Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition

by

Jeffrey Mark Siskind

B.A. Computer Science (1979)
Technion, Israel Institute of Technology

S.M. Electrical Engineering and Computer Science (1989)
Massachusetts Institute of Technology

Submitted to the Department of
Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the
Massachusetts Institute of Technology
January 1992

© 1992 Massachusetts Institute of Technology. All rights reserved.

Signature of Author _____
Department of Electrical Engineering and Computer Science
January 24, 1992

Certified by _____
Robert C. Berwick
Associate Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by _____
Campbell L. Searle
Chair, Department Committee on Graduate Students

Revised April 13, 1993

Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition

by
Jeffrey Mark Siskind

Submitted to the Department of Electrical Engineering and Computer Science
on January 24, 1992, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Revised April 13, 1993

Abstract

This thesis addresses two questions related to language. First, how do children learn the language-specific components of their native language? Second, how is language grounded in perception? These two questions are intimately related. One piece of language-specific information which children must learn is word meanings. Knowledge of the meanings of utterances containing unknown words presumably aids children in the process of determining the meanings of those words. A complete account of such a process must ultimately explain how children extract utterance meanings from their non-linguistic context. In the first part of this thesis I present precisely formulated algorithms which attempt to answer the first question. These algorithms utilize a cross-situational learning strategy whereby the learner finds a language model which is consistent across several utterances paired with their non-linguistic context. This allows the learner to acquire partial knowledge from ambiguous situations and combine such partial knowledge across situations to infer a unique language model despite the ambiguity in the individual isolated situations. These algorithms have been implemented in a series of computer programs which test this cross-situational learning strategy on linguistic theories of successively greater sophistication. In accord with current hypotheses about child language acquisition, these systems use only positive examples to drive their acquisition of a language model. MAIMRA, the first program described, learns word-to-meaning and word-to-category mappings from a corpus pairing utterances with sets of expressions representing the potential meanings of those utterances hypothesized by the learner from the non-linguistic context. MAIMRA's syntactic theory is embodied in a fixed context-free grammar. DAVRA, the second program described, extends MAIMRA by replacing the context-free grammar with a parameterized variant of $\overline{\mathrm{X}}$ theory. Given the same corpus as MAIMRA, DAVRA learns the parameter settings for $\overline{\mathrm{X}}$ theory in addition to a lexicon mapping words to their syntactic category and meaning. DAVRA has been successfully applied, without change, to tiny corpora in both English and Japanese, learning the requisite lexica and parameter settings despite differences in word order between the two languages. KENUNIA, the third program described, incorporates a more comprehensive model of universal grammar supporting movement, adjunction, and empty categories, as well as more extensive parameterization of its $\overline{\mathrm{X}}$ theory component. This model of universal grammar is based on

recent linguistic theory and includes such notions as the DP hypothesis, VP-internal subjects, and V-to-I movement. KENUNIA is able to learn the parameter settings of this model, as well as word-to-category mappings, in the presence of movement and empty categories. The algorithms underlying MAIMRA, DAVRA, and KENUNIA are presented in detail along with annotated examples depicting their operation on sample learning tasks.

In the second part of this thesis I present a novel approach to event perception, the processes of determining when events described by simple spatial motion verbs such *throw*, *pick up*, *put*, and *walk* occur in visual input. This approach is motivated by recent experimental studies of adult visual perception and infant knowledge of object permanence. In formulating this approach I advance three claims about event perception and the process of grounding language in visual perception. First, I claim that the notions of support, contact, and attachment play a central role in defining the meanings of simple spatial motion verbs in a way that delineates prototypical occurrences of events described by those verbs from non-occurrences. Prior approaches to lexical semantic representation focussed primarily on movement and lacked the ability to incorporate these crucial notions into the definitions of simple spatial motion verbs. Second, I claim that support, contact, and attachment relations between objects are recovered from images by a process of counterfactual simulation. For instance, one object supports another object if the latter does not fall when the short-term future of the image is predicted, but does fall if the former is removed. Such counterfactual simulations are performed by a modular imagination capacity. Third, I claim that this imagination capacity, while superficially similar in intent to traditional kinematic simulation, is actually based on a drastically different foundation. This foundation takes the process of enforcing naive physical constraints such as substantiality, continuity, and attachment relations between objects to be primary. In doing so it sacrifices physical accuracy and coverage. This is in contrast to the traditional approach which achieves physical accuracy and coverage by numerical integration, relegating the maintenance of constraints to a process of secondary importance built around the numerical integration core. A simplified version of this theory of event perception has been implemented in a program called ABIGAIL which watches a computer-generated animated movie and produces a description of the objects and events which occur in that movie. ABIGAIL's event perception processes rely on counterfactual simulation to recover changing support, contact, and attachment relations between objects in the movie. Prior approaches to this task were based solely on determining the spatial relations between objects in the image sequence, grounding verb meanings in static geometric predicates used to compute those spatial relations without counterfactual analysis. The detailed algorithms underlying the novel implementation are presented along with annotated examples depicting its analysis of sample movies.

Thesis Supervisor: Robert C. Berwick
Title: Associate Professor of Computer Science and Engineering

מוקדש לאמי מורתי
עיטא בת אברהם הכהן ז"ל
ת.נ.צ.ב.ה.

Dedicated to the memory of my mother:
Edna Roskin Siskind

Acknowledgments

Last week sometime, I met a young girl while eating at the KK. To spark conversation, I asked her what grade she was in, to which she promptly replied 'first, and you?'. Feeling obliged to respond, I thought for a moment and answered 'twenty-fifth'. Yes, I am in twenty-fifth grade, give or take a few depending on how you count. Family members and friends often prod me asking when I am going to finish my thesis, get out of school, and get a real job. That just misses the point: there is little more to life than learning—and children. Perhaps that is why I choose child language acquisition as my research topic.

The acknowledgment page for my masters thesis ended with the following statement.

> I am far too ashamed to acknowledge my family, friends, and teachers on such a meager and paltry document. That will have to wait for my Ph.D. Perhaps then I can present a document worthy of acknowledging them.

Getting my masters degree was a long and arduous task. Both the research and the writing were unfulfilling. Getting my Ph.D. was different. The research was pleasurable—writing the document was agonizing. One thing I can say, however, is that I am much more content with the results. So I feel comfortable delivering on my past promise.

Before doing that, however, I must first gratefully acknowledge the generous support I have received for my research from a number of sources during my graduate career. Those sources which contributed to the work described in this document include AT&T, for a four year Ph.D. fellowship, Xerox Corporation, for affording me the opportunity to work on this project during my more recent visits to PARC, a Presidential Young Investigator Award to Professor Robert C. Berwick under National Science Foundation Grant DCR–85552543, a grant from the Siemens Corporation, and the Kapor Family Foundation. The fact that this generous funding has come with no strings attached has has allowed me to approach this research with child-like playfulness and to continue schooling through the twenty-fifth grade. I am indebted to Patrick Winston, the principal of the MIT AI Lab, who along with Peter Szolovits and Victor Zue, encouraged me to pursue this research.

While writing my masters thesis, I wasted several hours of Xerox's money recalling the names of my grade school teachers and preparing the LaTeX that prints the next page. That list omits many other teachers, as well as family and friends, whose names are far too numerous to list. Their names are included herein by reference (or however you say that in legalese). Several people however, deserve special mention. My parents, for long ago giving up hope that I would ever finish; Naomi, Yaneer, Shlomiya, Yavni, and Maayan Bar-Yam, Tova, Steve, Asher, and Arie Greenberg, and Avi, Andi, Zecharia, and Yael Klausner, for being there to comfort me when I thought I would never finish; and Jeremy Wertheimer, Jonathan Amsterdam, and Carl de Marcken for proofreading this document as I was finishing. Jeremy and Carl helped unwedge me at key times during the research and writing that went into this thesis.

Lui Collins once commented that she was fortunate to have never been so unwise as to incorporate the name of a lover in a song she had written, thus avoiding the pain and embarrassment of needing to perform that song after the relationship was over. At the risk of being foolish, there is one additional friend whom I would like to thank explicitly. Beth Kozinn has been a source of support throughout much of this thesis project. She was one of the few people who understood and appreciated the passion I have for my work and was even fascinated by the idea that I spent my days—and nights—trying to get a computer to understand cartoons so simple that any two year old would find them boring. Yet she herself always wanted to know what John and Mary were doing in frame 750. Beth taught me a lot about contact. This helped me get through the anguish of applying for—and not being offered—an academic position. I wish I could teach her something about attachment.

In summary I attribute this thesis to all of my teachers, from Ms. Solomon, my nursery school teacher, through David McAllester and Bob Berwick, my twenty-fifth grade teachers, to Lila Gleitman, who will be my twenty-sixth grade teacher. Someone once said that the only reason people have kids

is so that they have an excuse to break out of the pretense of being an adult and to act like a child in their presence. Perhaps the only reason people go to school is for the privilege and obligation for acknowledging and honoring one's teachers. I hope that childhood and schooling never cease so that I will forever retain that privilege and obligation.

To:

Ms. Solomon, my nursery school teacher
Mrs. Miller, my kindergarten teacher
Mrs. Savrin, my first grade teacher
Miss Kallman, my second grade teacher
Mrs. Theodor, my third grade teacher
Mrs. Keogh, my teacher for EAP I and II

my English teachers:

Mrs. Goldberg, for seventh grade
Mr. Bershaw, for eighth grade
Mr. Iglio, for ninth grade
Mrs. Johnson, for tenth grade
Mr. Kraus, for eleventh grade
Mr. Taussig, for twelfth grade
Mr. Guy, for twelfth grade

my Spanish teachers:

Miss di Prima, for seventh grade
Miss Murphy, for eighth grade
Mrs. Gomez, for tenth grade

my Social Studies teachers:

Mrs. Caldera, for seventh grade
Mr. Markfield, for eighth grade
Mr. Lerman, for ninth grade
Mrs. Cohen, for tenth grade
Dr. Kelly, for eleventh grade

my Music teachers:

Mr. Sepe,
Mr. de Silva,
Mr. Carubia,
Mr. Katz,

my Science teachers:

Mrs. Baron, for seventh grade
Mr. Sponenberg, for Biology
Mr. Palazzo, for Physics

my Mathematics teachers:

Mr. Gould, for seventh grade
Mr. Day, for Algebra
Mr. Fitzgerald, for Geometry
Mr. Okun, for Trigonometry
Mr. Metviner, for eleventh grade
Mr. Gorman, for Calculus

and with particular respect and fondness to:

Mr. Silver, my eighth grade Science teacher, from whom I learned conviction, cynicism, precision, and the love of the pursuit of knowledge.

and to:

Mr. Gerardi, my Chemistry teacher and mentor for teacher-student relationships. May I be so fortunate to relate to my future students, and be their role model, as well as he related to his.

and to:

Mrs. Jagos, my ninth grade Spanish teacher, for teaching me friendship, and how to be a *mensch*.

Contents

1 Overview **11**

I Language Acquisition **23**

2 Introduction **25**
2.1 The Bootstrapping Problem . . . 26
2.2 Outline . . . 31

3 Cross-Situational Learning **33**
3.1 Linking and Fracturing . . . 33
3.2 Learning Syntactic Categories . . . 37
3.3 Learning Syntactic Categories and Word Meanings Together . . . 39

4 Three Implementations **47**
4.1 Maimra . . . 48
4.2 Davra . . . 55
4.2.1 Alternate Search Strategy for Davra . . . 63
4.3 Kenunia . . . 66
4.3.1 Overview of Kenunia . . . 66
4.3.2 Linguistic Theory Incorporated in Kenunia . . . 68
4.3.3 Search Strategy . . . 73
4.3.4 The Parser . . . 74
4.3.5 Additional Restrictions . . . 77
4.3.6 Kenunia in Operation . . . 78

5 Conclusion **83**
5.1 Related Work . . . 83
5.1.1 Semantic Bootstrapping . . . 83
5.1.2 Syntactic Bootstrapping . . . 86
5.1.3 Degree 0+ Learning . . . 87
5.1.4 Salveter . . . 88
5.1.5 Pustejovsky . . . 89
5.1.6 Rayner et al. . . . 89
5.1.7 Feldman . . . 90
5.2 Discussion . . . 91

II Grounding Language in Perception **95**

6 Introduction **97**
6.1 The Event Perception Task . . . 100
6.2 Outline . . . 104

7 Lexical Semantics **105**

8 Event Perception **121**
8.1 The Ontology of Abigail's Micro-World . . . 122
8.1.1 Figures . . . 123
8.1.2 Limitations and Simplifying Assumptions . . . 124
8.1.3 Joints . . . 127
8.1.4 Layers . . . 128
8.2 Perceptual Processes . . . 129
8.2.1 Deriving the Joint and Layer Models . . . 129
8.2.2 Deriving Support, Contact, and Attachment Relations . . . 140
8.3 Experimental Evidence . . . 150
8.4 Summary . . . 156

9 Naive Physics **157**
9.1 Simulation Framework . . . 160
9.2 Translation and Rotation Limits . . . 165
9.3 Complications . . . 180
9.3.1 Clusters . . . 181
9.3.2 Tangential Movement . . . 183
9.3.3 Touching Barriers . . . 186
9.3.4 Tolerance . . . 188
9.4 Limitations . . . 188
9.5 Experimental Evidence . . . 191
9.6 Summary . . . 198

10 Conclusion **199**
10.1 Related Work . . . 199
10.1.1 Kramer . . . 199
10.1.2 Funt . . . 201
10.2 Discussion . . . 202

A Maimra in Operation **205**

B Kenunia in Operation **235**

C Abigail in Operation **267**

List of Figures

1.1 A generic language processing architecture . . . 12
1.2 The English corpus presented to DAVRA . . . 17
1.3 DAVRA's output . . . 18
1.4 ABIGAIL's movie . . . 20
1.5 Imagining frame 11 . . . 21

2.1 A generic language processing architecture . . . 28
2.2 Cross-situational learning architecture . . . 29

3.1 Jackendoff's linking rule . . . 35
3.2 Analyses of five and six word utterances . . . 40
3.3 Weak cross-situational learning of syntactic categories . . . 41
3.4 Consistent but incorrect analyses after strong cross-situational syntax . . . 42
3.5 All submeanings of the meaning expressions in the sample corpus . . . 43
3.6 Word meanings inferred by weak cross-situational semantics . . . 44

4.1 MAIMRA's grammar . . . 48
4.2 The English corpus presented to MAIMRA and DAVRA . . . 50
4.3 MAIMRA's output . . . 54
4.4 DAVRA's search strategy . . . 58
4.5 DAVRA's output for the English corpus . . . 62
4.6 The Japanese corpus presented to DAVRA . . . 63
4.7 DAVRA's output for the Japanese corpus . . . 64
4.8 KENUNIA's corpus . . . 69
4.9 KENUNIA's parser . . . 75
4.10 KENUNIA's prior semantic knowledge . . . 78
4.11 KENUNIA's output . . . 80

5.1 The technique used by Rayner et al. (1988) . . . 90

6.1 A typical movie frame . . . 98
6.2 ABIGAIL's language faculty . . . 99
6.3 A movie script . . . 102
6.4 ABIGAIL's movie . . . 103

7.1 Different varieties of support relationships . . . 111
7.2 Jackendoff's linking rule . . . 117
7.3 Borchardt's definitions . . . 119

8.1 The touch and overlap relations . . . 126
8.2 Event perception architecture . . . 130
8.3 The algorithm for updating the joint model . . . 133
8.4 The first twelve frames of ABIGAIL's movie . . . 135
8.5 Imagining frame 0 with empty joint and layer models . . . 136
8.6 Hypothesized joint model . . . 137
8.7 Hypothesized layer model . . . 138
8.8 Imagining frame 11 . . . 141
8.9 A short movie . . . 143
8.10 Event graph for the short movie . . . 144
8.11 Perceptual primitives recovered for the short movie—I . . . 145
8.12 Perceptual primitives recovered for the short movie—II . . . 146
8.13 Event graph for ABIGAIL's movie . . . 148
8.14 Imagining frame 172 . . . 151
8.15 Three tables collectively supporting an object . . . 152
8.16 Experiment 1 from Freyd et al. (1988) . . . 153
8.17 Experiment 2 from Freyd et al. (1988) . . . 154

9.1 One step continuous simulation . . . 159
9.2 Sliding . . . 162
9.3 Falling over . . . 164
9.4 Translating a line segment f until its endpoint $p(f)$ touches another line segment g . . . 167
9.5 Translating a line segment f until its endpoint $p(f)$ touches the endpoint $p(g)$ of another line segment g . . . 168
9.6 Translating a circle f until it is tangent to a line segment g . . . 169
9.7 Translating a line segment f until its endpoint $p(f)$ touches a circle g . . . 170
9.8 Translating a line segment f until its endpoint $p(f)$ touches a circle g . . . 171
9.9 Translating a circle f until blocked by another circle g when f and g are outside each other 172
9.10 Translating a circle f until blocked by another circle g when f is inside g . . . 173
9.11 Rotating a line segment f until its endpoint $p(f)$ touches another line segment g . . . 174
9.12 Rotating a line segment f until its endpoint $p(f)$ touches the endpoint $p(g)$ of another line segment g . . . 175
9.13 Rotating a circle f until it is tangent to a line segment g . . . 177
9.14 Rotating a line segment f until its endpoint $p(f)$ touches a circle g . . . 178
9.15 Rotating a circle f until blocked by another circle g when f and g are outside each other 179
9.16 Rotating a circle f until blocked by another circle g when f is inside g . . . 180
9.17 Situations requiring clusters . . . 182
9.18 Tangential translation . . . 184
9.19 Tangential rotation . . . 185
9.20 Barriers . . . 187
9.21 Coincident line segments . . . 189
9.22 Roundoff errors can cause substantiality violations . . . 190
9.23 Imagination limitations—I . . . 190
9.24 Imagination limitations—II . . . 191
9.25 Rube Goldberg mechanism . . . 192
9.26 An experiment demonstrating infant knowledge of substantiality . . . 194
9.27 A second experiment demonstrating infant knowledge of substantiality . . . 195
9.28 An experiment testing infant knowledge of gravity . . . 195
9.29 An experiment demonstrating infant knowledge of continuity . . . 197

Chapter 1

Overview

This thesis addresses two questions related to language. First: *How do children learn the language-specific components of their native language?* Second: *How is language grounded in perception?* These two questions are intimately related. One piece of language-specific information which children must learn is word meanings. Knowledge of the meanings of utterances containing unknown words presumably aids children in the process of determining the meanings of those words. A complete account of such a process must ultimately explain how children extract utterance meanings from their non-linguistic context. Thus the study of child language acquisition has motivated the study of *event perception* as a means of grounding language in perception.

The long-term goal of this research is a comprehensive theory of language acquisition grounded in visual perception. This thesis however, presents more modest short-term accomplishments. Currently, the language acquisition and perception components are the subjects of independent investigation. Part I of this thesis discusses language acquisition while part II discusses event perception. These two parts, however, fit into a common language processing architecture, which this thesis takes to be reflective of the actual human language faculty. Figure 1.1 depicts this architecture. In its entirety, the architecture constitutes a relation between linguistic *utterances*, the non-linguistic *observations* to which those utterances refer, and a *language model* which mediates that mapping. The architecture itself is presumed to be innate and universal. Any language-specific information is encoded in the language model. Language acquisition can be seen as the task of learning that language model from utterances paired with observations derived from the non-linguistic context of those utterances. The language model to be acquired is the one which successfully maps those utterances heard by a child to the observed events.

The language processing architecture divides into three processing modules which relate seven representations. The language model contains two parts, a *grammar* encoding language-specific syntactic knowledge, and a *lexicon*. The lexicon in turn contains two parts, one mapping words to their *syntactic categories* and the other mapping words to their *meanings*. A *parser* relates utterances to their *syntactic structure*. While the parser itself encodes universal syntactic knowledge, presumed to be innate, the mapping between utterances and their syntactic structure is also governed by the language-specific grammar and the syntactic categories of words. A *linker* relates the meaning of an entire utterance, represented as a *semantic structure*, to the meanings of the words comprising that utterance, taken from the lexicon. This mapping is presumably mediated by the syntactic structure of the utterance. Finally, a *perception* module relates semantic structures denoting the meanings of utterances to the non-linguistic observations referred to by those utterances.

This architecture can be though of as an undirected declarative relation. By specifying the direction of information flow, the architecture can by applied to different tasks. Taking an utterance and language model as input and producing predicted observations as output constitutes using the architecture as

Figure 1.1: A generic language processing architecture. It contains three processing modules: a parser, a linker, and a perceptual component, that mutually constrain five representations: the input utterance, the syntax of that utterance, the meaning of that utterance, the visual perception of events in the world, and a language model comprising a grammar and a lexicon. The lexicon in turn maps words to their syntactic category and meaning. Given input comprising utterances paired with observations of their use, this architecture can produce as output, a language model which allows the utterances to explain those observations. The bulk of this thesis is an elaboration on this process.

a language comprehension device. Taking an observation and language model as input and producing as output utterances that describe that observation constitutes using the architecture as a language generation device. Taking an utterance paired with an observation as input and producing as output a language model which allows the utterance to have an interpretation consistent with the observation constitutes using the architecture as a language acquisition device. The first two applications of this architecture are conventional and well-known. The third application, language acquisition, is the novel application considered by this thesis.

