */
```