Part I of this thesis addresses the two leftmost modules of the architecture from figure 1.1, namely the parser and linker. It presents a theory, implemented in three different computer programs, for deriving a language model from utterances paired with semantic structures denoting their meaning. Part II of this thesis address the third module from figure 1.1, namely perception. It presents a theory, again implemented as a computer program, for deriving semantic structures which describe the events observed in visual input. As stated before, the long-term goal of this research is to tie these two components together. Currently however, the two halves of this thesis are formulated using incompatible representations of semantic structure. This is due primarily to the preliminary nature of this work. The work on language acquisition predates the work on event perception and was formulated around a semantic representation which later proved inadequate for grounding language in perception. While in its details, the techniques presented in part I of this thesis depend on the old representation, preventing the joint operation of the two programs, at a more general level the techniques transcend the particular representations used. This, combined with the fact that the semantic representation used in part I of this thesis is still widely accepted in the linguistic community, precludes obsolescence of the material presented in part I.

As part of learning their native language, children must learn at least three types of information:

word-to-category mappings, word-to-meaning mappings, and language-specific syntactic information. Collectively, this information is taken to constitute a language model. Part I of this thesis discusses techniques for learning a language model given utterances paired with semantic structures denoting their meaning. The language model can be seen a set of propositions, each denoting some linguistic fact particular to the language being learned. For example, the language model for English might contain the propositions '*table* is a noun', '*table* means **table**', and 'prepositions precede their complements'.[1]

Acquisition of the language model might proceed in stages. The process of learning new propositions might be aided by propositions already acquired in previous stages. To avoid infinite regress, however, the process must ultimately start with an empty language model containing no language-specific information. The task of learning a language model with no prior language-specific information has become known a language *bootstrapping*. The models explored in part I of this thesis address language bootstrapping.

The language bootstrapping task is illustrated by the following small example. Let us assume that the learner hears the utterance *John walked to school*. In addition, let us assume that the learner can discern the meaning of that utterance from its non-linguistic context. Furthermore, let us take WALK(**John**, TO(**school**)) to be the representation of that meaning. The learner would attempt to form an analysis of this input which was consistent with her model of universal grammar. For instance, the learner might postulate the following analysis.

If the learner could determine that this analysis was correct, she could add a number of propositions to her language model, including '*John* is a noun', '*John* means **John**', and 'prepositions precede their complements'. Unfortunately, the following analysis might also be consistent with the learner's model of universal grammar.

[1] Throughout this thesis, words in *italics* denote linguistic tokens while words in **boldface** or UPPER CASE denote semantic representations of word meanings. Furthermore, there is no prior correspondence between a linguistic token such as *table* and a semantic token such as **table**, even though they share the same spelling. They are treated as uninterpreted tokens. The task faced by the learner is to acquire the appropriate correspondences as word-to-meaning mappings.

If the learner adopted this analysis, she would incorrectly augment her language model with the propositions '*John* is a verb', '*John* means WALK(x, y)', and 'prepositions follow their complements'. During later stages of language acquisition, the partial language model might aid the learner in filtering out incorrect analyses. Such assistance in not available during language bootstrapping however.

Many competing theories of language acquisition (cf. Pinker 1984 and Lightfoot 1991) address this problem by suggesting that the learner employs a conservative *trigger*-based strategy whereby she augments her language model with only those propositions that are uniquely determined given her current language model and the current input utterance taken in isolation. In the above situation, trigger-based strategies would not make any inferences about the language being learned since such inferences could not uniquely determine any language-specific facts. Trigger-based strategies have difficulty explaining language bootstrapping due to the rarity of situations where an input utterance has a single analysis given a sparse language model.

This thesis adopts an alternative *cross-situational* learning strategy to account for language bootstrapping. Under this strategy, the learner attempts to find a language model which is consistent across multiple utterances. Each utterance taken in isolation might admit multiple analyses while the collection of several utterances might allow only a single consistent analysis. This allows the learner to acquire partial knowledge from ambiguous situations and combine such partial knowledge across situations to infer a unique language model despite the ambiguity in the individual isolated situations. For example, the learner could rule out the second analysis given above upon hearing the utterance *Mary walked to school* paired with WALK(**Mary**, TO(**school**)) since this utterance does not admit an analysis which takes *school* to mean **John**. This cross-situational approach thus also alleviates the need to assume prior knowledge, since all such knowledge can be acquired simultaneously by the same mechanism. A naive implementation of cross-situational learning would require the learner to remember prior utterances to make a collection of utterances available to cross-situational analysis. Such an approach would not be cognitively plausible. Part I of this thesis explores a number of techniques for performing cross-situational learning without keeping track of prior utterances.

Let me elaborate a bit on my use of the term cross-situational. While learning language, children are exposed to a continual stream of situations where they hear utterances in their non-linguistic context. Intuitively, the term cross-situational describes a strategy whereby the learner acquires language by analyzing multiple situations. Clearly, a child cannot learn her entire native language from a single pair of linguistic and non-linguistic observations. Thus in a trivial sense, all learning strategies are cross-situational. This thesis however, uses the term to describe a very particular strategy, one whereby the

learner finds a single language model which can consistently account for all of the observed situations. A language model must meet two criteria to account for an observed situation. First, it must allow the utterances heard in that situation to be syntactically well-formed. Second, it must allow those utterances to be semantically true and relevant to their non-linguistic context. Thus using this strategy, the learner applies all possible syntactic and semantic constraints across all of the observed situations to the language acquisition task. This strategy is described in greater detail in chapter 3 where it is called *strong* cross-situational learning. This strategy dates back at least to Chomsky (1965). This thesis renders more precision to this strategy and tests it on several concrete linguistic theories.

It is instructive to contrast this strategy with a number of alternatives. Gold (1967) describes a strategy whereby the learner enumerates the possible language models $\{L_1, L_2, \ldots\}$, first adopting the language model L_1 and subsequently switching to the next language model in the sequence when the current language model cannot account for the current observation. Hamburger and Wexler (1975) describe a variant of this strategy where learner does not try the alternative language models in any particular enumerated order but rather switches to a new language model at random when the current language model fails to account for the observation. The new language model is restricted to be related to the previous language model by a small number of change operators. These strategies are weaker than strong cross-situational learning since when the learner switches to a new language model that is consistent with the current observation, she does not check that it is also consistent with all prior observations.

The strategy adopted by Gold does not impart any structure on the language model. It is often natural, however, to view the language model as comprising attribute-value pairs. Such pairs may represent word-to-category mappings, word-to-meaning mappings, or values of syntactic parameters. Another common learning strategy is to form the set of alternate values for each attribute that are consistent with each utterance as it is processed and intersect those sets. The value of an attribute is determined when a singleton set remains. Pinker (1987a) adopts this strategy to describe the acquisition of word-to-meaning mappings. More generally it can be used to learn any information represented as attribute-value pairs, including word-to-category mappings and syntactic parameter settings. Chapter 3 refers to this strategy as *weak* cross-situational learning and demonstrates that it is weaker than strong cross-situational learning. This reduction in power can be explained simply as follows. Consider a language model with two attributes a_1 and a_2 each having two possible values v_1 and v_2. Nominally, this would allow four distinct language models. It may be the case that setting a_1 to v_1 is mutually inconsistent with setting a_2 to v_2, even though all three remaining possible language models are consistent. It is impossible to represent such information using only sets of possible attribute values since in this case, there exists some language model consistent with each attribute-value pair in isolation. Thus weak cross-situational learning may fail to rule out some inconsistent language models which would be ruled out by strong cross-situational learning.

Strong cross-situational learning is a powerful but computationally expensive technique. Some of the implementations discussed in chapter 4 do use full strong cross-situational learning. For reasons of computational efficiency, however, some of the implementations, use weaker strategies. These weaker strategies differ from both weak cross-situational learning and the enumeration strategies described above. They will be described in detail in chapter 4.

The actual language learning task faced by children is somewhat more complex than the task portrayed by the example described earlier. That example assumed that the learner could determine the correct meaning of an utterance from context and simply needed to associate parts of that meaning with the appropriate words in the utterance. It is likely however, that children face *referential uncertainty* during language learning, situations where the meaning of an utterance is uncertain. They might be able to postulate several possible meanings consistent with the non-linguistic context of an utterance but might not be sure which of these possible meanings is the correct meaning of the utterance. Unlike trigger-based strategies, cross-situational learning techniques can learn in the presence of referential

uncertainty.

Part I of this thesis applies a cross-situational learning strategy to the task of learning a language model comprising word-to-category mappings, word-to-meaning mappings, and language-specific components of grammar, without access to prior language-specific knowledge, given utterance-meaning pairs which exhibit referential uncertainty. This strategy has been implemented in a series of computer programs which test this strategy on linguistic theories of successively greater sophistication. In accord with current hypotheses about child language acquisition, these systems use only positive examples to drive their acquisition of a language model. The operation of DAVRA is typical of these programs. Figure 1.2 illustrates a sample corpus presented as input to DAVRA. Note that this corpus exhibits referential uncertainty in that each utterance is paired with several possible meanings for that utterance. Given this corpus, DAVRA can derive the language model illustrated in figure 1.3. DAVRA learns that English is head-initial and SPEC-initial. Furthermore, DAVRA learns unique word-to-category and word-to-meaning mappings for most of the words in the corpus.

Part I of this thesis discusses three language acquisition programs which incorporate cross-situational learning techniques. MAIMRA, the first program developed, learns word-to-meaning and word-to-category mappings from a corpus pairing utterances with sets of expressions representing the potential meanings of those utterances hypothesized by the learner from the non-linguistic context. MAIMRA's syntactic theory is embodied in a fixed context-free grammar. DAVRA, the second program developed, extends MAIMRA by replacing the context-free grammar with a parameterized variant of $\overline{\mathrm{X}}$ theory. Given the same corpus as MAIMRA, DAVRA learns the parameter settings for $\overline{\mathrm{X}}$ theory in addition to a lexicon mapping words to their syntactic category and meaning. DAVRA has been successfully applied, without change, to tiny corpora in both English and Japanese, learning the requisite lexica and parameter settings despite differences in word order between the two languages. KENUNIA, the third program developed, incorporates a more comprehensive model of universal grammar supporting movement, adjunction, and empty categories, as well as more extensive parameterization of its $\overline{\mathrm{X}}$ theory component. This model of universal grammar is based on recent linguistic theory and includes such notions as the DP hypothesis, VP-internal subjects, and V-to-I movement. KENUNIA is able to learn the parameter settings of this model, as well as word-to-category mappings, in the presence of movement and empty categories. All of these programs strive to model language bootstrapping, with little or no access to prior language-specific knowledge, in the presence of referential uncertainty. Chapter 4 will present, in detail, the algorithms underlying MAIMRA, DAVRA, and KENUNIA along with annotated examples depicting their operation on sample learning tasks.

Part II of this thesis addresses the task of grounding semantic representations in visual perception. In doing so it asks three questions, offering novel answers to each. The first question is: *What is an appropriate semantic representation that can allow language to be grounded in perception?* Chapter 7 advances the claim that an appropriate semantic representation for the meanings of simple spatial motion verbs such as *throw*, *pick up*, *put*, and *walk* must incorporate the notions of *support*, *contact*, and *attachment* as these notions play a central role in differentiating occurrences of events described by those words from non-occurrences. Prior representations of verb meaning focussed on the aspects of motion depicted by the verb. For example, Miller (1972), Schank (1973), Jackendoff (1983), and Pinker (1989) all gloss *throw* roughly as 'to cause an object to move'. This misses two crucial components of throwing—the requirement that the motion be caused by moving one's hand while grasping the object (contact and attachment) and the requirement that the resulting motion be unsupported. Chapter 7 presents a novel lexical semantic representation based on the notions of support, contact, and attachment, and uses that representation to characterized the prototypical events described by numerous spatial motion verbs.

Given that support, contact, and attachment relations play a central role in defining verb meanings, a natural second question arises: *How are support, contact, and attachment relations between objects perceived?* Chapter 8 offers an answer to that question: *counterfactual simulation*—imagining the short-term future of a potentially modified image under the effects of gravity and other physical forces. For

$\mathrm{BE}(\mathbf{person}_1, \mathrm{AT}(\mathbf{person}_3)) \vee \mathrm{BE}(\mathbf{person}_1, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_1, \mathrm{FROM}(\mathbf{person}_3)) \vee$ $\mathrm{GO}(\mathbf{person}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_3), \mathrm{TO}(\mathbf{person}_2)])$ *John rolled.*
$\mathrm{BE}(\mathbf{person}_2, \mathrm{AT}(\mathbf{person}_3)) \vee \mathrm{BE}(\mathbf{person}_2, \mathrm{AT}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_2, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_2, \mathrm{FROM}(\mathbf{person}_3)) \vee$ $\mathrm{GO}(\mathbf{person}_2, \mathrm{TO}(\mathbf{person}_1)) \vee \mathrm{GO}(\mathbf{person}_2, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_3), \mathrm{TO}(\mathbf{person}_1)])$ *Mary rolled.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *Bill rolled.*
$\mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{object}_1, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{object}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *The cup rolled.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *Bill ran to Mary.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *Bill ran from John.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{object}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{object}_1)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{object}_1)])$ *Bill ran to the cup.*
$\mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{object}_1, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{object}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *The cup slid from John to Mary.*
$\mathrm{ORIENT}(\mathbf{person}_1, \mathrm{TO}(\mathbf{person}_2)) \vee$ $\mathrm{ORIENT}(\mathbf{person}_2, \mathrm{TO}(\mathbf{person}_3)) \vee$ $\mathrm{ORIENT}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_1))$ *John faced Mary.*

Figure 1.2: A sample corpus presented to DAVRA. The corpus exhibits referential uncertainty in that each utterance is paired with several possible meaning expressions. DAVRA is not told which is the correct meaning, only that one of the meanings is correct.

Head Initial, SPEC Initial.		
John:	[N]	$\mathbf{person}_1$
Mary:	[N]	$\mathbf{person}_2$
Bill:	[N]	$\mathbf{person}_3$
cup:	[N]	$\mathbf{object}_1$
the:	[$\mathrm{N}_{\mathrm{SPEC}}$]	$\perp$
rolled:	[V]	$\mathrm{GO}(x, [_{\mathrm{Path}}\ ])$
ran:	[V]	$\mathrm{GO}(x, y)$
slid:	[V]	$\mathrm{GO}(x, [_{\mathrm{Path}}\ y, z])$
faced:	[V]	$\mathrm{ORIENT}(x, \mathrm{TO}(y))$
from:	[N,V,P]	$\mathrm{FROM}(x)$
to:	[N,V,P]	$\mathrm{TO}(x)$

Figure 1.3: The language model inferred by DAVRA for the corpus from figure 1.2. Note that DAVRA has converged to a unique word-to-meaning mapping for each word in the corpus, as well as a unique word-to-category mapping for all but two words.

instance, one determines that an object is unsupported if one imagines it falling. Likewise, one determines that an object A supports an object B if B is supported but falls when one imagines a world without A. An object A is attached to another object B if one must hypothesize such an attachment to explain the fact that one object supports the other. Likewise, two objects must be in contact if one supports the other.

Counterfactual simulation relies on a modular *imagination capacity*. This capacity takes the representation of a possibly modified image as input and predicts the short-term consequences of such modifications, determining whether some predicate P holds in any of the series of images depicting the short-term future. The imagination capacity is modular in the sense that the same unaltered mechanism is used for a variety of purposes, varying only the predicate P and the initial image model between calls. This leads to the third question: *How does the imagination capacity operate?* Nominally, the imagination capacity can be though of as a kinematic simulator. To predict the future, this simulator would embody physical knowledge of how objects behave under the influence of physical forces such as gravity. Traditional approaches to kinematic simulation take physical accuracy and the ability to simulate mechanisms of arbitrary complexity to be primary. They typically operate by integrating the aggregate forces on objects, relegating collision detection to a process of secondary importance.

Human perception appears to be based on different principles however. These include the following.

substantiality: Solid objects don't pass through one another.

continuity: Objects follow continuous paths when moving from one location to another. They don't disappear and reappear elsewhere later.

gravity: Unsupported objects fall.

ground plane: The ground acts as universal support for all objects.

These principles are pervasive. It is hard to imagine situations that violate these principles. Traditional kinematic simulation, however, violates some of these principles as a matter of course. Numerical integration violates continuity. Performing collision detection exogenous to numerical integration will admit substantiality violations up to the tolerance allowed by the integration step size. Thus traditional

approaches to kinematic simulation do not appear to be appropriate foundations for a model of the human imagination capacity.

Chapter 9 advances the claim that the imagination capacity used for counterfactual simulation and event perception is organized along very different lines than traditional kinematic simulators. It directly encodes the principles of substantiality, continuity, gravity, and ground plane. It takes collision detection to be primary and physical accuracy to be secondary. In doing so it must forego the ability to simulate mechanisms of arbitrary complexity. The reason for this shift in priorities is that collision detection is more important than physical accuracy in determining support, contact, and attachment relations.

Chapters 8 and 9 review some experiments reported by Freyd et al. (1988), Baillargeon et al. (1985), Baillargeon (1986, 1987) and Spelke (1988) which support the claims made in part II of this thesis. As additional evidence, a simplified version of this theory has been implemented as a working computer program called ABIGAIL. ABIGAIL watches a computer-generated animated movie depicting objects participating in various events. Figure 1.4 illustrates selected frames from a sample movie shown to ABIGAIL. The images in this movie are constructed out of line segments and circles. The input to ABIGAIL consists solely of the positions, orientations, shapes, and sizes of these line segments and circles during each frame of the movie. ABIGAIL is not told which collections of line segments and circles constitute objects. By applying the techniques described above, she must segment the image into objects and determine the support, contact, and attachment relations between these objects as a foundation for producing semantic descriptions of the events in which these objects participate. For example, ABIGAIL can determine that the man is unsupported in frame 11 of the movie by imagining him falling, as depicted in figure 1.5.

The remainder of this thesis is divided into two parts comprising nine chapters. Chapters 2 through 5 constitute part I which discusses language acquisition. Chapter 2 introduces part I by defining the bootstrapping problem and giving an overview of the cross-situational techniques used to address that problem. Chapter 3 illustrates the power cross-situational learning has over trigger-based approaches by demonstrating several small examples, completely worked through by hand, where cross-situational techniques allow the learner to converge on a unique language model for a set of utterances even though each utterance in isolation admits multiple analyses. Chapter 4 presents a detailed discussion of MAIMRA, DAVRA, and KENUNIA—three implemented computer models of language acquisition which incorporate cross-situational techniques. Chapter 5 concludes part I by reviewing related work on language acquisition and suggesting continued work for the future. Chapters 6 through 10 constitute part II which addresses the grounding of language in perception. Chapter 6 introduces part II by describing the event perception task faced by ABIGAIL. Chapter 7 presents a novel lexical semantic representation centered around the notions of support, contact, and attachment, giving definitions in this representation for numerous simple spatial motion verbs. Chapter 8 discusses the event perception mechanisms used by ABIGAIL to segment images into objects and to recover the changing support, contact, and attachment relations between those objects. Chapter 9 discusses ABIGAIL's imagination capacity in detail, showing how the imagination capacity explicitly encodes the naive physical constraints of substantiality, continuity, gravity, and ground plane. Chapter 10 concludes part II by reviewing related work on event perception and suggesting continued work for the future.

Figure 1.4: Several key frames depicting the general sequence of events from the movie used to drive the development of ABIGAIL.

Figure 1.5: The sequence of images produced by ABIGAIL while imagining the short-term future of frame 11 from the movie described in figure 1.4.

Part I

Language Acquisition

Chapter 2

Introduction

We can all agree that as part of the process of acquiring their native language, children must learn at least three things: the syntactic categories of words, their meanings, and the language-specific components of syntax. Such knowledge constitutes, at least in part, the language-specific linguistic knowledge which children must acquire to become fluent speakers of their native language. Initially, children lack any such language-specific knowledge. Yet they come to acquire that knowledge through the language acquisition process. Part I of this thesis attempts to answer the following question: *What procedure might children employ to learn their native language, without any access to previously acquired language-specific knowledge?*

This question is not new nor is this the first attempt at providing an answer. The account offered in this thesis, however, differs from prior accounts in a number of ways. These differences are summarized by three issues highlighted in the question's formulation.

procedure: This thesis seeks a procedural description of the language acquisition process. To be an adequate description, the procedure must be be shown to work. Ideally, one must demonstrate that it is capable of acquiring language given the same input that is available to children. Pinker (1979) calls this the *fidelity criterion*. Such demonstration requires that the procedure be precisely specified. Imprecise procedural specifications, typical of much prior work on language acquisition in cognitive science,[1] admit only speculative evidence that such procedures do actually work and are therefore an inadequate account of the language acquisition process. Ultimately, the most satisfying account would be a procedural specification which is precise enough so that, at least in principle, it could be implemented as a computer program. This thesis presents three different precise procedures, each implemented as a working computer program which successfully solves very small language acquisition tasks. The input to these programs approximates the input available to children.

might: An ultimate account of child language acquisition would demonstrate not only a working language acquisition procedure but also evidence that that procedure was the one actually used by children. This thesis demonstrates only that certain procedures work. It makes no claim that children utilize these procedures. Clearly, it makes sense to suggest that children employ a given procedure only once one knows that the procedure works. Doing otherwise would be putting the cart before the horse. This thesis views the task of proposing working procedures, irrespective of whether children employ these procedures, as the first step toward the ultimate goal of determining the procedures utilized by children.

[1] Notable exceptions to imprecise procedural specifications include the work of Hamburger and Wexler (1975) and Berwick (1979, 1982).

without any prior access to previously acquired language-specific knowledge: To be a complete account, a language acquisition procedure must not rely on previously acquired language-specific knowledge. Doing so only reduces one problem to another unsolved problem. The problem of how children begin the task of language acquisition, without any prior language-specific knowledge, has become known as the *bootstrapping* problem. Most previous accounts assume that children possess some language-specific knowledge, such as the meanings or syntactic categories of nouns, before beginning to acquire the remaining language-specific information. Since these accounts do not present methods for acquiring such preliminary language-specific knowledge, they at worst suffer from problems of infinite regress. At best they describe only part of the language acquisition process. While it may be the case that the language acquisition procedure employed by children is indeed a staged process, to date no one has given a complete account of that entire process. In contrast, the goal of this research program is to propose algorithms which do not rely on any prior language-specific knowledge. Significant progress has been made toward this goal. Chapter 4 presents three implemented language acquisition models. In accord with current hypotheses about child language acquisition, these systems use only positive examples to drive their acquisition of a language model. The first learns both word-to-category and word-to-meaning mappings given prior access only to grammar. The second learns both word-to-category and word-to-meaning mappings, as well as the grammar. The third learns word-to-category mappings along with the grammar, given prior access only to word-to-meaning mappings. All of these models, however, assume prior access to the phonological and morphological knowledge needed to acoustically segment an utterance into words and recognize those words.

Part I of this thesis focuses solely on **language bootstrapping**. The remainder of this chapter describes the bootstrapping problem in greater detail. It makes precise some assumptions this thesis makes about the nature of the input to the language acquisition device, as well as the language-specific knowledge to be learned. Some competing theories about language acquisition share a common learning strategy: they attempt to glean linguistic facts from *isolated observations*. I call this strategy *trigger*-based learning. This thesis advocates an alternative strategy, *cross-situational* learning, and suggests that it may offer a better account of child language acquisition.

2.1 The Bootstrapping Problem

The task of modeling child language acquisition is overwhelmingly complex. Given our current lack of understanding, along with the immensity of the task, any proposed procedure will necessarily address only an idealization of the task actually faced by children. Any idealization will make assumptions about the nature of the input to the language acquisition device. Furthermore, any idealization will address only a portion of the complete language acquisition task, and consider the remainder to be external to that task. Before presenting the language acquisition procedures that I have developed, I will first delineate the idealized problem which they attempt to solve.

I assume that the input to the language acquisition device contains both linguistic and non-linguistic information. It seems clear that the input must contain linguistic information. Assuming that the input contains non-linguistic information deserves some further discussion. Practically everyone will agree that non-linguistic information is required for learning the meaning of words. As Fisher et al. (1991) aptly state: "You can't learn a language simply by listening to the radio". It is not clear however, that non-linguistic information is required for learning syntax. The tacit assumption behind the entire field of formal learning theory (cf. Gold 1967 and Blum and Blum 1975) is that a learner can learn syntax, or at least the ability to make grammaticality judgments, by observing linguistic information alone. It might be the case that this is feasible. Furthermore, both Gleitman (1990) and Fisher et al. (1991) suggest that, at least in part, verb meanings are constrained by their subcategoriza-

tion frames. Brent (1989, 1990, 1991a, 1991b, 1991c) shows how verb subcategorization frames can be derived from an untagged corpus of utterances without any non-linguistic information.[2] Though neither Gleitman, Fisher et al., nor Brent suggest this, it is conceivable that a learner could potentially learn all of syntax, and some semantics, through exposure to linguistic information alone. Whether or not children do so is an open question. Nonetheless, the procedures presented in this thesis utilize both linguistic and non-linguistic information in the process of inferring both syntactic and semantic knowledge, as is in fact typical of most other work in the field.

In the model considered here, the linguistic input to the language acquisition device is a symbolic token stream consisting of a list of grammatical utterances, each utterance being a string of words. Since, the actual linguistic evidence available to children consists of an acoustic signal, this assumes that children have the capacity for segmenting the acoustic stream into utterances and words, as well as classifying different occurrences of a given word as the same symbolic token despite differences in their acoustic waveform. These segmentation and classifications procedures, however, are likely to rely at least in part on language-specific information. An ultimate account of language acquisition would have to explain how children acquire such word segmentation and classification knowledge along with other language-specific knowledge. For pragmatic reasons, the language acquisition procedures proposed in this thesis, like most other proposed procedures, ignore this problem and assume that the learner has the ability to preprocess the acoustic input to provide a symbolic token stream as input to the language acquisition device. Also, like most other proposed procedures, this thesis assumes that the symbolic information recovered from the acoustic input comprises only word and utterance boundary information and word identity. Gleitman (1990) and Fisher et al. (1991) argue that children can also recover information about syntactic structure from the prosodic portion of the acoustic signal and that they utilize such information to aid the language acquisition process. It may be possible to extend the strategies discussed in this thesis to use such prosodic information in a way that would improve their performance. Such exploration remains for future work.

The general learning strategy put forth in this thesis is one of cross-situational learning. This strategy is depicted in figures 2.1 and 2.2. It is incorporated, with minor variation, in all three of the implemented systems discussed in chapter 4. Figure 2.1 illustrates a general language processing architecture. This architecture is a portion of the more complete architecture depicted in figure 1.1. The perception component has been omitted as that will be the focus of part II of this thesis. Part I of this thesis instead focuses on the remaining two processing modules, namely the parser and linker. These two processing modules relate six representations. The parser takes an utterance as input and produces syntactic structures as output. The parsing process uses language-specific syntactic knowledge, in the form of a grammar, along with the syntactic categories of words derived from the lexicon. Taken together, the grammar and lexicon form a language model. The linker implements compositional semantics, combining the meanings of individual words in the utterance, taken from the lexicon, and producing a semantic structure representing the meaning of the entire utterance. This linking process is mediated by the syntactic structure produced by the parser.

Traditionally, the architecture in figure 2.1 is conceived of as being a directed computing device. As a language comprehension device, it receives an utterance, a grammar, and a lexicon as input, and produces (perhaps several ambiguous) semantic structures as output. These semantic structures constitute a representation of the meaning of the input utterance. As a language production device, it receives a communicative goal as input, in the form of a semantic structure, along with a grammar and a lexicon, and produces (perhaps several possible) utterances as output, each of which conveys the semantic content of the desired communicative goal. These two uses of this architecture are conventional and well-known. This thesis explores a novel third possibility. The architecture from figure 2.1 can be viewed instead as a declarative relation that must hold between an utterance u, a semantic structure s, a

[2] His technique requires a small amount of prior language-specific knowledge in the form of a lexicon of closed-class words and a small regular (finite state) covering grammar for English.

Figure 2.1: A generic language processing architecture. The parser takes an input utterance, along with a grammar and syntactic category information from the lexicon, and produces syntactic structures as output. The linker then forms the meaning of the utterance, i.e. its semantic structure, out of the meanings of its constituent words. Word meanings are taken from the lexicon. The linking process is mediated by the syntactic structure produced by the parser for the utterance.

Figure 2.2: This figures illustrates how the generic language processing architecture from figure 2.1 can be used to support cross-situational learning. A copy of the architecture from figure 2.1 is made for each utterance-meaning pair in the corpus. All of these copies are constrained to use the same language model, i.e. the same grammar and lexicon. The learner must find a language model which is consistent across the corpus.

grammar G, and a lexicon L. I will denote this declarative relation via the predicate $U(G, L, u, s)$. Here, U indicates whatever universal linguistic knowledge is presumed to be innate while G and L indicate language-specific grammatical and lexical knowledge that must be acquired. This architecture can be presented with an input utterance u, paired with a semantic structure s representing its meaning. The semantic structure s corresponding to u could be derived by observing the non-linguistic context of the utterance u. The predicate $U(G, L, u, s)$ then constrains the set of possible grammars G and lexica L that are consistent with the assumption that the input utterance u has the given meaning s. Thus U can be used in this fashion as a language acquisition device.

A single utterance paired with a single semantic structure is usually not sufficient to uniquely determine the grammar and lexicon. The grammar and lexicon can, however, be uniquely determined through *cross-situational* learning. The idea behind cross-situational learning is depicted in figure 2.2. Here, the learner is presented with a sequence of utterances, each paired with a representation of its meaning. The architecture from figure 2.1 is replicated, with each utterance-meaning pair being applied to its own copy of the architecture. The different copies however, are constrained to share the same grammar and lexicon. This amounts to the following learning strategy.

Find G and L such that:

$$\begin{array}{c} U(G, L, u_1, s_1) \wedge \\ U(G, L, u_2, s_2) \wedge \\ \vdots \\ U(G, L, u_n, s_n). \end{array}$$

The above learning strategy has a limitation however. It requires that the learner unambiguously know the complete and correct meaning of each input utterance. If the learner was mistaken and associated the wrong meaning with but a single utterance, this architecture either will produce the wrong grammar and lexicon as output, or will not be able to find any grammar and lexicon consistent with the input data. This limitation can be alleviated somewhat by relaxing the input requirement. We could instead allow the learner to hypothesize a *set* of possible meanings for each utterance, most of which will be incorrect. So long as the correct meaning is included with the set of meanings hypothesized for each input utterance, the learner could still determine a grammar and lexicon using the following extended strategy.

Find G and L such that:

$$\begin{array}{c} [U(G, L, u_1, s_{11}) \vee \cdots \vee U(G, L, u_1, s_{1m_1})] \wedge \\ [U(G, L, u_2, s_{21}) \vee \cdots \vee U(G, L, u_2, s_{2m_2})] \wedge \\ \vdots \\ [U(G, L, u_n, s_{n1}) \vee \cdots \vee U(G, L, u_n, s_{nm_n})]. \end{array}$$

Here the learner simply knows that one of the meanings $s_{i1}, \ldots, s_{im_i}$ is the correct meaning for utterance u_i, yet need not know which is actually the correct one. For example, a child hearing the utterance *John threw the ball to Mary* in a situation where John threw the ball to Mary while walking home from school might conjecture that the utterance meant that John and Mary were playing, that Mary wanted the ball, that John and Mary were walking, or a myriad of other possible meanings in addition to the correct one. This type of ambiguity in the mapping of input utterances to their correct meaning will be referred to as *referential uncertainty*. The process of determining G and L will, in retrospect, eliminate the referential uncertainty and allow the learner to determine the correct meanings to associate with each input utterance.

The above strategy still makes some residual assumptions about the input to learner. It requires that each of the input utterances be grammatical in the language to be learned. This is a standard assumption in the field of language acquisition modeling. It also requires that the learner postulate the correct meaning for each utterance as one of the hypothesized meanings for that utterance. The learner would fail to converge to the correct grammar and lexicon if either of these requirements are not met. Furthermore, the strategy becomes intractable if the set of hypothesized meanings paired with each input utterance grows very large. Thus, this strategy is feasible only if the learner possesses some way of narrowing the set of hypothesized meanings using some criteria of salience. Potential solutions to these issues are discussed in section 5.2.

The key claim made in this thesis is that an appropriately constraining theory of universal linguistic knowledge, combined with a large corpus of utterances paired with possible meanings, is sufficient to uniquely determine a language-specific grammar and lexicon, using cross-situational learning. Using cross-situational learning, there is no problem of regress. Unlike other recent proposals (cf. Pinker 1984), this strategy makes no assumption that some language-specific knowledge must be acquired by unspecified means before acquiring other language-specific knowledge.[3]

Let me point out how the above strategy differs from the traditional folklore account of language acquisition. The traditional account claims that children learn a word's meaning by observing situations depicting its use. Presumably, a child hears the word *ball* while being shown a ball and learns to pair the word *ball* with the concept **ball**. For the traditional approach to work, the child must be able to unambiguously pair a word with its concept. This requires that there be at least one situation to which the child is exposed where (a) no other words are uttered along with *ball* while in the presence of balls, and (b) no other objects are present which are potential referents of the word *ball*. Otherwise, a child hearing *Pick up the ball* in the presence of a ball and a truck, could pair *pick* with **ball**, *ball* with **truck**, or even worse, *pick* with **truck**. While undoubtedly, most children are exposed to *some* situations where a single word is uttered in the context of a single salient referent, it seems unlikely that the language acquisition device, robust as it is, could be relying on this strategy given the fleetingly rare possibilities for its use. The cross-situational strategy outlined in this thesis does not make such restrictive assumptions about the nature of the input to the language acquisition device.

2.2 Outline

The remainder of part I of this thesis is divided into three chapters, Chapter 3 motivates the need for cross-situational learning by demonstrating two small examples, fully worked through by hand, which illustrate how cross-situational techniques work and how they can be more powerful than alternate approaches. Before presenting the details of cross-situational learning, chapter 3 first covers some preliminary background material. It discusses a particular semantic linking rule, namely composition by substitution, and how to apply that rule in reverse. Inverse linking, which I call *fracturing*, plays a central role in cross-situational semantic learning. Chapter 4 then presents three implemented systems which apply cross-situational strategies to successively more sophisticated linguistic theories which make fewer and fewer assumptions about the nature of the linguistic input and the child's prior language-specific knowledge. Chapter 5 compares the cross-situational approach to several competing language acquisition theories which do not use cross-situational techniques. It also summarizes the claims made and results reported in part I of this thesis, discussing current limitations and areas for future work.

[3]Except perhaps the language-specific knowledge needed to acoustically segment utterances and recognize words. It may be possible to extend the cross-situational learning techniques presented in this thesis to simultaneously acquire such knowledge as well.

Chapter 3

Cross-Situational Learning

Section 5.1 will review a number of competing approaches to language bootstrapping. Many of the approaches reviewed use *trigger*-based strategies. Trigger-based strategies attempt to learn linguistic facts by observing isolated utterances. There is an alternative to trigger-based learning. Rather than attempting to glean a linguistic fact from a single utterance or utterance-observation pair, it is possible to try to find those linguistic facts that are consistent across multiple utterances and utterance-observation pairs. I will call such techniques *cross-situational* learning. These techniques allows the learner to acquire partial knowledge from ambiguous situations and combine such partial knowledge across situations to infer a unique language model despite the ambiguity in the individual isolated situations.

There are a number of different techniques, some stronger and some weaker, that all fall within the general framework of cross-situational learning. The similarities and differences between these techniques, as well as the power of the general approach, are best illustrated by way of several small examples. This chapter presents two examples of cross-situational learning. They are designed for expository purposes, to characterize in a simple way the techniques used by more complex implementations. Accordingly they utilize simple linguistic theories and make use of some prior language-specific knowledge in the form of a fixed context-free grammar for the language being learned. In chapter 4, I present three implemented systems that incorporate more substantive linguistic theories. Some of these systems require less prior language-specific knowledge then the simple pedagogical examples discussed in this chapter.

Before presenting the examples, I will first discuss *fracturing*, a key technique used in both the examples and the implemented systems to be described. Fracturing is a way of running the linking rules in reverse. Linking rules are normally conceived of as a means for combining the meanings of words into the meanings of utterances comprising those words. During language acquisition, the learner is faced with the opposite task. After pairing utterances with potential meanings derived from the non-linguistic context of those utterances, the learner must pull apart an utterance meaning to map fragments of that meaning to individual words in the utterance. The next section will present a technique for running a particular linking rule in reverse, namely the linking rule proposed by Jackendoff (1983). Sections 3.2 and 3.3 will then present two fully worked-out examples of cross-situational learning in action.

3.1 Linking and Fracturing

Throughout much of part I of this thesis, I will represent meanings as terms, i.e. expressions composed of primitive constant and function symbols. For expository purposes, I will use primitives taken primarily from Jackendoff's (1983) conceptual structure notation, though I will extend this

set arbitrarily as needed.[1] Thus typical meaning expressions will include GO(**cup**, FROM(**John**)) and SEE(**John**, **Mary**). None of the techniques in part I of this thesis attribute any interpretation to the primitives. In every way, the meaning expression GO(**cup**, FROM(**John**)) is treated the same as $f(a, g(b))$.[2]

Variable-free meaning expressions such as those given above will denote the meanings of whole utterances. The meanings of utterance fragments in general, and words in particular, will be represented as meaning expression fragments that may contain variables as place holders for unfilled portions of that fragment. Thus, the word *from* might have the meaning FROM(x) while the word *John* might have the meaning **John**.[3] Crucial to many of the techniques discussed in part I of this thesis is a particular *linking rule* used to combine the meanings of words to form the meanings of phrases and whole utterances. This linking rule is adopted by numerous authors including Jackendoff (1983, 1990), Pinker (1989), and Dorr (1990a, 1990b). Informally, the linking rule forms the meaning of the prepositional phrase *from John* by combining FROM(x) with **John** to form FROM(**John**).

This linking rule can be stated more formally as follows. Each node in a parse tree is assigned an expression to represent its meaning. The meaning of a terminal node is taken from the lexical entry for the word constituting that node. The meaning of a non-terminal node is derived from the meanings of its children. Every non-terminal node u has exactly one distinguished child called its *head*. The remaining children are called the *complements* of the head. The meaning of u is formed by substituting the meanings of each of the complements for all occurrences of some variable in the meaning of the head. To avoid the possibility of variable capture, without adding the complexity of a variable renaming process, we require that the meaning expression fragments associated with complements be variable-free. Notice that this rule does not stipulate which complements substitute for which variables. Thus if GO(x, TO(y)) is the meaning of the head of some phrase, and **John** is the meaning of its complement, the linking rule can produce either GO(x, TO(**John**)) or GO(**John**, TO(y)) as the meaning of the phrase. The only restriction on linking is that the head meaning must contain at least as many distinct variables as there are complements.

Some authors propose variants of the above linking rule that further specifies which variables are linked with which argument positions. For example, Pinker (1989) stipulates that the x in GO(x, y) is always linked to the direct internal argument. Irrespective of whether this is true, either for English specifically, or cross-linguistically in general, I refrain from adopting such restrictions here for two reasons. First, the algorithms presented in part I of this thesis apply generally to any expressions denoting meaning. They transcend a particular representation such as Jackendovian conceptual structures. Linking restrictions such as those adopted by Pinker apply only to expressions constructed out of Jackendovian primitives. Since, for reasons to be discussed in part II of this thesis, the Jackendovian representation is inadequate, it does not make sense to base a learning theory on restrictions which are particular to that representation. Second, the learning algorithms presented here are capable of learning without making such restrictions. In fact, such restrictions could be learned if they were indeed true. The standard motivation for assuming a faculty to be innate is the poverty of stimulus argument. This

[1] In part II of this thesis, I will discuss the inadequacies of both Jackendovian conceptual structure representations as well as substitution-based linking rules. Since much of the work in part I predates the work described in part II, it was formulated using the Jackendovian representation and associate linking rule as a matter of expedience, since that is what was prominent in the literature at the time. In more recent work, such as that described in section 4.3, I abandon the Jackendovian representation in favor of simple thematic role assignments, a much weaker form of semantic representation. This also entails abandoning substitution-based linking in favor of θ-marking. In the future I hope to incorporate the more comprehensive semantic representations discussed in part II of this thesis into the techniques described in part I.

[2] This mitigates to some extent, the inadequacies of the Jackendovian representation. Nothing in part I of this thesis relies on the semantic content of a particular set of primitives. The techniques described here apply equally well to any representation provided that the representation adheres to a substitution-based linking rule. The inadequacies of such a linking rule still limit the applicability of these techniques, however.

[3] Throughout this chapter and the next I will use the somewhat pretentious phrase 'the meaning of x' to mean 'the meaning expression associated with x'.

Figure 3.1: A derivation of the meaning of the utterance *John slid the cup from Mary to Bill* from the meanings of its constituent words using the linking rule proposed by Jackendoff.

argument is falsified most strongly with a demonstration that something is learnable. It is important not to get carried away with our rationalist tendencies making unwarranted innateness assumptions in light of the rare observation of something that is indeed learnable by empiricist methods.

Some words, such as determiners and auxiliaries, appear not to have a meaning that can be easily characterized as meaning expressions to be combined by the above linking rule. To provide an escape hatch for semantic notions that fall outside the system described above, we provide the distinguished meaning symbol $\perp$. Typically, words such as *the* will bear $\perp$ as their meaning. The linking rule is extended so that any complements that have $\perp$ as their meaning are not substituted into the meaning of the head. This allows forming **cup** as the meaning of *the cup* when *the* has $\perp$ and *cup* has **cup** as their respective meanings. Using this linking rule, the meaning of phrases, and ultimately entire utterances can be derived from the meanings of their constituent words, given a parse tree annotated as to which children are heads and which are complements. A sample derivation is shown in figure 3.1. Note that the linking rule is ambiguous and can produce multiple meanings, even in the absence of lexical and structural ambiguity, since it does not specify which variables are linked to which complements. Also note that the aforementioned linking rule addresses only issues of argument structure. No attempt is made to support other aspects of compositional semantics such as quantification.

Substitution-based linking rules are not new. They are widely discussed in the literature (cf. Jackendoff 1983, 1990, Pinker 1989, and Dorr 1990a, 1990b). The techniques in this thesis explore a novel application of such linking rules: the ability to use them in reverse. Traditionally, compositional semantics is viewed as a process for deriving utterance meanings from word meanings. This thesis will explore the opposite possibility: deriving the meanings of individual words from the meanings of utterances containing those words. I will refer to this inverse linking process as *fracturing*.

Fracturing is best described by way of an example. Assume some node in some parse tree has the meaning GO(**cup**, TO(**John**)). Furthermore, assume that the node has two children. In this case there are four possibilities for assigning meanings to the children.

Head	Complement
$\mathrm{GO}(x, \mathrm{TO}(\mathbf{John}))$	**cup**
$\mathrm{GO}(\mathbf{cup}, x)$	$\mathrm{TO}(\mathbf{John})$
$\mathrm{GO}(\mathbf{cup}, \mathrm{TO}(x))$	**John**
$\mathrm{GO}(\mathbf{cup}, \mathrm{TO}(\mathbf{John}))$	$\perp$

Note specifically the last possibility of assigning $\perp$ as the meaning of the complement. This option will always be present when fracturing any node. The above fracturing process can be applied recursively, starting at the root node of a tree, proceeding toward its leaves, to derive possible word meanings from the meaning of a whole utterance. More formally, fracturing a node u is accomplished by the following algorithm.

ALGORITHM To fracture the meaning expression associated with a node u into meaning expression fragments associated with the head of u and its complements:

> Let e be the meaning of u. For each complement, either assign $\perp$ as the meaning of that complement or perform the following two steps.
>
> 1. Select some subexpression s of e and assign it as the meaning of that complement. The subexpression s must not contain any variables introduced in step 2.
> 2. Replace one or more occurrences of s in e with a new variable.
>
> After all complements have been assigned meanings, assign e as the meaning of the head. □

As stated above, the fracturing process is mediated by a parse tree annotated with head-child markings. Given a meaning expression e, one can enumerate all meaning expression fragments which can possibly link together to form e, irrespective of any parse tree for deriving e. Such a meaning fragment is called a *submeaning* of e. For example, the following are all of the submeanings of $\mathrm{GO}(\mathbf{cup}, \mathrm{FROM}(\mathbf{John}))$.

$\mathrm{GO}(\mathbf{cup}, \mathrm{FROM}(\mathbf{John}))$
cup
$\mathrm{GO}(x, \mathrm{FROM}(\mathbf{John}))$
$\mathrm{GO}(x, \mathrm{FROM}(y))$
$\mathrm{GO}(x, y)$
$\mathrm{GO}(\mathbf{cup}, \mathrm{FROM}(x))$
$\mathrm{GO}(\mathbf{cup}, x)$
$\mathrm{FROM}(\mathbf{John})$
$\mathrm{FROM}(x)$
John
$\perp$

If an utterance has e as its meaning, then every word in that utterance must have a submeaning of e as its meaning. The set of submeanings for a meaning expression e can be derived by the following algorithm.

ALGORITHM To enumerate all submeanings of a meaning expression e:

> Let s be some subexpression of e. Repeat the following two steps an arbitrary number of times.
>
> 1. Select some subexpression t of s not containing any variables introduced in step 2.
> 2. Replace one or more occurrences of t in s with a new variable.
>
> Upon completion, s is a possible submeaning of e. Furthermore, $\perp$ is a possible submeaning of every expression. □

Both the fracturing algorithm, as well as the algorithm for enumerating all submeanings of a given

meaning expression, will play a prominent role throughout the remainder of part I of this thesis.

3.2 Learning Syntactic Categories

Consider the following problem. Suppose that a learner was given a fixed context-free grammar along with a corpus of utterances generated by that grammar.[4] Given such information, the learner must derive a lexicon mapping the words in the corpus to their syntactic category. No non-linguistic information is given to the learner.

This problem is typified by the following example. Suppose that the learner is given the following context-free grammar.

$$\begin{array}{rcl} \text{S} & \rightarrow & \text{NP VP} \\ \text{NP} & \rightarrow & \{\text{D}\}\ \text{N} \\ \text{VP} & \rightarrow & \text{V}\ \{\text{NP}\ \{\text{NP}\}\} \end{array}$$

I will refer to this grammar as G_1. Now suppose that the learner hears the utterance *John saw Mary*. Since G_1 generates only two three-word terminal strings, namely N V N and D N V, the learner can conclude that *John* must be either a noun or a determiner, *saw* a verb or a noun, and *Mary* either a noun or a verb, given their respective positions in the input string. If the learner later hears the utterance *Mary ate breakfast*, she can perform a similar analysis and conclude that *Mary* must be a noun since only nouns can appear as both the first and third words of a three word utterance.

This analysis is based on one crucial assumption: that each word bear only one syntactic category. I will call this assumption the *monosemy constraint*. Clearly language contains polysemous words. I will discuss potential ways of relaxing the monosemy constraint in section 5.2.

I will refer to the above technique as *weak* cross-situational learning. In the above example, weak cross-situational learning constrains only the syntactic category of *Mary*, and not any of the remaining words, since only *Mary* appears in multiple utterances. The learner can nonetheless perform more aggressive inference given the above information. Once the learner infers that *Mary* is a noun, she can rule out D N V as a possible analysis for *Mary ate breakfast*, leaving only the N V N analysis. Thus the learner can also infer that *ate* is a verb and *breakfast* is a noun. Furthermore, if the learner was able to reanalyze previous utterances, she could perform a similar analysis on *John saw Mary* and determine that *John* is a noun and *saw* is a verb. The given grammar and corpus permit only one consistent analysis and thus entail a unique lexicon. I will call the process of finding such a consistent analysis, *strong* cross-situational learning. In the above example, weak cross-situational learning could never converge to a unique lexicon since a noun can appear anywhere a determiner can appear. Thus strong cross-situational learning is strictly more powerful than weak cross-situational learning.

As formulated above, cross-situational learning requires the learner to remember prior utterances. This may not be cognitively plausible. An alternative formulation, however, alleviates this drawback. A lexical entry can be viewed as a proposition, for example

$$\text{category}(\textit{John}) = \text{N}.$$

A lexicon is normally thought of as a set of lexical entries. This can be viewed as a conjunction of propositions, for example

$$\text{category}(\textit{John}) = \text{N} \wedge \text{category}(\textit{saw}) = \text{V} \wedge \text{category}(\textit{Mary}) = \text{N}.$$

[4] Clearly children do not have prior access to such language-specific information. This is example is simplified for expository purposes. The DAVRA and KENUNIA systems discussed in chapter 4 do not assume prior access to a language-specific grammar.

The concept of a lexicon formula can be extended to include disjunctions of propositions. Such disjunctive lexicon formulae can represent intermediate states of partial information about the lexicon being learned. Thus after hearing the utterance *John saw Mary*, the learner can form the following disjunctive lexicon formula.

$$(\text{category}(\textit{John}) = \text{N} \wedge \text{category}(\textit{saw}) = \text{V} \wedge \text{category}(\textit{Mary}) = \text{N}) \vee$$
$$(\text{category}(\textit{John}) = \text{D} \wedge \text{category}(\textit{saw}) = \text{N} \wedge \text{category}(\textit{Mary}) = \text{V})$$

The learner can discard the utterance and retain only the derived lexicon formula. Upon hearing each new utterance, the learner can form a new lexicon formula for that utterance and conjoin it with the previous lexicon formula. In this case, the entire lexicon formula would be a conjunction of disjunctions of conjunctions of lexical entry propositions. Further formulae representing the monosemy constraint can be conjoined with the lexicon formula. Such *monosemy formulae* take the following form

$$\overline{\text{category}(\textit{saw}) = \text{N} \wedge \text{category}(\textit{saw}) = \text{V}}$$

which states that no word can bear two different categories. Strong cross-situational learning can then be seen as finding truth assignments to the lexical entry propositions which satisfy the resulting lexicon formula. Though determining propositional satisfiability is NP-complete, well-known heuristics, such as boolean constraint propagation, can usually solve such problems efficiently in practice (cf. McAllester unpublished, 1978, 1980, 1982, and Zabih and McAllester 1988).

The difference between weak and strong cross-situational learning can be seen as generating different forms of lexicon formulae. Given the utterance *John saw Mary*, weak cross-situational learning can be viewed as constructing the following lexicon formula

$$(\text{category}(\textit{John}) = \text{N} \vee \text{category}(\textit{John}) = \text{D}) \wedge$$
$$(\text{category}(\textit{saw}) = \text{V} \vee \text{category}(\textit{saw}) = \text{N}) \wedge$$
$$(\text{category}(\textit{Mary}) = \text{N} \vee \text{category}(\textit{Mary}) = \text{V})$$

instead of the formula described previously. It is easy to see that the lexicon formula created for weak cross-situational learning is linear in the size of the input utterance. The naive approach for generating the lexicon formula corresponding to strong cross-situational learning would generate a disjunct for each possible parse. Since there could be an exponential number of parses, this would appear intractable.

It is possible however, to use a variant of the CKY algorithm (Kasami 1965, Younger 1967) to share common subformulae and generate, in polynomial time, a lexicon formula whose size is polynomial in the length of the input utterance. This is done as follows. Lexical entry propositions of the form l_{wc} are created for each word w and syntactic category c. Next, for each utterance, propositions of the form p_{ijc} are created for each syntactic category c and each $0 \leq i \leq j \leq n$ where n is the length of the utterance. Intuitively, the proposition p_{ijc} is true if the subphrase from position i through position j in the utterance can be parsed as category c. For each binary branching rule A $\rightarrow$ B C in the grammar,[5] and for each $0 \leq i \leq j \leq n$, propositional formulae of the form

$$p_{ij\text{A}} \rightarrow \bigvee_{k=i}^{j} p_{ik\text{B}} \wedge p_{kj\text{C}}$$

are conjoined to form a large formula. To this one conjoins all formulae of the form

$$p_{ii\text{C}} \rightarrow l_{w\text{C}}$$

where C is a category, $0 \leq i \leq n$, and w is the word at position i, as well as asserting the single proposition $p_{0n\text{S}}$ where S is the root category of the grammar. Formulae such as these are created

[5] Any context-free grammar can be converted into a weakly equivalent grammar containing only binary branching rules. This conversion process, known as conversion to Chomsky Normal Form, does not affect the category learning process.

for each utterance in the corpus and conjoined together. Finally, monosemy formulae over the l_{wc} propositions are added to enforce the monosemy constraint. This whole formula can be converted to conjunctive normal form yielding a formula whose size is polynomial in the length of the corpus.[6] Satisfying assignments to this formula constitute word-to-category mappings that are consistent with both the corpus and the grammar.

3.3 Learning Syntactic Categories and Word Meanings Together

The previous example illustrated the use of weak and strong cross-situational techniques for learning syntactic categories from linguistic information alone without any reference to semantics. It is possible to extend these techniques to learn both syntactic and semantic information when given both linguistic and non-linguistic input. As the next example will illustrate, non-linguistic input can help not only in the acquisition of word meanings but can also assist in learning syntactic categories as well. Furthermore, syntactic knowledge can aid the acquisition of word meanings. The example will demonstrate how strong cross-situational learning, applied to a combined syntactic and semantic theory, is more powerful than either weak or strong cross-situational learning applied to either syntax or semantics alone.

Consider a learner who possess the following context-free grammar.

$$\begin{array}{rcl} \text{S} & \rightarrow & \text{NP VP} \\ \text{NP} & \rightarrow & \{\text{D}\}\ \text{N} \\ \text{VP} & \rightarrow & \text{V}\ \{\text{NP}\ \{\text{NP}\}\}\ \text{PP}^* \\ \text{PP} & \rightarrow & \text{P NP} \end{array}$$

I will refer to this grammar as G_2. Now suppose that the learner hears the following five utterances.[7][8]

s_1:	*John fled from the dog.*	FLEE(**John**, FROM(**dog**))
s_2:	*John walked from a corner.*	WALK(**John**, FROM(**corner**))
s_3:	*Mary walked to the corner.*	WALK(**Mary**, TO(**corner**))
s_4:	*Mary ran to a cat.*	RUN(**Mary**, TO(**cat**))
s_5:	*John slid from Bill to Mary.*	SLIDE(**John**, [$_{\text{Path}}$ FROM(**Bill**), TO(**Mary**)])

Each utterance is paired with its correct meaning as derived by the learner from observation of its non-linguistic context.[9] Furthermore, I will assume that the learner knows that each of the input utterances is generated by G_2 and that the meanings associated with each utterance are derived from the meanings of the words in that utterance via the syntax-mediated linking rule described in section 3.1. In this example however, I assume that the learner does *not* know which syntactic categories constitute the heads of the rules in G_2. Thus the learner must consider all possibilities. The task faced by the

[6] The size of the formula constructed for each utterance is cubic in the length of that utterance. Assuming a bound on utterance length, the size of the formula constructed is thus linear in the number of utterances and quadratic in the number of distinct words appearing in the corpus, due to the monosemy formulae.

[7] To reiterate, words in *italics* denote linguistic tokens while words in **boldface** or UPPER CASE denote semantic representations of word meanings. There is no prior correspondence between a linguistic token such as *John* and a semantic token such as **John**, even though they share the same spelling. They are treated as uninterpreted tokens. The task faced by the learner is to acquire the appropriate correspondences as word-to-meaning mappings.

[8] For the purposes of this thesis, the notation [$_{\text{Path}}$ x, y] can be viewed as a two argument function which combines two paths to yield an aggregate path with the combined properties of the path arguments x and y.

[9] To simplify this example I will assume that the learner unambiguously knows the meaning of each utterance in the corpus. Techniques described section 2.1 can be used to relax this assumption and allow referential uncertainty. Such techniques are incorporated in all of the implementations described in chapter 4.

	1	2	3	4	5
(a)	D	N	V	D	N
(b)	D	N	V	N	N
(c)	D	N	V	P	N
(d)	N	V	D	N	N
(e)	N	V	N	D	N
(f)	N	V	N	P	N
(g)	N	V	P	D	N
	{N, D}	{N, V}	{N, V, P, D}	{N, P, D}	{N}

	1	2	3	4	5	6
(a)	D	N	V	N	D	N
(b)	D	N	V	D	N	N
(c)	D	N	V	N	P	N
(d)	D	N	V	P	D	N
(e)	N	V	D	N	D	N
(f)	N	V	D	N	P	N
(g)	N	V	N	P	D	N
(h)	N	V	N	N	P	N
(i)	N	V	P	N	P	N
	{N, D}	{N, V}	{N, V, P, D}	{N, P, D}	{N, P, D}	{N}

Figure 3.2: All possible terminal category strings for five and six word utterances generated by grammar G_2.

learner is to discern a lexicon that maps words both to their syntactic categories, as well their meanings, so that the derived lexicon consistently allows the utterances to be generated by G_2 and their associated meanings to be derived by the linking rule.

Consider first what the learner can glean by applying weak cross-situational techniques to the linguistic information alone. Each of the input utterances is five words long, except for the last utterance which is six words long. There are seven possible terminal strings of length five, and nine of length six. These are illustrated in figure 3.2.

The syntactic category assignments produced by weak cross-situational learning are illustrated in figure 3.3. Note that weak cross-situational learning can uniquely determine only the syntactic categories of *Mary*, *corner*, *cat*, and *dog*. These are uniquely determined because they occur in utterance final positions and G_2 allows only nouns to appear as the last word of utterances of length greater than three. Furthermore, notice that in the above corpus, most of the words appear cross-situationally in the same position of an utterance of the same length. Thus the set intersection techniques of weak cross-situational learning offer little help here in reducing the possible category mappings. In fact, only the words *Mary* and *to* engender the intersection of two distinct category sets. Even here though, one set is a subset of the other. Thus for this example, weak cross-situational learning provides no information.

Strong cross-situational learning can improve upon this somewhat but not significantly. The fact that *Mary* is a noun rules out the first three analyses for both s_3 and s_4 since they require the first word to be a determiner. This implies that both *walked* and *ran* must be verbs since the remaining four analyses all have verbs in second position. Discovering that *walked* is a verb can allow the learner to rule

John:	[N,D]	$\{N,D\} \cap \{N,D\} \cap \{N,D\}$
fled:	[N,V]	$\{N,V\}$
from:	[N,V,P,D]	$\{N,V,P,D\} \cap \{N,V,P,D\} \cap \{N,V,P,D\}$
the:	[N,P,D]	$\{N,P,D\} \cap \{N,P,D\}$
dog:	[N]	$\{N\}$
walked:	[N,V]	$\{N,V\} \cap \{N,V\}$
a:	[N,P,D]	$\{N,P,D\} \cap \{N,P,D\}$
corner:	[N]	$\{N\}$
Mary:	[N]	$\{N,D\} \cap \{N,D\} \cap \{N\}$
to:	[N,P,D]	$\{N,V,P,D\} \cap \{N,V,P,D\} \cap \{N,P,D\}$
ran:	[N,V]	$\{N,V\}$
cat:	[N]	$\{N\}$
slid:	[N,V]	$\{N,V\}$
Bill:	[N,P,D]	$\{N,P,D\}$

Figure 3.3: An illustration of the syntactic category assignments that weak cross-situational learning can infer for the sample corpus using linguistic information alone.

out the first three analyses for s_2 since they require a noun in second position. This allows the learner to infer that *John* must be a noun and *from* cannot be a verb. Since *John* is a noun, s_1 cannot have the first three analyses and s_5 cannot have the first four. Thus *fled* and *slid* must be verbs and *Bill* cannot be a determiner.

At this point the learner knows the syntactic categories of all of the words in the corpus except for *from*, *to*, *the*, *a*, and *Bill*. The words *from*, *to*, *the*, and *a* might still be either nouns, prepositions, or determiners, and *Bill* might be either a noun or a preposition. There are however, additional cross-situational constraints between the possible category assignments of these words. Not all possible combinations are consistent with G_2. One can construct a constraint satisfaction problem (CSP) whose solutions correspond to the allowable combinations. The variables of this CSP are the words *from*, *to*, *the*, and *a*. Each of these variables range over the categories N, D, and P. Define $P(x, y)$ to be the constraint which is true if one of the last four analyses for five word utterances allows category x to appear in third position at the same time that category y can appear in fourth position. Thus $P(x, y)$ is true only for the pairs (D, N), (N, D), (N, P), and (P, D). Furthermore, define $Q(x, y)$ to be the constraint which is true if one of the last five analyses for six word utterances allows category x to appear in third position at the same time that category y can appear in fifth position. Thus $Q(x, y)$ is true only for the pairs (D, D), (D, P), (N, D), (N, P), and (P, P). The allowed category mappings must satisfy the following constraint.

$$P(\mathit{from}, a) \wedge P(\mathit{from}, \mathit{the}) \wedge P(\mathit{to}, a) \wedge P(\mathit{to}, \mathit{the}) \wedge Q(\mathit{from}, \mathit{to})$$

This constraint admits only three solutions. The following table outlines these possible simultaneous category mappings along with the analyses they entail for each of the five utterances in the corpus.

from	*to*	*the*	*a*	s_1	s_2	s_3	s_4	s_5
N	P	D	D	(e)	(e)	(g)	(g)	(h)
P	P	D	D	(g)	(g)	(g)	(g)	(i)
D	D	N	N	(d)	(d)	(d)	(d)	(e)

Thus *the* and *a* cannot be prepositions and *to* cannot be a noun. Furthermore, this analysis has shown that *Bill* must be a noun. In this example, strong cross-situational learning cannot, however, narrow

from can be a N					
John	*fled*	*from*	*the*	*dog.*	
N	V	N	D	N	
John	*walked*	*from*	*a*	*corner.*	
N	V	N	D	N	
Mary	*walked*	*to*	*the*	*corner.*	
N	V	P	D	N	
Mary	*ran*	*to*	*a*	*cat.*	
N	V	P	D	N	
John	*slid*	*from*	*Bill*	*to*	*Mary.*
N	V	N	N	P	N

from can be a D *to* can be a D *the* can be a N *a* can be a N					
John	*fled*	*from*	*the*	*dog.*	
N	V	D	N	N	
John	*walked*	*from*	*a*	*corner.*	
N	V	D	N	N	
Mary	*walked*	*to*	*the*	*corner.*	
N	V	D	N	N	
Mary	*ran*	*to*	*a*	*cat.*	
N	V	D	N	N	
John	*slid*	*from*	*Bill*	*to*	*Mary.*
N	V	D	N	D	N

Figure 3.4: Analyses of the corpus which are consistent with the language model after strong cross-situational techniques have been applied to syntax, but which are nonetheless incorrect.

down the possible syntactic categories for *from*, *to*, *the*, and *a* any further. Figure 3.4 shows consistent analyses where *the* and *a* can be a noun, *to* can be a determiner, and *from* can be either a noun or a determiner.

Cross-situational learning can be applied to semantics much in the same way as syntax. Using the fracturing technique described in section 3.1, it is possible to enumerate all of the submeanings of the meaning expressions associated with each utterance in the corpus. These are illustrated in figure 3.5

Applying weak cross-situational learning techniques, the learner can constrain the possible meanings of *Mary* to the intersection of the sets of submeanings for each of the utterances s_3, s_4, and s_5, since *Mary* appears in each of these three utterances. Thus *Mary* must take on one of the meanings $\perp$, **Mary**, or TO(x) to be consistent with these utterances. A similar analysis can narrow the possible meanings of the words *a*, *the*, *John*, *walked*, *from*, *to*, and *corner* since each of these words appears in more than one utterance. Figure 3.6 gives the restricted sets of possible meanings derived for these seven words. Weak cross-situational learning cannot constrain the meaning of the remaining words since they each appear only in a single utterance in the corpus. Note that for this example, weak cross-situational learning applied to semantics has succeeded in uniquely determining the meaning of only two words,

John fled from the dog.	*John walked from a corner.*	*Mary walked to the corner.*
FLEE(**John**, FROM(**dog**))	WALK(**John**, FROM(**corner**))	WALK(**Mary**, TO(**corner**))
John	**John**	**Mary**
FLEE(x, FROM(**dog**))	WALK(x, FROM(**corner**))	WALK(x, TO(**corner**))
FLEE(x, FROM(y))	WALK(x, FROM(y))	WALK(x, TO(y))
FLEE(x, y)	WALK(x, y)	WALK(x, y)
FLEE(**John**, FROM(x))	WALK(**John**, FROM(x))	WALK(**Mary**, TO(x))
FLEE(**John**, x)	WALK(**John**, x)	WALK(**Mary**, x)
FROM(**dog**)	FROM(**corner**)	TO(**corner**)
FROM(x)	FROM(x)	TO(x)
dog	**corner**	**corner**
$\perp$	$\perp$	$\perp$

Mary ran to a cat.	*John slid from Bill to Mary.*	
RUN(**Mary**, TO(**cat**))	SLIDE(**John**, [$_{\text{Path}}$ FROM(**Bill**), TO(**Mary**)])	[$_{\text{Path}}$ FROM(**Bill**), TO(x)]
Mary	SLIDE(x, [$_{\text{Path}}$ FROM(**Bill**), TO(**Mary**)])	[$_{\text{Path}}$ x, TO(**Mary**)]
RUN(x, TO(**cat**))	SLIDE(**John**, [$_{\text{Path}}$ FROM(x), TO(**Mary**)])	[$_{\text{Path}}$ FROM(**Bill**), x]
RUN(x, TO(y))	SLIDE(**John**, [$_{\text{Path}}$ FROM(**Bill**), TO(x)])	[$_{\text{Path}}$ FROM(x), TO(y)]
RUN(x, y)	SLIDE(**John**, [$_{\text{Path}}$ x, TO(**Mary**)])	[$_{\text{Path}}$ x, TO(y)]
RUN(**Mary**, TO(x))	SLIDE(**John**, [$_{\text{Path}}$ FROM(**Bill**), x])	[$_{\text{Path}}$ FROM(x), y]
RUN(**Mary**, x)	SLIDE(x, [$_{\text{Path}}$ FROM(y), TO(**Mary**)])	[$_{\text{Path}}$ x, y]
TO(**cat**)	SLIDE(x, [$_{\text{Path}}$ FROM(**Bill**), TO(y)])	FROM(**Bill**)
TO(x)	SLIDE(x, [$_{\text{Path}}$ y, TO(**Mary**)])	FROM(x)
cat	SLIDE(x, [$_{\text{Path}}$ FROM(**Bill**), y])	**Bill**
$\perp$	SLIDE(**John**, [$_{\text{Path}}$ FROM(x), TO(y)])	TO(**Mary**)
	SLIDE(**John**, [$_{\text{Path}}$ x, TO(y)])	TO(x)
	SLIDE(**John**, [$_{\text{Path}}$ FROM(x), y])	**Mary**
	SLIDE(**John**, [$_{\text{Path}}$ x, y])	**John**
	SLIDE(x, [$_{\text{Path}}$ FROM(y), TO(z)])	SLIDE(x, y)
	SLIDE(x, [$_{\text{Path}}$ y, TO(z)])	SLIDE(**John**, x)
	SLIDE(x, [$_{\text{Path}}$ FROM(y), z])	
	SLIDE(x, [$_{\text{Path}}$ y, z])	
	[$_{\text{Path}}$ FROM(**Bill**), TO(**Mary**)]	
	[$_{\text{Path}}$ FROM(x), TO(**Mary**)]	

Figure 3.5: An enumeration of all possible submeanings of the meaning expressions associated with each utterance in the sample corpus. The meaning of a word must be one of the submeanings of each meaning expression associated with an utterance containing that word.

a	$=$	$\perp$	$s_2 \cap s_4$
the	$=$	$\perp$	$s_1 \cap s_3$
John	$\in$	$\{\perp, \mathbf{John}, \mathrm{FROM}(x)\}$	$s_1 \cap s_2 \cap s_5$
Mary	$\in$	$\{\perp, \mathbf{Mary}, \mathrm{TO}(x)\}$	$s_3 \cap s_4 \cap s_5$
walked	$\in$	$\{\perp, \mathrm{WALK}(x, y), \mathbf{corner}\}$	$s_2 \cap s_3$
from	$\in$	$\{\perp, \mathbf{John}, \mathrm{FROM}(x)\}$	$s_1 \cap s_2 \cap s_5$
to	$\in$	$\{\perp, \mathbf{Mary}, \mathrm{TO}(x)\}$	$s_3 \cap s_4 \cap s_5$
corner	$\in$	$\{\perp, \mathrm{WALK}(x, y), \mathbf{corner}\}$	$s_2 \cap s_3$

Figure 3.6: Weak cross-situational techniques can form these narrowed sets of possible meanings for the words which appear in more than one utterance in the sample corpus.

namely that *a* and *the* both mean $\perp$.

Neither strong cross-situational learning applied to syntax alone, nor weak cross-situational learning applied to semantics alone, are sufficient to uniquely determine the syntactic categories or meanings of all of the words in this example. It is possible however, to apply strong cross-situational learning techniques to this problem, incorporating *both* syntactic and semantic constraints. This will force a unique determination of the lexicon. To see this, first remember that strong cross-situational syntax learning has determined that s_3 must have either analysis (d) or analysis (g). If s_3 took on analysis (d) then it would have the following structure.

We know that the root node must mean WALK(**Mary**, TO(**corner**)) since that is given by observation. Furthermore, we know that *the* must mean $\perp$. Since the root meaning contains the symbol TO, which cannot be contributed by the possible meanings for *walk* and *corner*, either the word *Mary* or the word *to* must take on TO(x) as its meaning. Analysis (d) will not allow *Mary* to mean TO(x) since the linking rule could not then produce the desired root meaning. Thus *to* must mean TO(x). Furthermore, *Mary* must mean **Mary** since the root meaning contains the symbol **Mary** which no other word can contribute. At this point, since the meanings of both *to* and *the* have been determined, the linking rule then fixes the meaning of the phrase *to the* to be TO(x). The linking rule can also operate in reverse, using the known meanings of both *Mary* and the root utterance to determine that the phrase *walked to the corner* must mean WALK(x, TO(*corner*)). At this point however, the learner can determine that the linking rule has no way of forming the meaning of *walked to the corner* out of the known meaning for *to the* and the potential meanings for *walked* and *corner*. Thus the learner can infer that utterance s_3 cannot have analysis (d), and must therefore have analysis (g).

Analysis (g) has the following structure.

The learner can annotate this structure with the known meaning for *the* as well as the root meaning. As before, either the word *Mary* or the word *to* must mean TO(x) since no other word can contribute the symbol TO to the root meaning. Furthermore, *Mary* cannot mean TO(x) since the linking rule would not then be able to derive the root meaning. Thus *to* must mean TO(x). Likewise, *Mary* must mean **Mary** since at this point no other word can contribute the necessary symbol **Mary** to the root meaning. Inverse linking can then determine that *walked to the corner* must mean WALK(x, TO(**corner**)). Under analysis (g), the only way to derive this meaning, given the possible meanings for its constituent words, is for *walked* to mean WALK(x, y) and *corner* to mean **corner**.

This type of reasoning has allowed the learner to uniquely determine not only the meanings of the words *Mary*, *walked*, *to*, *the*, and *corner*, but also that *to* must be a preposition and *the* must be a determiner. This rules out the third possible solution to the CSP problem presented earlier implying that *a* must be determiner and *from* cannot be a determiner. Furthermore, s_4 must have analysis (g), s_5 cannot have analysis (e), and neither s_1 nor s_2 can have analysis (d).

Since s_4 must have analysis (g), it must have the following structure.

Knowing the meaning of the root node, as well as the meanings of the words *Mary*, *to*, and *a*, allows the learner to uniquely determine that *ran* must mean RUN(x, y) and *cat* must mean **cat** since these are the only meanings with which the linking rule can produce the desired root meaning.

At this point the learner can analyze s_2 in a fashion similar to s_3. By an argument analogous to the one used for s_3, the learner can rule out analysis (d), determining that only analysis (g) is consistent. In doing so, the learner will assign the meanings **John** to *John* and FROM(x) to *from*. Thus *from* must be a preposition, s_1 must have analysis (g), and s_5 must have analysis (i). At this point, FLEE(x, y) is the only possible meaning for *fled* which will allow s_1 to take on the desired root meaning consistent with analysis (g). Finally, by a similar argument, *slid* must mean SLIDE(x, [$_{\text{Path}}$ x, y]) and *Bill* must mean **Bill** since only these meanings can let the linking rule produce the desired meaning of s_5 under

analysis (i). With this, the learner has completely determined a unique lexicon that is consistent with the corpus.

While this example is somewhat contrived, it nonetheless illustrates a situation in which the combination of syntactic and semantic reasoning is strictly stronger than either applied in isolation. It is particularly important to highlight the fact that syntactic reasoning can help constrain semantic choices and vice versa. The above example demonstrated a continual interplay between syntax and semantics. The central claim of part I this thesis is that such interplay is crucial to language learning. It is the key that can unlock the quagmire of the various bootstrapping hypotheses reviewed in section 5.1, showing that it is not necessary to assume prior language-specific knowledge before the onset of the primary phase of language acquisition. The problem of infinite regress is thus avoided. While actual child language acquisition could not proceed according to the overly simplistic linguistic theory utilized in this example, I conjecture that the process actually performed by children does nonetheless incorporate an interplay between syntax and semantics using cross-situational techniques interwoven with whatever turns out to be the correct linguistic theory. The claim that children learn by an interplay of syntactic and semantic knowledge is fairly uncontroversial. The claim that they utilize a cross-situational strategy to do so is, however, a controversial conjecture. The next chapter attempts to explore the consequences of this conjecture for more substantial linguistic theories.

Chapter 4

Three Implementations

To test the ideas discussed in the previous chapter, I have constructed three systems that incorporate these ideas into working implementations. Each of these systems applies cross-situational learning techniques to a combination of both linguistic and non-linguistic input. In accord with current hypotheses about child language acquisition, these systems use only positive examples to drive their acquisition of a language model. These systems differ from one another in the syntactic and semantic theory which they use. MAIMRA,[1] the first system constructed, incorporates a fixed context-free grammar as its syntactic theory, and represents word and utterance meanings using Jackendovian conceptual structures. MAIMRA learns both the syntactic categories and meanings of words, given a corpus of utterances paired with sets of possible meanings. DAVRA,[2] the second system constructed, extends the results obtained with MAIMRA by replacing the fixed context-free grammar with a parameterized version of $\overline{\mathrm{X}}$ theory. This grammar contains two binary-valued parameters which determine whether the language is head-initial or head-final, and SPEC-initial or SPEC-final. Given a corpus much like that given to MAIMRA, DAVRA learns not only a lexicon similar to that learned by MAIMRA, but the syntactic parameter settings as well. DAVRA has been successfully applied to very small corpora in both English and Japanese, learning that English is head-initial while Japanese is head-final. KENUNIA,[3] the third system constructed, incorporates the most substantial linguistic theory of the three systems. This theory closely follows current linguistic theory and is based on the DP hypothesis, base generation of VP-internal subjects, and V-to-I movement. KENUNIA incorporates a version of $\overline{\mathrm{X}}$ theory with sixteen binary-valued parameters that supports both adjunction as well as head-complement structures. More importantly, KENUNIA supports movement and empty categories. Two types of empty categories are supported: traces of movement, and non-overt words and morphemes. KENUNIA incorporates several other linguistic subsystems in addition to $\overline{\mathrm{X}}$ theory. These include θ-theory, the empty category principle (ECP), and the case filter. The current version of KENUNIA has learned both the parameter settings of this theory, as well as the syntactic categories of words, given an initial lexicon pairing words to their θ-grids. Future work will extend KENUNIA to learn these θ-grids from the corpus, along with the syntactic categories and parameters, instead of giving them to KENUNIA as prior input. In the longer term, I also plan to integrate the language learning strategies from MAIMRA, DAVRA, and KENUNIA with the visual perception mechanisms incorporated in ABIGAIL[4] and discussed in part II of this thesis. The remainder of this chapter will discuss MAIMRA, DAVRA, and KENUNIA in greater detail.

[1] MAIMRA, or מימרא, is an Aramaic word which means *word*.

[2] DAVRA, or דברא, is an Aramaic word which does not mean *word*.

[3] KENUNIA, or קנוניא, is an Aramaic word which means *conspiracy*. In KENUNIA, the linguistic principles conspire to enable the learner to acquire language.

[4] ABIGAIL is not an Aramaic word.

$$
\begin{array}{rcl}
\mathrm{S} & \rightarrow & \mathrm{NP}\ \boxed{\mathrm{VP}} \\
\overline{\mathrm{S}} & \rightarrow & \{\mathrm{COMP}\}\ \boxed{\mathrm{S}} \\
\mathrm{NP} & \rightarrow & \{\mathrm{DET}\}\ \boxed{\mathrm{N}}\ \{\overline{\mathrm{S}}|\mathrm{NP}|\mathrm{VP}|\mathrm{PP}\}^* \\
\mathrm{VP} & \rightarrow & \{\mathrm{AUX}\}\ \boxed{\mathrm{V}}\ \{\overline{\mathrm{S}}|\mathrm{NP}|\mathrm{VP}|\mathrm{PP}\}^* \\
\mathrm{PP} & \rightarrow & \boxed{\mathrm{P}}\ \{\overline{\mathrm{S}}|\mathrm{NP}|\mathrm{VP}|\mathrm{PP}\}^* \\
\mathrm{AUX} & \rightarrow & \{\mathrm{DO}|\mathrm{BE}|\{\mathrm{MODAL}|\mathrm{TO}|\{\{\mathrm{MODAL}|\mathrm{TO}\}\ \mathrm{HAVE}\}\ \{\mathrm{BE}\}\}
\end{array}
$$

Figure 4.1: The context-free grammar used by MAIMRA. The categories enclosed in boxes indicate the heads of each phrase type. The distinction between head and complement children is used by the linking rule to form the meaning of a phrase out of the meaning of its constituents.

4.1 Maimra

MAIMRA (Siskind 1990) was constructed as an initial test of the feasibility of applying cross-situational learning techniques to a combination of linguistic and non-linguistic input in an attempt to simultaneously learn both syntactic and semantic information about language. MAIMRA is given a fixed context-free grammar as input; grammar acquisition is not part of the task faced by MAIMRA. Though the grammar is not hardwired into MAIMRA, and could be changed to attempt acquisition experiments with different input grammars, all of the experiments discussed in this chapter utilize the grammar given in figure 4.1. This grammar was derived from a variant of $\overline{\mathrm{X}}$ theory by fixing the head-initial and SPEC-initial parameters, and adding rules for S, $\overline{\mathrm{S}}$, and AUX. Note that this grammar severely overgenerates due to the lack of subcategorization restrictions. The grammar allows nouns, verbs, and prepositions to take an arbitrary number of complements of any type. MAIMRA is nonetheless able to learn despite the ensuing ambiguity.

MAIMRA incorporates a semantic theory based on Jackendovian conceptual structures. Words, phrases, and complete utterances are assigned fragments of conceptual structure as their meaning. The meaning of a phrase is derived from the meanings of its constituents by the linking rule discussed in section 3.1. To reiterate briefly, the linking rule operates as follows. The linking rule is mediated by a parse tree. Lexical entries provide the meanings of terminal nodes. Each non-terminal node has a distinguished child called its *head*. The remaining children are called the *complements* of the head. Unlike the puzzle given in section 3.3, the grammar given to MAIMRA indicates the head child for every phrase type. Figure 4.1 depicts this information by enclosing the head of each phrase with a box. The meaning of a non-terminal is derived from the meaning of its head by substituting the meaning of the complements for the variables in the meaning of the head. Complements whose meaning is the distinguished symbol $\perp$ are ignored and not linked to a variable in the head. MAIMRA restricts all complement meanings to be variable-free so that no variable renaming is required.

In addition to the grammar, MAIMRA is given a corpus of linguistic and non-linguistic input. Figure 4.2 depicts one such corpus given to MAIMRA. This corpus consists of a sequence of nine multi-word utterances, ranging in length from two to seven words. Each utterance is paired with a set of between three and six possible meanings.[5] MAIMRA is not told which of the meanings is the correct one for each

[5] As described in Siskind (1990), MAIMRA is not given this set of meanings directly but instead derives this set from more primitive information using perceptual rules. These rules state, for instance, that seeing an object at one location followed by seeing it later at a different location implies that the object moved from the first location to the second. The corpus actually given to MAIMRA pairs utterances with sequences of states rather than potential utterance meanings. Thus MAIMRA would derive $\mathrm{GO}(x, [_{\mathrm{Path}}\ \mathrm{FROM}(y), \mathrm{TO}(z)])$ as a potential meaning for an utterance if the state sequence paired

utterance, only that the set contains the correct meaning as one of its members. Thus the corpus given to MAIMRA can exhibit referential uncertainty in mapping the linguistic to the non-linguistic input.

MAIMRA processes the corpus, utterance by utterance, producing a disjunctive lexicon formula for each utterance meaning-set pair. No information other than this lexicon formula is retained after processing an utterance. This processing occurs in two phases, corresponding to the parser and linker from the architecture given in figure 2.1. In the first phase, MAIMRA constructs a disjunctive parse tree representing the set of all possible ways of parsing the input utterance according to the given context-free grammar. Appendix A illustrates sample disjunctive parse trees which are produced by MAIMRA when processing the corpus from figure 4.2. Structural ambiguity can result both from the fact that the grammar is ambiguous, as well as the fact that MAIMRA does not yet have unique mappings from words to their syntactic categories. Initially, MAIMRA assumes that each word can assume any terminal category. This introduces substantial lexical ambiguity and results in corresponding structural ambiguity. As MAIMRA further constrains the lexicon, she can rule out some word-to-category mappings and thus reduce the lexical ambiguity when processing subsequent utterances. Thus parse trees tend to have less ambiguity as MAIMRA processes more utterances. This is evident in the parse trees depicted on pages 210 and 213 which are also illustrated below. When MAIMRA first parses the utterance *Bill ran to Mary*, the syntactic category of *ran* is not yet fully determined. Thus MAIMRA produces the following disjunctive parse tree for this utterance.

```
(OR (S (OR (NP (N BILL) (NP (N RAN)))
           (NP (N BILL) (VP (V RAN)))
           (NP (N BILL) (PP (P RAN))))
       (VP (V TO) (NP (N MARY))))
    (S (NP (N BILL))
       (OR (VP (V RAN) (PP (P TO)) (NP (N MARY)))
           (VP (V RAN) (VP (V TO)) (NP (N MARY)))
           (VP (V RAN) (NP (N TO)) (NP (N MARY)))
           (VP (OR (AUX (DO RAN))
                   (AUX (BE RAN))
                   (AUX (MODAL RAN))
                   (AUX (TO RAN))
                   (AUX (HAVE RAN)))
               (V TO)
               (NP (N MARY)))
           (VP (V RAN)
               (OR (NP (DET TO) (N MARY))
                   (NP (N TO) (NP (N MARY)))))
           (VP (V RAN) (VP (V TO) (NP (N MARY))))
           (VP (V RAN) (PP (P TO) (NP (N MARY)))))))
```

As a result of processing that utterance, in conjunction with the constraint provided by prior utterances, MAIMRA can determine that *ran* must be a verb. Thus when parsing the subsequent utterance *Bill ran from Mary*, which nominally has the same structure, MAIMRA can nonetheless produce the following smaller disjunctive parse tree by taking into account partial information acquired so far.

with that utterance contained a state in which $\mathrm{BE}(x, \mathrm{AT}(y))$ was true, followed later by a state where $\mathrm{BE}(x, \mathrm{AT}(z))$ was true. This primitive theory of event perception is grossly inadequate and largely irrelevant to the remainder of the learning strategy. For the purposes of this chapter, MAIMRA's perceptual rules can be ignored and the input to MAIMRA viewed as comprising a set of potential meanings associated with each utterance. The ultimate goal is to base future language acquisition models on the theory of event perception put forth in part II of this thesis, instead of the simplistic rules used by MAIMRA.

$\mathrm{BE}(\mathbf{person}_1, \mathrm{AT}(\mathbf{person}_3)) \vee \mathrm{BE}(\mathbf{person}_1, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_1, \mathrm{FROM}(\mathbf{person}_3)) \vee$ $\mathrm{GO}(\mathbf{person}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_3), \mathrm{TO}(\mathbf{person}_2)])$ *John rolled.*
$\mathrm{BE}(\mathbf{person}_2, \mathrm{AT}(\mathbf{person}_3)) \vee \mathrm{BE}(\mathbf{person}_2, \mathrm{AT}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_2, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_2, \mathrm{FROM}(\mathbf{person}_3)) \vee$ $\mathrm{GO}(\mathbf{person}_2, \mathrm{TO}(\mathbf{person}_1)) \vee \mathrm{GO}(\mathbf{person}_2, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_3), \mathrm{TO}(\mathbf{person}_1)])$ *Mary rolled.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *Bill rolled.*
$\mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{object}_1, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{object}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *The cup rolled.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *Bill ran to Mary.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *Bill ran from John.*
$\mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{person}_3, \mathrm{AT}(\mathbf{object}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{person}_3, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{person}_3, \mathrm{TO}(\mathbf{object}_1)) \vee \mathrm{GO}(\mathbf{person}_3, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{object}_1)])$ *Bill ran to the cup.*
$\mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_1)) \vee \mathrm{BE}(\mathbf{object}_1, \mathrm{AT}(\mathbf{person}_2)) \vee$ $\mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ ]) \vee \mathrm{GO}(\mathbf{object}_1, \mathrm{FROM}(\mathbf{person}_1)) \vee$ $\mathrm{GO}(\mathbf{object}_1, \mathrm{TO}(\mathbf{person}_2)) \vee \mathrm{GO}(\mathbf{object}_1, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{person}_1), \mathrm{TO}(\mathbf{person}_2)])$ *The cup slid from John to Mary.*
$\mathrm{ORIENT}(\mathbf{person}_1, \mathrm{TO}(\mathbf{person}_2)) \vee$ $\mathrm{ORIENT}(\mathbf{person}_2, \mathrm{TO}(\mathbf{person}_3)) \vee$ $\mathrm{ORIENT}(\mathbf{person}_3, \mathrm{TO}(\mathbf{person}_1))$ *John faced Mary.*

Figure 4.2: A sample corpus presented to both MAIMRA and DAVRA. The corpus exhibits referential uncertainty in that each utterance is paired with several possible meanings. Neither MAIMRA nor DAVRA are told which is the correct meaning, only that one of the meanings is correct.

```
(OR (S (NP (N BILL) (VP (V RAN))) (VP (V FROM) (NP (N JOHN))))
    (S (NP (N BILL))
       (OR (VP (V RAN) (PP (P FROM)) (NP (N JOHN)))
           (VP (V RAN) (VP (V FROM)) (NP (N JOHN)))
           (VP (V RAN) (NP (N FROM)) (NP (N JOHN)))
           (VP (OR (AUX (DO RAN))
                   (AUX (BE RAN))
                   (AUX (MODAL RAN))
                   (AUX (TO RAN))
                   (AUX (HAVE RAN)))
               (V FROM)
               (NP (N JOHN)))
           (VP (V RAN)
               (OR (NP (DET FROM) (N JOHN))
                   (NP (N FROM) (NP (N JOHN)))))
           (VP (V RAN) (VP (V FROM) (NP (N JOHN))))
           (VP (V RAN) (PP (P FROM) (NP (N JOHN)))))))
```

MAIMRA uses a derivative of the CKY parsing algorithm (Kasami 1965, Younger 1967) to produce the disjunctive parse tree. Thus the size of disjunctive parse tree will always be polynomial in the length of the input. The resulting tree may appear larger when printed since a given entry from the well-formed substring table may be a constituent of several other entries and thus may be printed multiple times. Nonetheless, the internal representation of the parse tree is factored to retain its polynomial size. This factored representation stores only a single copy of each subtree in the disjunctive parse tree, even though that subtree may be referenced multiple times. Furthermore, the fracturing process, to be described shortly, preserves the factored representation so that the resulting disjunctive lexicon formulae are kept to a manageable size.

After constructing the disjunctive parse tree for an input utterance, MAIMRA applies the linking rule in reverse to produce a disjunctive lexicon formula. This second phase is a variant of the fracturing procedure described in section 3.1. Recall that the fracturing procedure recursively applies to two arguments: a parse tree fragment and a meaning expression fragment. For the base case, when the parse tree fragment consists of a terminal node, a lexical entry proposition is formed, pairing the word associated with that node with the syntactic category labeling that node and the input meaning expression fragment. For example, fracturing the parse tree fragment `(p to)` with the meaning expression fragment `(from ?0)` would produce the lexical entry proposition `(definition to p (from ?0))`. For the inductive case, MAIMRA forms all possible ways of assigning subexpressions of the meaning expression fragment as the meaning of each complement constituent of the parse tree fragment. MAIMRA then replaces those subexpressions in the original meaning expression fragment with variables, and assigns the resulting meaning expression fragment to the head constituent of the parse tree fragment. Each constituent of the parse tree fragment is then recursively fractured with its associated meaning expression fragment to yield a disjunctive lexicon formula. For each possible subexpression assignment, MAIMRA forms a conjunction of the lexicon formulae returned for each constituent. MAIMRA then forms a disjunction of these conjunctions. Thus the recursive fracturing process produces a formula with alternating layers of disjunction and conjunction.

This process of constructing a disjunctive lexicon formula is best illustrated by way of an example. Consider fracturing the following parse tree:

along with the meaning expression ORIENT(**John**, TOWARD(**Mary**)). This meaning expression has four subexpressions, namely $\perp$, **John**, **Mary**, and TOWARD(**Mary**). Each of these can be assigned as a potential meaning for *John*. Thus, the following reduction illustrates the first step in producing a disjunctive lexicon formula.[6]

$$\texttt{fracture}(\textit{John faced Mary}, \text{ORIENT}(\textbf{John}, \text{TOWARD}(\textbf{Mary})))$$
$$\Downarrow$$
$$\begin{array}{ll}
\text{(i)} & (\textit{John} = \perp \wedge \texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(\textbf{John}, \text{TOWARD}(\textbf{Mary}))))\vee \\
\text{(ii)} & (\textit{John} = \textbf{John} \wedge \texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(x, \text{TOWARD}(\textbf{Mary}))))\vee \\
\text{(iii)} & (\textit{John} = \textbf{Mary} \wedge \texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(\textbf{John}, \text{TOWARD}(x))))\vee \\
\text{(iv)} & (\textit{John} = \text{TOWARD}(\textbf{Mary}) \wedge \texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(\textbf{John}, x)))
\end{array}$$

In case (i), when *John* is assigned $\perp$ as its meaning, *Mary* can then obviously take on as its meaning any of the four subexpressions of ORIENT(**John**, TOWARD(**Mary**)).

$$\texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(\textbf{John}, \text{TOWARD}(\textbf{Mary})))$$
$$\Downarrow$$
$$\begin{array}{c}
(\textit{Mary} = \perp \wedge \textit{faced} = \text{ORIENT}(\textbf{John}, \text{TOWARD}(\textbf{Mary})))\vee \\
(\textit{Mary} = \textbf{John} \wedge \textit{faced} = \text{ORIENT}(x, \text{TOWARD}(\textbf{Mary})))\vee \\
(\textit{Mary} = \textbf{Mary} \wedge \textit{faced} = \text{ORIENT}(\textbf{John}, \text{TOWARD}(x)))\vee \\
(\textit{Mary} = \text{TOWARD}(\textbf{Mary}) \wedge \textit{faced} = \text{ORIENT}(\textbf{John}, x))
\end{array}$$

In case (ii), when *John* is assigned **John** as its meaning, *Mary* can take on three possible meanings.

$$\texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(x, \text{TOWARD}(\textbf{Mary})))$$
$$\Downarrow$$
$$\begin{array}{c}
(\textit{Mary} = \perp \wedge \textit{faced} = \text{ORIENT}(x, \text{TOWARD}(\textbf{Mary})))\vee \\
(\textit{Mary} = \textbf{Mary} \wedge \textit{faced} = \text{ORIENT}(x, \text{TOWARD}(y)))\vee \\
(\textit{Mary} = \text{TOWARD}(\textbf{Mary}) \wedge \textit{faced} = \text{ORIENT}(x, y))
\end{array}$$

In case (iii), when *John* is assigned **Mary** as its meaning, *Mary* can take on two possible meanings.

$$\texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(\textbf{John}, \text{TOWARD}(x)))$$
$$\Downarrow$$
$$\begin{array}{c}
(\textit{Mary} = \perp \wedge \textit{faced} = \text{ORIENT}(\textbf{John}, \text{TOWARD}(x)))\vee \\
(\textit{Mary} = \textbf{John} \wedge \textit{faced} = \text{ORIENT}(x, \text{TOWARD}(y)))
\end{array}$$

[6]The astute reader may wonder why a fifth possibility is not considered where the entire expression ORIENT(**John**, TOWARD(**Mary**)) is associated with *John* and the meaning of *faced Mary* is taken to be simply the variable x. MAIMRA adopts an additional restriction that does not allow a head to take on a meaning that is simply a variable, thus ruling out this fifth possibility. This restriction can be interpreted as stating that every head must contribute some semantic content to the meaning of its parent phrase. The motivation for this restriction is simply computational efficiency. Adopting this restriction reduces the ambiguity introduced during the fracturing process. The downside of this restriction is that it rules out the standard analysis of the preposition *of*. In this analysis, *of* it is treated simply as a case marker such that the meaning of the phrase *of NP* would be taken to be the same as the meaning of the NP. This requires taking the meaning of *of* to be simply the variable x, in contradiction to the above restriction.

In case (iv), when *John* is assigned TOWARD(**Mary**) as its meaning, *Mary* can also take on two possible meanings.

$$\texttt{fracture}(\textit{faced Mary}, \text{ORIENT}(\textbf{John}, x))$$
$$\Downarrow$$
$$(\textit{Mary} = \bot \wedge \textit{faced} = \text{ORIENT}(\textbf{John}, x)) \vee$$
$$(\textit{Mary} = \textbf{John} \wedge \textit{faced} = \text{ORIENT}(x, y))$$

Putting this all together yields the following disjunctive lexicon formula.

```
(or (and John = ⊥
        (or (and Mary = ⊥
                 faced = ORIENT(John, TOWARD(Mary)))
            (and Mary = John
                 faced = ORIENT(x, TOWARD(Mary)))
            (and Mary = Mary
                 faced = ORIENT(John, TOWARD(x)))
            (and Mary = TOWARD(Mary)
                 faced = ORIENT(John, x))))
    (and John = John
        (or (and Mary = ⊥
                 faced = ORIENT(x, TOWARD(Mary)))
            (and Mary = Mary
                 faced = ORIENT(x, TOWARD(y)))
            (and Mary = TOWARD(Mary)
                 faced = ORIENT(x, y))))
    (and John = Mary
        (or (and Mary = ⊥
                 faced = ORIENT(John, TOWARD(x)))
            (and Mary = John
                 faced = ORIENT(x, TOWARD(y)))))
    (and John = TOWARD(Mary)
        (or (and Mary = ⊥
                 faced = ORIENT(John, x))
            (and Mary = John
                 faced = ORIENT(x, y)))))
```

The fracturing procedure actually used by MAIMRA is slightly more complex than the above procedure, in two ways. First, it is extended to accept disjunctive parse trees. Fracturing a disjunctive parse tree fragment with a meaning expression fragment is simply the disjunction of the result of fracturing each disjunct in the disjunctive parse tree fragment with the same meaning expression fragment. MAIMRA memoizes recursive calls to `fracture` to mirror the factored nature of the disjunctive parse tree in the resulting disjunctive lexicon formula.[7] Second, recall that to handle referential uncertainty, each input utterance is associated with a *set* of meaning expressions. MAIMRA fractures each meaning expression for the current utterance with the same disjunctive parse tree for this utterance to produce a disjunctive lexicon formula. A disjunction is formed from these formulae to yield the aggregate lexicon formula for the input utterance.

[7] Memoization eliminates multiple evaluations of a function called with the same arguments. The first time $f(x_1, \ldots, x_n)$ is called, the function is evaluated and the result stored in a table. Subsequent calls to f with with the same arguments $x_1, \ldots, x_n$ retrieve this result from the table instead of reevaluating $f(x_1, \ldots, x_n)$. An additional benefit of memoization is that multiple evaluations of a function called with the same arguments return pointers to the same copy of the result thus creating a factored representation.

John:	[N]	$\mathbf{person}_1$
Mary:	[N]	$\mathbf{person}_2$
Bill:	[N]	$\mathbf{person}_3$
cup:	[N]	$\mathbf{object}_1$
the:	[$\mathrm{N}_{\mathrm{SPEC}}$]	$\perp$
rolled:	[V]	$\mathrm{GO}(x, [_{\mathrm{Path}}\ ])$
ran:	[V]	$\mathrm{GO}(x, y)$
slid:	[V]	$\mathrm{GO}(x, [_{\mathrm{Path}}\ y, z])$
faced:	[V]	$\mathrm{ORIENT}(x, \mathrm{TO}(y))$
from:	[N,V,P]	$\mathrm{FROM}(x)$
to:	[N,V,P]	$\mathrm{TO}(x)$

Figure 4.3: The lexicon inferred by MAIMRA for the corpus from figure 4.2. Note that MAIMRA has converged to a unique word-to-meaning mapping for each word in the corpus, as well as a unique word-to-category mapping for all but two words.

Appendix A illustrates the series of disjunctive parse trees and disjunctive lexicon formulae produced by MAIMRA when processing the corpus from figure 4.2. Each lexicon formula produced corresponds to a single input utterance. MAIMRA determines the lexicon corresponding to the corpus by forming a conjunction of these lexicon formulae, conjoining this with a conjunction of monosemy formulae to implement the monosemy constraint, and finding satisfying truth assignments to the lexical entry propositions in the entire resulting formula. MAIMRA actually performs this process repeatedly as each new utterance arrives. Even though there may be multiple consistent lexica during intermediate stages when only part of the corpus has been processed, nonetheless it may be possible to rule out some word-to-category or word-to-meaning mappings. MAIMRA can use this partial information to reduce the size of structures produced when processing subsequent input utterances. I have already discussed how reduced lexical ambiguity can result in smaller disjunctive parse trees. Furthermore, reduced structural ambiguity, combined with ruling out impossible word-to-meaning mappings, can result in the production of smaller disjunctive lexicon formulae. This is evident when comparing the lexicon formula corresponding to *Bill ran to Mary* on page 211 with the lexicon formula corresponding to *Bill ran from John* on page 214. Though the input utterances are similar, and are paired with analogous meaning expressions, the latter utterance yields a smaller disjunctive lexicon formula due to the knowledge gleaned from prior input.

Using the above techniques, MAIMRA can successfully derive the lexicon shown in figure 4.3 from the corpus given in figure 4.2. Inferring this lexicon requires several minutes of elapsed time on a Symbolics XL1200^{TM} computer. Thus MAIMRA converges to a unique and correct meaning for every word in the corpus as well as a unique and correct syntactic category for all but two of the words in the corpus.

From a theoretical perspective, the lexicon produced by MAIMRA is independent of the order in which the corpus is processed. This is because each utterance in the corpus is processed to yield a lexicon formula which characterizes those lexica that are consistent with that utterance. MAIMRA simply conjoins those formulae to find a lexicon consistent with the entire corpus. As a practical matter, however, the computational complexity of the learning algorithm is affected by the processing order, since MAIMRA uses previously acquired knowledge to reduce the size of subsequently generated lexicon formulae. MAIMRA works best if the corpus is ordered so that shorter utterances and utterances with fewer unknown words appear first.

4.2 Davra

Despite MAIMRA's success in inferring a lexicon from semantically annotated input utterances, the theory underlying MAIMRA suffers from two severe limitations that preclude it from being a complete account of child language acquisition. First, MAIMRA relies on a fixed context-free grammar being available prior to the lexicon acquisition process. It appears unreasonable to assume that children know the grammar of their native language before they learn the syntactic categories or meanings of any words. More likely, they must learn the grammar either along with, or subsequent to, the lexicon. Second, MAIMRA has been tested only on an English corpus. A satisfying theory of language acquisition must be capable of acquiring any human language, not just English.

In attempt to rectify the above two shortcomings, a second system called DAVRA (Siskind 1991) was constructed. DAVRA is very similar to MAIMRA in many ways. Both represent word, phrase, and utterance meanings using the same form of Jackendovian conceptual structure meaning expressions. Furthermore, both receive input in the same form: a corpus of utterances, each paired with a set of potential meanings for that utterance. Thus DAVRA, like MAIMRA, learns in the presence of referential uncertainty. DAVRA differs from MAIMRA however, in basing its syntactic theory on a parameterized version of $\overline{\mathrm{X}}$ theory rather than on a fixed context-free grammar given as input to the learner. DAVRA's innate endowment includes the formulation of $\overline{\mathrm{X}}$ theory, embodied in the acquisition model, but does not include the parameter settings particular to the language being learned. DAVRA acquires the parameter settings from the corpus, simultaneously with the lexicon, using the cross-situational learning architecture described in section 2.1. Thus DAVRA learns three things—parameter settings, word-to-category mappings, and word-to-meaning mappings—without any prior knowledge of such parameter settings or mappings.

The variant of $\overline{\mathrm{X}}$ theory incorporated into DAVRA can be summarized as follows.

1. The syntactic structures constructed by DAVRA are binary branching. Each node has zero, one, or two children. Nodes with no children are *terminals*. Nodes with one or two children are *head-complement* structures. One child of a head-complement structure is always the *head*. The remaining child, if present, is its *complement*.

2. DAVRA labels each node with one of the category labels X, $\mathrm{X}_{\mathrm{SPEC}}$,[8] $\overline{\mathrm{X}}$, or XP, where X is one of the base categories N, V, P, or I.

3. Terminals must labeled with either $\mathrm{X}_{\mathrm{SPEC}}$ or X for some base category X.

4. Non-terminal nodes take on one of the following five configurations

where X and Y freely range over the base categories. The nodes enclosed in boxes indicate which child is taken to be the head of a head-complement structure as far as the linking rule is concerned.

[8] The linguistic literature has waffled somewhat over the term SPEC, sometimes considering it to be a category label, or class of category labels such as determiner, and other times taking it to be the name of a position, where determiners, among other things, can appear. DAVRA takes $\mathrm{X}_{\mathrm{SPEC}}$ to be a class of category labels—$\mathrm{N}_{\mathrm{SPEC}}$, for instance, being a synonym for DET. This is a somewhat outdated approach to $\overline{\mathrm{X}}$ theory. In contrast, KENUNIA takes SPEC to be a position, namely the non-adjunct sister to a node of bar-level one. This approach is more in line with the variant of $\overline{\mathrm{X}}$ theory presented in Chomsky (1985). DAVRA should not be considered *a priori* incorrect because of this. Many current authors still adopt the former position (cf. Lightfoot 1991 pp. 186–187).

A given language will allow only a subset of the above five structures however. One binary-valued parameter determines whether the language is *SPEC-initial* or *SPEC-final*. Structure (c) is not allowed if the language is SPEC-initial, while structure (b) is not allowed if the language is SPEC-final. A second binary-valued parameter determines whether the language is *head-initial* or *head-final*. Structure (e) is not allowed if the language is head-initial, while structure (d) is not allowed if the language is head-final.

5. The top-level node corresponding to an input utterance must be labeled IP.

6. The category label $\mathrm{I}_{\mathrm{SPEC}}$ is taken to be a synonym for the category label NP.

7. The category label $\overline{\mathrm{I}}$ is taken to be a synonym for the category label VP.

In addition to the above variant of $\overline{\mathrm{X}}$ theory, DAVRA incorporates the linking rule given in section 3.1. This linking rule is simplified in DAVRA since, unlike MAIMRA's syntactic theory, DAVRA's syntactic theory allows only binary branching structures.[9] Furthermore, like MAIMRA, DAVRA adopts two additional restrictions First, the meaning expressions associated with complements must be variable-free. This eliminates the need to rename variables during the linking process. Second, the meaning expression associated with a head must not be simply a variable. With these restrictions, the linking rule incorporated into DAVRA can be summarized by the following five cases

where the nodes enclosed in boxes indicate the heads of head-complement structures. Case (i) is used for unary branching structures of type (a). Both cases (ii) and (iv) apply to SPEC-final structures like (c) and head-initial structures like (d), while both cases (iii) and (v) apply to SPEC-initial structures like (b) and head-final structures like (e). For example, in English, a head-initial language, case (ii) would be used to derive the meaning of *from John*, namely FROM(**John**), from FROM(x) and **John**, the meanings of *from* and *John* respectively. Likewise, case (v) would be used to derive the meaning of *the book*, namely **book**, from $\perp$ and **book**, the meanings of *the* and *book* respectively. In Japanese, a head-final language, case (ii) would be used to derive the meaning of *Taro kara*, namely FROM(**Taro**), from **Taro** and FROM(x), the meanings of *Taro* and *kara* respectively.

The nodes in the syntactic tree constructed by DAVRA correspond to substrings of the input utterance in the standard fashion that disallows crossovers. DAVRA allows *non-overt* nodes, i.e. nodes that correspond to empty substrings. Both terminal and non-terminals nodes may be non-overt. DAVRA enforces the constraint that overt terminal nodes correspond to a single word of the input utterance. Furthermore, DAVRA enforces several additional constraints designed to reduce the size of the search space in the underlying language acquisition task. First, nodes labeled $\overline{\mathrm{X}}$ must be overt. Second, non-overt nodes must be assigned $\perp$ as their meaning. Stated informally, this means that non-overt phrases cannot contribute substantive semantic content to an utterance. Finally, any node labeled XP cannot be assigned $\perp$ as its meaning.

For reasons of simplicity, DAVRA does not generate disjunctive lexicon formulae the way MAIMRA does. Instead, the design of DAVRA directly follows the architecture from figure 2.2. DAVRA retains the entire corpus in memory and tries to find a lexicon and a set of parameter settings that are consistent across this corpus. DAVRA employs straightforward blind search to find this lexicon and set of parameter

[9] Restricting the linking rule to binary branching structures is not a severe limitation. Most current variants of $\overline{\mathrm{X}}$ theory adopt the binary branching restriction as it appears to be sufficient to describe the requisite syntactic phenomena.

settings. The motivation behind the design of DAVRA was not the construction of an accurate process model of child language acquisition. DAVRA's use of blind search over a corpus retained in memory is not a plausible process model. It does, however, allow one to determine whether a linguistic theory of the form described above, namely parameterized $\overline{\mathrm{X}}$ theory, offers enough constraint to uniquely determine the lexicon and parameter settings when supplied with a very small corpus. Only once it has been determined that the theory is sufficiently constraining does it make sense to explore more efficient and plausible search algorithms.

The linguistic theory incorporated in DAVRA can be phrased as a simple nondeterministic program that describes the search space for possible lexica and parameter settings. This program, which I will call `fracture`, operates in a top-down divide-and-conquer fashion where nondeterministic choices are made at each divide-and-conquer step. Backtracking through these nondeterministic choices allows straightforward though inefficient search for possible solutions. The divide-and-conquer steps interleave a top-down parsing strategy with the fracturing procedure discussed in section 3.1.

One such nondeterministic path through the divide-and-conquer sequence is illustrated in figure 4.4. For each divide-and-conquer step, `fracture` is called with three arguments: a phrase, a meaning expression to be associated with that phrase, and a category label for that phrase. At the top level, `fracture` is called with an input utterance paired nondeterministically with one of its possible meanings. The input utterance is labeled with the category IP.

Several nondeterministic choices are made at each recursive call to `fracture`. First, the phrase is split into two subphrases. For example, the input phrase *The cup slid from John to Mary* might be split into the subphrases *The cup* and *slid from John to Mary*. The split point is chosen nondeterministically. Second, the SPEC-initial parameter is nondeterministically set to **true**. This allows the first subphrase to be assigned the category $\mathrm{I}_{\mathrm{SPEC}}$, which is treated as NP, and the second subphrase to be assigned the category $\overline{\mathrm{I}}$, which is treated as VP. Since $\overline{\mathrm{I}}$ is the head of IP, some subexpression of $\mathrm{GO}(\mathbf{cup}, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{John}), \mathrm{TO}(\mathbf{Mary})])$ is nondeterministically selected, namely **cup**, and associated with the first subphrase, as this subphrase is the complement. The subexpression **cup** is then extracted from $\mathrm{GO}(\mathbf{cup}, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{John}), \mathrm{TO}(\mathbf{Mary})])$, leaving a variable behind, to yield the expression $\mathrm{GO}(x, [_{\mathrm{Path}}\ \mathrm{FROM}(\mathbf{John}), \mathrm{TO}(\mathbf{Mary})])$. This meaning expression fragment is then assigned to the head subphrase. The `fracture` routine is then recursively called on each of the two subphrases with their associated meaning expression fragments and category labels. This recursive process terminates when `fracture` is called on a singleton word. In this case, a lexical entry is created mapping the word to the given meaning expression and syntactic category label. Figure 4.4 illustrates two such mappings: one from the word *the* to the category label $\mathrm{N}_{\mathrm{SPEC}}$ and meaning expression $\perp$, and one from the word *cup* to the category label N and meaning expression **cup**.

The `fracture` routine makes many nondeterministic choices at each step. For pedagogical purposes, figure 4.4 illustrates a path containing the correct choices, though many alternative paths contain incorrect choices that are filtered out by backtracking. Backtracking is initiated by two types of failure. One type occurs when an attempt is made to set a parameter to a different setting than has already been made. The linguistic theory incorporated into DAVRA states that a given language is either head-initial or head-final but not both. The second type occurs when an attempt is made to create a lexical entry for a word which assigns it a different meaning or syntactic category than it has already been assigned. This is an embodiment of the monosemy constraint.

The nondeterministic search process just described can be written as a program in nondeterministic LISP(Siskind and McAllester 1992). This program is really quite small and modular. An annotated description of the essential routines in this program is given below. It can be seen that this program straightforwardly embodies the linguistic theory stated above.

```
(defun fracture (words category meaning)
 (declare (special categories head-initial? spec-initial? lexicon))
```

Figure 4.4: DAVRA incorporates a divide-and-conquer search strategy illustrated by this figure. This process is embodied in a recursive routine called `fracture` which takes three arguments: a phrase, a meaning expression fragment, and a category label. First, the phrase is nondeterministically split into two subphrases. Next, the meaning expression fragment is nondeterministically split into two submeanings, one to be assigned to each subphrase. Finally, $\overline{\mathrm{X}}$ theory determines the category labels to assign to each subphrase given the input category label. Each subphrase is then recursively fractured with its associated submeaning and category label. The recursion terminates when a single word is assigned a category and meaning. There may be many possible divide-and-conquer paths due to nondeterminism. This figure illustrates just a portion of one such path, the correct one. DAVRA enumerates all possible divide-and-conquer paths to find those that contain consistent parameter settings, as well as consistent word-to-category and word-to-meaning mappings, across the entire corpus.

The essence of DAVRA is the routine `fracture`. `Fracture` attempts to assign a syntactic `category` label and `meaning` expression fragment to a list of `words`. The basic strategy is top down: nondeterministically split `words` into two phrases, a head and a complement; nondeterministically assign part of the parent `meaning` to the head and part to the complement according to the linking rule; and recursively call `fracture` on both the head and complement. This routine uses four pieces of information global to the language acquisition process: the base `categories` that project into the $\overline{\mathrm{X}}$ system, a flag indicating whether the language is `head-initial?` or final, another flag indicating whether the language is `spec-initial?` or final, and the `lexicon`, a map from words to their syntactic categories and meanings.

```
(if (and (consp category) (eq (second category) 'p) (eq meaning '⊥)) (fail))
```

The above statement implements the third additional restriction, namely that a node labeled XP cannot have $\perp$ as its meaning.

```
(if (and (null words) (not (eq meaning '⊥))) (fail))
```

The above statement implements the second additional restriction, namely that non-overt nodes must be assigned $\perp$ as their meaning.

```
(cond
  ((equal category '(i spec)) (fracture words '(n p) meaning))
  ((equal category '(i bar)) (fracture words '(v p) meaning))
```

There are five cases in the `fracture` routine. The above two cases implement principles 6 and 7 of the variant of $\overline{\mathrm{X}}$ theory presented on page 55 (that $\mathrm{I}_{\mathrm{SPEC}}$ is processed as NP and that $\overline{\mathrm{I}}$ is processed as VP).

```
  ((and (consp category) (eq (second category) 'bar))
   (either
    (fracture words (first category) meaning)
```

The third case handles phrases of type $\overline{\mathrm{X}}$. A node of category $\overline{\mathrm{X}}$ can be either unary or binary branching. A nondeterministic choice is made between the two by the `either` clause. The above statement handles the case of unary branching.

```
    (let* ((split (split words))
           (head (if head-initial? (first split) (second split)))
           (complement (if head-initial? (second split) (first split))))
     (if (null head) (fail))
     (if (null complement) (fail))
     (let ((complement-meaning (possible-complement-meaning meaning)))
      (fracture complement `(,(member-of categories) p) complement-meaning)
      (fracture
       head category (possible-head-meaning complement-meaning meaning))))))
```

The above statement implements the second alternative for phrases of type $\overline{\mathrm{X}}$. It nondeterministically splits the phrase into two halves, one to become the `head`, the other to become the `complement`. The choice of which half becomes the `head`, and which the `complement`, is determined by the `head-initial?` parameter. Note that the `head` must not be null, since the first additional restriction states that nodes labeled $\overline{\mathrm{X}}$ must be overt. Furthermore, the `complement` must not be null, since complements are labeled XP, nodes labeled XP cannot have $\perp$ as their meaning, and non-overt nodes must mean $\perp$. The routines `possible-complement-meaning` and `possible-head-meaning` implement the linking process in reverse. Given a parent `meaning`, they nondeterministically return all possible head meanings and `complement-meaning`s that can combine to form the parent `meaning`. They will be described in greater

detail later. Two recursive calls are made to `fracture`, one to fracture the `complement` as a phrase of the category YP, nondeterministically for some base category Y, and one to fracture the `head` as a phrase of category $\overline{\mathrm{X}}$.

```
((and (consp category) (eq (second category) 'p))
 (let* ((split (split words))
        (head (if spec-initial? (second split) (first split)))
        (complement (if spec-initial? (first split) (second split))))
  (if (null head) (fail))
  (let ((complement-meaning (possible-complement-meaning meaning)))
   (fracture complement `(,(first category) spec) complement-meaning)
   (fracture
    head
    `(,(first category) bar)
    (possible-head-meaning complement-meaning meaning)))))
```

The fourth case handles phrases of type XP. Like before, it nondeterministically splits the phrase into two halves, one to become the `head`, the other to become the `complement` (in this case actually the specifier). The choice of which half becomes the `head`, and which the `complement`, is determined by the `spec-initial?` parameter. Again, note that the `head` must not be null, since the first additional restriction states that nodes labeled $\overline{\mathrm{X}}$ must be overt. Like before, the parent `meaning` is nondeterministically divided into a head meaning and an `complement-meaning`. Two recursive calls are made to `fracture`, one to fracture the `complement` as a phrase of category $\mathrm{X}_{\mathrm{SPEC}}$ and one to fracture the `head` as a phrase of category $\overline{\mathrm{X}}$.

```
((or (and (consp category) (eq (second category) 'spec)) (symbolp category))
 (unless (null words)
  (unless (null (rest words)) (fail))
  (let* ((new-definition (list category (canonicalize-meaning meaning)))
         (old-definition (gethash (first words) lexicon)))
   (if old-definition
       (unless (equal new-definition old-definition) (fail))
       (locally-setf (gethash (first words) lexicon) new-definition)))))))
```

The final case handles terminals. According to principle 3 of the variant of $\overline{\mathrm{X}}$ theory presented on page 55, categories $\mathrm{X}_{\mathrm{SPEC}}$ and base categories X are terminal. A lexical entry comprising a syntactic category and meaning is created. If this word already has a different lexical entry then enforce the monosemy constraint by failing. If a terminal is non-overt no lexical entry is added to the lexicon.

```
(defun subexpression (expression)
 (if (consp expression)
     (either expression (subexpression (member-of (rest expression))))
     expression))

(defun possible-complement-meaning (parent-meaning)
 (either '⊥
         (let ((complement-meaning (subexpression parent-meaning)))
          (unless (variable-free? complement-meaning) (fail))
          (if (equal complement-meaning parent-meaning) (fail))
          complement-meaning)))
```

The function `subexpression` nondeterministically returns some subexpression of an `expression`. The function `possible-complement-meaning` implements half of the inverse linking rule. It returns possible `complement-meaning`s that can link with an appropriate head meaning to yield the `parent-meaning`. Such a `complement-meaning` can be either $\perp$, or some subexpression of the `parent-meaning`. Remember that the linking rule carries two stipulations. First, meanings of complements must be variable-free. Thus `complement-meaning`s containing variables are filtered out. Second, a head cannot have a meaning which is just a variable. If the `complement-meaning` were to be the same as the `parent-meaning`, then the head meaning would have to be just a variable. Thus, `complement-meaning`s which are the same as the `parent-meaning` are filtered out.

```
(defun variable-substitute (subexpression expression variable)
 (cond ((equal expression subexpression) (either variable expression))
       ((consp expression)
        (cons (variable-substitute subexpression (car expression) variable)
              (variable-substitute subexpression (cdr expression) variable)))
       (t expression)))

(defun possible-head-meaning (complement-meaning parent-meaning)
 (if (eq complement-meaning '⊥)
     parent-meaning
     (let ((head-meaning
            (variable-substitute
             complement-meaning
             parent-meaning
             (make-variable (1+ (highest-variable parent-meaning))))))
      (if (equal head-meaning parent-meaning) (fail))
      head-meaning)))
```

The function `variable-substitute` takes a meaning `expression` and returns a similar expression where subexpressions of that expression which are equal to `subexpression` are nondeterministically either replaced, or not replaced, by a `variable`. The function `possible-head-meaning` implements the other half of the inverse linking rule. It returns possible `head-meaning`s that can link with a given `complement-meaning` to yield the `parent-meaning`. If the `complement-meaning` is $\perp$ then the `head-meaning` is the same as the `parent-meaning`. Otherwise, we nondeterministically substitute a new variable for occurrences of the `complement-meaning` within the `parent-meaning`. Note that since the linking rule requires that the complement meaning be substituted for *some* variable in the head meaning, when doing the nondeterministic inverse substitution of a variable for occurrences of the `complement-meaning` in the `parent-meaning`, we must guarantee that at least one such substitution has occurred. We must filter out a `head-meaning` that is equal to the `parent-meaning` since a substitution has not occurred.

Davra was presented with the same corpus that was given to Maimra. This corpus is illustrated in figure 4.2. This corpus consists of nine multi-word utterances ranging in length from two to seven words. Each utterance is paired with between three and six possible meaning expressions. Given this corpus, Davra is able to learn the lexicon and parameter settings given in figure 4.5. Inferring this information requires about an hour of elapsed time on a Symbolics XL1200™ computer. Note that Davra determines that the linguistic theory allows the corpus to have only one consistent analysis where the language is head-initial and SPEC-initial. Furthermore, the theory and corpus together fully determine most of the lexicon. Davra finds unique mappings for all words to their associated meaning expressions and for all but two words to their associated syntactic categories. For example, the linguistic theory generates the corpus only under the assumption that *cup* is a noun which means $\mathbf{object}_1$ and *slid*

Head Initial, SPEC Initial.		
John:	[N]	$\mathbf{person}_1$
Mary:	[N]	$\mathbf{person}_2$
Bill:	[N]	$\mathbf{person}_3$
cup:	[N]	$\mathbf{object}_1$
the:	[$\mathrm{N}_{\mathrm{SPEC}}$]	$\perp$
rolled:	[V]	$\mathrm{GO}(x, [_{\mathrm{Path}}\ ])$
ran:	[V]	$\mathrm{GO}(x, y)$
slid:	[V]	$\mathrm{GO}(x, [_{\mathrm{Path}}\ y, z])$
faced:	[V]	$\mathrm{ORIENT}(x, \mathrm{TO}(y))$
from:	[N,V,P]	$\mathrm{FROM}(x)$
to:	[N,V,P]	$\mathrm{TO}(x)$

Figure 4.5: The lexicon and parameter settings inferred by DAVRA for the corpus from figure 4.2. Note that DAVRA has uniquely determined that English is head-initial and SPEC-initial. Furthermore, DAVRA has converged to a unique word-to-meaning mapping for each word in the corpus, as well as a unique word-to-category mapping for all but two words.

is a verb which means $\mathrm{GO}(x, [_{\mathrm{Path}}\ y, z])$. The only language-specific information which DAVRA is not able to converge on is the syntactic category of the words *from* and *to*. It is easy to see that DAVRA can never uniquely determine that prepositions like *from* and *to* should be labeled with category P since according to the linguistic theory incorporated into DAVRA, words labeled N and V can co-occur anywhere words labeled with category P can appear. This is a shortcoming of DAVRA that can be addressed by the addition of case theory and c-selection principles. Case theory includes a *case filter* which states that overt noun phrases must receive *case*, an abstract property assigned by certain lexical items to certain complement positions. The case filter would not allow *from* to be labeled with category N since nouns do not assign case to their complement and thus the noun phrase *John* in *Bill ran from John* would not be assigned case. C-selection principles state that certain categories must appear as complements of other specific categories. For example, a verb phrase must appear as the complement of an inflection. This principle would not allow *from* to be labeled with category V since *from John* does not appear as the complement of an inflectional element in *Bill ran from John*. The next section will discuss KENUNIA, a system built subsequent to DAVRA, that incorporates such additional linguistic constraints.

As discussed previously, one of the main objectives for DAVRA was to construct a single linguistic theory that could acquire lexica and parameter settings for different languages. To test the cross-linguistic applicability of DAVRA, the corpus in figure 4.2 was translated from English to Japanese, retaining the same non-linguistic annotation.[10] The resulting linguistic component of the Japanese corpus is illustrated in figure 4.6. Note that the syntax of Japanese differs from English in a number of key ways. First, Japanese is a head-final language; prepositions follow their complements (and are thus really postpositions) and the underlying word order is subject-object-verb. Second, Japanese subjects are generally marked with the word *ga*. Third, the Japanese word *tachimukau* takes a prepositional phrase complement (i.e. *Eriko ni*) while the corresponding English word *faced* takes a direct object (i.e. *faced Mary*).

When presented with this Japanese corpus, DAVRA produced the lexicon and parameter settings given in figure 4.7. Processing this corpus took about twelve hours of elapsed time on a Symbolics XL1200™ computer. Note that DAVRA produced essentially the same result for the Japanese corpus as for the

[10] I would like to thank Linda Hershenson, Michael Caine, and Yasuo Kagawa, who graciously performed this translation for me.

Taro ga korogashimashita.
Eriko ga korogashimashita.
Yasu ga korogashimashita.
Chawan ga korogashimashita.
Yasu ga Eriko ni hashirimashita.
Yasu ga Taro kara hashirimashita.
Yasu ga chawan ni hashirimashita.
Chawan ga Taro kara Eriko ni suberimashita.
Taro ga Eriko ni tachimukau.

Figure 4.6: The linguistic component of a sample Japanese corpus presented to DAVRA. This corpus is a translation of the English corpus given in figure 4.2. The non-linguistic component of the Japanese corpus is identical to that of the English corpus.

English corpus despite the syntactic differences between the two languages. Thus DAVRA determined that Japanese was head-final but SPEC-initial, accounting for the postpositional and verb-final properties. DAVRA was not hindered by the presence of *ga*, and by assigning it the meaning expression $\perp$, determined that its meaning was outside the realm of the Jackendovian semantic representation used.[11] Just as for the English corpus, DAVRA determines unique word-to-meaning mappings for all words in the Japanese corpus, as well as unique word-to-category mappings for all but two words in that corpus. DAVRA exhibits the same limitations in Japanese as in English and is unable to narrow the possible syntactic categories assigned to prepositions like *kara* and *ni*. Notice however, that DAVRA does determine that *tachimukau* does not incorporate a path in its meaning representation (i.e. $\text{ORIENT}(x, y)$), while *faced* does (i.e. $\text{ORIENT}(x, \text{TO}(y))$), accounting for the different argument structure of these two words.

Thus DAVRA has been successful as an initial attempt to demonstrate cross-linguistic language acquisition. DAVRA has simultaneously learned syntactic parameter settings, and a lexicon mapping words to their syntactic categories and meanings, with no prior information of that type, for very small corpora in two different languages.

As was the case for MAIMRA, the language model produced by DAVRA does not depend on the order of the utterances in the corpus since DAVRA simply finds all language models consistent with the entire corpus. Again however, the complexity of the search task can heavily depend on the order in which the utterances are presented to DAVRA. The search space grows intractably large if the corpus is ordered so that earlier utterances have many consistent language models that are filtered out only by latter utterances.

4.2.1 Alternate Search Strategy for Davra

As discussed previously, one of the unsatisfying aspects of DAVRA is its use of blind search across the entire corpus retained in memory. For just this reason, this is not a plausible process model for child language acquisition. An initial experiment was undertaken to explore more plausible alternative learning strategies within the same linguistic theory used by DAVRA. A different top-level search strategy was built for DAVRA that retained the same underlying parsing mechanism. This experiment was attempted only for the English corpus. Furthermore, for this experiment, DAVRA was given the correct parameter settings as input and asked to learn only the lexicon.

[11] DAVRA assigns the category V_{SPEC} to *ga*. This is probably not linguistically accurate but nonetheless is consistent with the limited variant of $\overline{X}$ theory incorporated into DAVRA.

Head Final, SPEC Initial.		
Taro:	[N]	$\mathbf{person}_1$
Eriko:	[N]	$\mathbf{person}_2$
Yasu:	[N]	$\mathbf{person}_3$
chawan:	[N]	$\mathbf{object}_1$
ga:	[V_{SPEC}]	$\perp$
korogashimashita:	[V]	$\text{GO}(x, [_{\text{Path}}\ ])$
hashirimashita:	[V]	$\text{GO}(x, y)$
suberimashita:	[V]	$\text{GO}(x, [_{\text{Path}}\ y, z])$
tachimukau:	[V]	$\text{ORIENT}(x, y)$
kara:	[N,V,P]	$\text{FROM}(x)$
ni:	[N,V,P]	$\text{TO}(x)$

Figure 4.7: DAVRA inferred this lexicon and set of parameter settings when processing the Japanese utterances from figure 4.6 when paired with the non-linguistic input from figure 4.2. DAVRA has correctly determined that Japanese is a head-final language. Furthermore, as in figure 4.5, DAVRA has converged on a single correct meaning for all words in the corpus as well as a single correct category label for all but two words. Note that DAVRA has determined that the word *ga* has meaning outside the realm of Jackendovian conceptual structures and that *tachimukau* does not incorporate a path, in contrast to *faced* which does.

The alternate search strategy employed is weaker than strong cross-situational learning. In this strategy, DAVRA processes the input utterances one by one, retaining only two types of information between utterances: the current hypothesized lexicon and sets of previously tried inconsistent hypotheses. Once DAVRA processes an utterance, all information about that utterance is discarded, save the above two types of information. DAVRA starts out with the empty lexicon. When processing each input utterance, DAVRA searches for an extension to that lexicon that allows the current utterance to meet the constraints imposed by the linguistic theory and non-linguistic input. The extension must obey the monosemy constraint, i.e. new words can be assigned an arbitrary lexical entry but words encountered in previous utterances must be interpreted according to the lexical entries already in the lexicon. There may be several different extensions, i.e. several different assignments of lexical entries to novel words, which are consistent with the current utterance. In this case, DAVRA arbitrarily picks only one consistent extension. If DAVRA is successful in extending the lexicon to account for the new utterance, the extension is adopted, the utterance discarded, and processing continues with the next utterance. (The extended lexicon might be the same as the previous lexicon if the input utterance does not contain novel words and can be parsed with the existing lexicon.)

More often, DAVRA is unsuccessful in finding a consistent extension, as would happen if DAVRA previously selected the wrong extension, thus making incorrect hypotheses about lexical entries. In this case, DAVRA attempts to find a small subset of the lexicon that is inconsistent with the current utterance. Such a subset of the lexicon is termed a *nogood* because it rules out any superset of that subset as a potential hypothesized lexicon. In particular, DAVRA finds a nogood N such that no extension of N allows the current utterance to be parsed, yet removing any single lexical entry from N yields an N' which can be extended to parse the current utterance. A nogood that has this property is called a *minimal* nogood. DAVRA constructs minimal nogoods by a simple linear process when the current lexicon cannot be extended to parse the input utterance. DAVRA starts out taking the entire current lexicon as the initial nogood. Lexical entries are removed from this initial nogood one by one and the resulting nogood tested to see whether it can be extended to parse the input utterance. If it can, the lexical entry just

dropped is put back in the initial nogood. Otherwise, it is discarded. It is easy to see that this linear process will produce a minimal nogood with the two aforementioned properties.

Two things are then done with the nogood just constructed. First, it is saved on a list of discovered nogoods. Whenever, DAVRA later extends the lexicon, the extended lexicon is checked to see that it is not a superset of any previously created nogood. Extensions that are supersets of some nogood are not considered. In this way DAVRA is guaranteed not to make the same mistake twice. Second, one lexical entry is selected arbitrarily from the current nogood. This lexical entry is removed from the current lexicon and a new attempt is made to extend the resulting lexicon to parse the current input utterance.

The revised search strategy used by DAVRA is similar in many ways to Mitchell's (1977) version space learning algorithm. Mitchell's algorithm was originally formulated for the *concept learning problem*, a more general task than language acquisition. In concept learning, the learner is presented with a stream of *instances* from some *instance space*. Each input instance is labeled as either an *positive* or *negative* instance of the concept to be learned. A *concept* is a total predicate C such that $C(x)$ returns **true** if x is an instance of the concept and **false** otherwise. Concepts are chosen from a finite set $\mathcal{C}$ called the *concept space*. The task faced by the concept learner is to select those $C \in \mathcal{C}$ such that $C(x)$ is **true** for each positive instance in the training set and **false** for each negative instance in the training set. Such a concept is said to *cover* the training set. Though general concept learning allows both positive and negative instances to appear in the input, I consider here only the restricted problem which utilizes positive input instances, since only that portion is relevant to the comparison with the search strategy used by DAVRA. Mitchell's version space algorithm operates as follows. First, a concept C' is called more *general* than a concept C if for all x in the instance space, $C(x) \rightarrow C'(x)$. Likewise, a concept C' is called more *specific* than a concept C if for all x in the instance space, $C'(x) \rightarrow C(x)$. As Mitchell's algorithm processes the instance one by one, it maintains a set S of concepts that satisfies two properties. First, each concept $C \in S$ must cover the set of instances processed so far. Second, for each concept $C \in S$ there cannot be a more specific concept $C' \in \mathcal{C}$ that also covers the set of instances processed so far. These properties are met by initializing S to contain the most specific concepts in $\mathcal{C}$ and updating S after processing each instance x by replacing those $C \in S$ for which $C(x)$ returns **false** with the most specific generalizations C' of C where $C'(x)$ returns **true**.

During the operation of Mitchell's algorithm, the target concept must always be more general than every element of S. Furthermore, any concept that is strictly less general than some element of S can be ruled out as a potential target concept. The set S can be seen as a border, dividing the concept space $\mathcal{C}$ into two regions, one containing potential target concepts, the other containing those concepts ruled out as potential target concepts. The ability for S to rule out potential concepts is analogous to the set of nogoods used by DAVRA's revised search strategy. The analogy can be made explicit as follows. Each utterance paired with its non-linguistic input is an instance of the concept to be learned, where concepts are language models. A language model returns **true** for an instance if some extension of that model allows the instance to be parsed. One language model is more general than another if the former is a subset of the latter. The set of nogoods maintained by DAVRA corresponds to S with one minor variation: S contains the most specific concepts which cover the input while a nogood is a most general concept which does not cover the input. The set of nogoods maintained by DAVRA thus constitutes one side of the border of the region bounding potential target concepts while S constitutes the other side. Modulo these differences, this border can be considered a *frontier*, which we can take to be on the same side of the border as in Mitchell's algorithm. Mitchell's algorithm uses the frontier to constrain the region of potential target concepts. DAVRA however, uses the frontier only to rule out potential target concepts. For reasons of efficiency, DAVRA maintains a less tight frontier than does Mitchell's algorithm, ruling out fewer potential target concepts. This less tight frontier is the result of the following differences between DAVRA and Mitchell's algorithm. First, Mitchell's algorithm initializes the frontier to contain all of the most specific concepts in $\mathcal{C}$. DAVRA's initial frontier consists of a single concept, the current language model. Second, Mitchell's algorithm replaces all elements of the frontier

with their generalizations when those elements do not cover an input instance. DAVRA replaces only one element of the frontier, namely the current language model, with its generalizations, when it does not cover an input instance. Finally, when an element of the frontier does not cover an input instance, Mitchell's algorithm replaces it with all of the most specific generalizations which do cover the input instance. DAVRA replaces such an element with only one most specific generalization which covers the input instance.

The aforementioned strategy was applied to the English corpus from figure 4.2. Since this strategy is weaker than strong cross-situational learning, the corpus is too short to allow DAVRA to converge to a correct lexicon. In the absence of a larger corpus, the existing corpus was repeatedly applied as input to the alternate strategy until DAVRA was able to make a complete pass through the corpus without needing to retract any lexical entries. DAVRA required two passes through the corpus in figure 4.2 for convergence and produced the same lexicon as shown in figure 4.5 as output. This strategy required only a few minutes of elapsed time on a Symbolics XL1200™ computer.

Note that, as formulated above, this strategy simply finds a single consistent lexicon. It does not determine that the linguistic theory and corpus imply a unique solution. One could extend this technique to determine all solutions by temporarily ruling out each solution as it was found and continuing the search for further solutions. This is done by considering each solution to be a nogood. No further solutions can be found when the empty nogood is produced. While it may be expensive to determine all solutions, a variant of this technique can be used to determine whether or not the learner has converged to a unique solution by simply checking whether a single additional solution exists. Also note that unlike the original implementation of DAVRA, the rate of convergence of this revised search strategy is dependent on the order in which utterances are processed. Future work will attempt to quantify the sensitivity of this search strategy to corpus ordering.

4.3 Kenunia

Like MAIMRA, DAVRA also suffers from a number of shortcomings that limit its viability as a complete theory of child language acquisition. Accordingly, I have constructed a third system, KENUNIA that attempts to address some of these shortcomings.

4.3.1 Overview of Kenunia

The following summarizes the limitations in DAVRA addressed by KENUNIA.

- DAVRA's syntactic theory is specified by setting two binary-valued parameters: head-initial/final and SPEC-initial/final. Thus except for lexical differences, DAVRA can support only four distinct language types. In KENUNIA, the analog of the head-initial/final and SPEC-initial/final parameters vary on a category by category basis, increasing the possible parametric diversity of languages to be learned. Furthermore, since KENUNIA supports base adjunction, additional parameters specify the adjunction order, again on a category by category basis. The KENUNIA syntactic theory is specified by setting sixteen binary-valued parameters, supporting 65,536 distinct possible languages types to be learned, independent of lexical variation.

- The syntactic theory incorporated into DAVRA is little more than $\overline{\mathrm{X}}$ theory. KENUNIA instead incorporates a much more substantial linguistic theory including $\overline{\mathrm{X}}$ theory, movement, θ-theory, case theory, and the empty category principle (ECP). While the variant of $\overline{\mathrm{X}}$ theory incorporated into DAVRA supports only head-complement structures over the categories N, V, P, and I, the variant incorporated into KENUNIA supports both head-complement structures, as well as free base adjunction, over the categories N, V, P, D, I, and C. Furthermore, the syntactic theory used

by KENUNIA incorporates a number of current linguistic notions, such as VP-internal subjects, the DP hypothesis, and V-to-I movement.

- DAVRA supports only a weak notion of empty category. DAVRA allows a terminal node to be non-overt so long as it does not contribute any semantic content to the resulting utterance. KENUNIA extends this capacity to provide for both non-overt words, as well as movement and its ensuing traces. KENUNIA incorporates the general notion of an *empty terminal*, a terminal with no overt phonological content. KENUNIA supports two types of empty terminals: *traces* which are bound by an antecedent arising from movement, and *zeros*, words or morphemes which are not phonologically overt but nonetheless contain the same full range of linguistic information as other overt elements. Thus unlike in DAVRA, in KENUNIA a language has an inventory of zeros, each of which has a specific syntactic category and contributes specific semantic content to utterances in which it appears. A severe problem facing any theory of language acquisition is the need to explain how children can learn the inventory of non-overt elements and their linguistic features. Furthermore, one must also explain how children learn in the presence of movement. This clearly holds for uncontroversial forms of movement such as Wh-movement. It is exacerbated by the current trend in linguistics to postulate radical forms of movement and numerous non-overt elements. VP-internal subjects and V-to-I movement are two examples of such radical forms of movement, while the Larson/Pesetsky analysis of the ditransitive is an example that requires the child to learn non-overt prepositions that bear specific lexical features. While KENUNIA cannot currently handle all such phenomena, the long-term objective is to tackle this problem head on and develop a theory that can explain language learning in the presence of movement and non-overt elements.

- DAVRA, like MAIMRA, represents word and utterance meanings using Jackendovian conceptual structures. The semantic theory used by MAIMRA and DAVRA relates the meaning of an utterance to the meanings of its constituent words via a linking rule based on substitution. Part II of this thesis will discuss many of the shortcomings of both the Jackendovian representation and its associated linking rule. Basing a theory of language acquisition on such a questionable semantic theory renders the language acquisition theory suspect. The ultimate goal of this research is to develop a comprehensive theory of language acquisition using the semantic representation to be discussed in part II of this thesis as its basis. Since that representation is not yet fully formulated, KENUNIA adopts a temporary stopgap measure. It uses θ-theory as its semantic representation. The rationale behind this move is simple. Basing the theory of language acquisition on the weakest possible, least controversial, semantic theory can yield a more robust theory of language acquisition. The fewer assumptions one makes about the semantic theory, the less likely the possibility that the theory need be retracted as a result of falsifying some semantic assumption.

MAIMRA and DAVRA represented word and utterance meanings as conceptual structure fragments. The meaning of *John* might be $\mathbf{person}_1$, while the meaning of *walked* might be $\mathrm{GO}(x, [_{\mathrm{Path}}\ ])$. The linking rule would combine these two fragments to yield $\mathrm{GO}(\mathbf{person}_1, [_{\mathrm{Path}}\ ])$ as the meaning of *John walked*. KENUNIA instead represents word meanings via two components: a *referent* and a *θ-grid*. The referent of a word is simply a token denoting the object to which that word refers. For example, the referent of the word *John* might be $\mathbf{person}_1$ while the referent of the word *cup* might be $\mathbf{object}_1$. Words such as *the*, *walk*, and *slide* which do not refer to anything are assigned $\perp$ as their referent.

A θ-grid denotes the argument taking properties of a word. Conceptually, a word assigns a distinct θ-*role* to each of its arguments. The θ-grid specifies which θ-role is assigned to which argument. Formally, a θ-grid consists of a set of *θ-assignments*, each θ-assignment being a θ-role paired with a *complement index*, an integer denoting the argument to which that θ-role is to be assigned. Words such as *the*, *John*, and *cup* which do not take any arguments would have an empty θ-grid. An intransitive verb such as *walk* would have $\{\mathrm{THEME} : 1\}$ as its θ-grid. This indicates that *walk* assigns one θ-role, namely

THEME, to its external argument. More formally, the notation THEME : 1 specifies that the complement of the bar-level "1" projection of the terminal node associated with the word *walk* is assigned the θ-role THEME. Likewise, the θ-grid for a transitive verb such as *slide* might be {THEME : 0, AGENT : 1} indicating that the θ-role THEME is assigned to the internal argument while AGENT is assigned to the external argument. An internal argument is the complement of a bar-level "0" projection while an external argument is the complement of a bar-level "1" projection. Using complement indices to denote argument positions, instead of the terms 'internal' and 'external', keeps θ-theory independent of the decision as to the number of bar-levels used by $\overline{\mathrm{X}}$ theory.

The referent and θ-grid components of a word are orthogonal. A given word may have just a referent, just a θ-grid, both, or neither. Typically however, all words other than nouns will have $\perp$ as their referent, and only verbs and prepositions will have non-empty θ-grids.

KENUNIA represents utterance meanings via a *θ-map* that is itself a set of *θ-mappings*. A θ-mapping is similar to a θ-assignment except that a referent replaces the complement index. Thus the meaning of *John walked* would be represented in KENUNIA as the θ-map {THEME : $\mathbf{person}_1$}. This θ-map is derived from the θ-grid for *walked* and the referent of *John* by a process called *θ-marking*. Intuitively, the θ-marking rule combines {THEME : 1}, the θ-grid for *walked*, with $\mathbf{person}_1$, the referent for *John* to form the θ-map {THEME : $\mathbf{person}_1$} for *John walked*. A more formal specification of this process will be given later. In KENUNIA, θ-marking plays the role previously played by the linking rule used in MAIMRA and DAVRA. Thus in KENUNIA, the corpus consists of utterances paired with a θ-map instead of a set of meaning expressions. Furthermore, the lexicon maps words to their referents and θ-grids instead of meaning expression fragments.

Figure 4.8 illustrates a corpus that has been presented as input to KENUNIA. This corpus contains the same nine utterances that were presented to MAIMRA and DAVRA except that θ-maps replace the meaning expressions as the non-linguistic input paired with each input utterance. Each utterance in the corpus is paired with a single θ-map. Like the corpora presented to both MAIMRA and DAVRA, this corpus also exhibits referential uncertainty. The mechanism used by KENUNIA to represent referential uncertainty differs from that used by MAIMRA and DAVRA. In MAIMRA and DAVRA, each utterance was paired with a set of meaning expressions, only one of which constituted the actual meaning. The same uncertainty mechanism could have been incorporated into KENUNIA. This would have entailed pairing each utterance with a set of θ-maps, only one of which corresponded to the θ-map generated by θ-theory for the utterance. KENUNIA however, supports uncertainty in pairing linguistic with non-linguistic input by an even more general mechanism. KENUNIA requires only that the actual θ-map produced by applying θ-theory to the input utterance be a subset of the θ-map given as the non-linguistic input paired with that utterance. The referential uncertainty implied by a set of distinct θ-maps can be emulated by this more general mechanism by simply forming a single θ-map that is the union of the individual distinct θ-maps.

4.3.2 Linguistic Theory Incorporated in Kenunia

The linguistic theory incorporated into KENUNIA can be specified more precisely via the following principles.

1. $\overline{\mathrm{X}}$ theory

   tree structure: The linguistic input to KENUNIA consists of a sequence of utterances, each utterance being a string of words. KENUNIA associates a set of *nodes* with each utterance. Nodes are organized in a parent-child relationship. Each node except for one has a distinguished node called its *parent*. The one node without a parent is called the *root*. Each node also has a (possibly empty) ordered set of nodes called its *children*. A node with no children is called *terminal*. Every node is associated with a (possibly empty) substring of the input utterance.

$\{\text{AGENT} : \mathbf{person}_1, \text{THEME} : \mathbf{person}_1\}$ *John rolled.*
$\{\text{AGENT} : \mathbf{person}_2, \text{THEME} : \mathbf{person}_2\}$ *Mary rolled.*
$\{\text{AGENT} : \mathbf{person}_3, \text{THEME} : \mathbf{person}_3\}$ *Bill rolled.*
$\{\text{THEME} : \mathbf{object}_1\}$ *The cup rolled.*
$\{\text{AGENT} : \mathbf{person}_3, \text{THEME} : \mathbf{person}_3, \text{GOAL} : \mathbf{person}_2\}$ *Bill ran to Mary.*
$\{\text{AGENT} : \mathbf{person}_3, \text{THEME} : \mathbf{person}_3, \text{SOURCE} : \mathbf{person}_1\}$ *Bill ran from John.*
$\{\text{AGENT} : \mathbf{person}_3, \text{THEME} : \mathbf{person}_3, \text{GOAL} : \mathbf{object}_1\}$ *Bill ran to the cup.*
$\{\text{THEME} : \mathbf{object}_1, \text{SOURCE} : \mathbf{person}_1, \text{GOAL} : \mathbf{person}_2\}$ *The cup slid from John to Mary.*
$\{\text{AGENT} : \mathbf{person}_1, \text{PATIENT} : \mathbf{person}_1, \text{GOAL} : \mathbf{person}_2\}$ *John faced Mary.*

Figure 4.8: A sample corpus presented to KENUNIA.

Nodes associated with empty substrings are called *empty*.[12] The substrings associated with nodes obey the following two constraints. First, the substring associated with a non-terminal node must equal the concatenation of the substrings of its children, taken in order. Second, the substring associated with the root must equal the input utterance.

binary branching: Each node has at most two children.

categories: Each node is labeled with a *category*, which is one of the symbols N, V, P, D, I, or C. KENUNIA is written so that the set of possible categories is a parameter of the linguistic theory. Currently, the value of this parameter is given as input to KENUNIA—it is not acquired. Future work may explore the feasibility of acquiring the set of possible category labels, i.e. treating category labels as integers and trying sets of ever increasing cardinality until one is found that is consistent with the input.

bar-level: Each node is labeled with a *bar-level*, an integer between 0 and M. A node labeled with bar-level 0 is called a *minimal* node, while a node labeled with bar-level M is called a *maximal* node. Here again, KENUNIA is written so that M is a parameter of the linguistic theory. Currently however, the value of M is fixed at 2 and not acquired. As for categories, future work may explore the feasibility of acquiring the value for M, instead of taking it as a fixed input value.

head-complement and adjunction structures: Each node is either a head-complement structure or an adjunction structure. In head-complement structures, one distinguished child is designated the *head* while the remaining children (if any) are the *complements* of the head.[13]

[12] Empty terminal nodes are typically called *empty categories* in linguistic parlance. This introduces an ambiguity in the term *category*, sometimes referring to a label for a node, for instance N, V, or P, and sometimes referring to a node bearing a particular label. In this formulation, I use the distinct terms *category* and *node* for these two different uses and thus what are typically called empty categories are here referred to as empty (terminal) nodes.

[13] The terminology used here differs somewhat from current linguistic parlance. According to my use of the term *head*, the head of an X^2 node is its X^1 child, while in standard usage it is the X^0 child of the X^1 node. Furthermore, I use

The category of a head-complement structure must be the same as the category of its head, while the bar-level of a head-complement structure must be one greater than the bar-level of its head. For an adjunction structure, one distinguished child is designated the *head* while the remaining children are the *adjuncts* of the head. An adjunction structure must have at least one adjunct. Both the category and bar-level of an adjunction structure must be the same as its head. Complements must be maximal nodes, while adjuncts must be either minimal or maximal nodes. This principle, combined with the principle of binary branching, implies that all non-terminal nodes have one of the following five configurations.

phrase order parameters: For each category X, and each $0 \leq i < M$, the language sets the binary-valued parameter [X^i initial/final] to either **initial** or **final.** In languages which set [X^i initial], a head labeled X^i must be the first child of a head-complement structure, while in languages which set [X^i final], it must be the last child. Furthermore, for each category X and each $0 \leq i \leq M$ the language sets the binary-valued parameter [adjoin X^i left/right] to either **left** or **right**. In languages which set [adjoin X^i right], a head labeled X^i must be the first child of an adjunction structure, while in languages which set [adjoin X^i left], it must be the last child.[14] Note that head-complement and adjunction order are specified on a per category and per bar-level basis.[15]

C-selection: Any language specifies a finite set $\mathcal{C}$ of pairs of the form $\langle X, Y\rangle$ where X and Y are categories. If $\langle X, Y\rangle \in \mathcal{C}$ we say that X c-selects Y. If X c-selects Y then two restrictions apply. First, any node labeled X^0 must have a single complement labeled Y. This restriction is called *c-selection*. Second, any node labeled Y^M must be the complement of a node labeled X^0. This restriction is called *inverse c-selection*. KENUNIA is written so that the set $\mathcal{C}$ of c-selection relations is a parameter of the linguistic theory. Currently, the value of this parameter is given as input to KENUNIA—it is not acquired. All of the work described in this chapter assumes a specific set $\mathcal{C}$ of c-selections, namely that D c-selects N,[16] I c-selects V, and C c-selects I. Future work may explore more basic principles which govern the acquisition of $\mathcal{C}$.

terminals: Terminals must be either minimal or maximal nodes.

roots: Root nodes must be maximal.

2. Move-α
KENUNIA does not construct an explicit D-structure representation and thus does not represent movement as a correspondence between such a representation and S-structure. Instead, KENUNIA operates in a fashion similar to Fong's (1991) parser and constructs only an S-structure representation that is annotated with co-indexing relations between antecedents and their bound traces. KENUNIA associates a set $\mathcal{M}$ of movement relations with the set of nodes constructed for each input utterance. Each movement relation is an ordered pair of nodes. If $\mathcal{M}$ contains the pair $\langle\alpha, \beta\rangle$,

a generalization of the term complement to refer to the siblings of heads bearing any bar-level. Standard usage applies the term complement only to siblings of X^0 heads, and instead applies the term SPEC to siblings of X^1 heads. My non-standard use of terminology affords greater uniformity in stating the theory described here.

[14] Note that this formulation of parameter settings is independent of the binary branching principle.

[15] With six categories and $M = 2$ there are nominally 30 binary-valued parameters. Additional principles and restrictions reduce this to 16 non-degenerate parameters.

[16] This is in accord with the DP hypothesis.

we say that β is the *antecedent* of α and that α is *bound* by β.[17] Movement relations are subject to the following constraints.

(a) Bound nodes must be empty terminals. Bound empty terminals are called *traces*.

(b) Nodes can bind only one trace.[18]

(c) Traces must have only one antecedent.

(d) Antecedents must be either minimal or maximal nodes. This means that only minimal and maximal nodes move.

(e) The head of an adjunction structure cannot be a trace. This means that a base generated adjunction structure cannot move without its adjuncts.

(f) The head of an adjunction structure cannot be an antecedent. This means that no node can adjoin to a moved node.

(g) Nodes cannot bind themselves. This is part of what is known as the i-within-i constraint.

(h) Antecedents must have the same category and bar-level as their bound traces.

(i) Antecedents must m-command their bound traces. This is a variant of ECP, the empty category principle.

(j) Antecedents and their bound traces cannot be siblings.

(k) Antecedents must not be θ-marked. The concept of θ-marking will be defined below. This means that a node cannot move to a θ-marked (argument) position.

3. θ-theory

KENUNIA incorporates the following variant of θ-theory. As discussed previously, each word has an associated referent and θ-grid. A θ-marking rule is used to construct a θ-map corresponding to an entire utterance from the referents and θ-grids associated with its constituent words. More precisely, a lexicon maps (possibly empty) strings of words to their associated referent and θ-grid. Each terminal is associated with some (possibly empty) substring of the input utterance. Every terminal, except for traces, is assigned both a referent and a θ-grid, in addition to a category and bar-level. This includes both overt as well as empty terminals. The referent and θ-grid for a terminal is taken from the lexical entry for the (possibly empty) substring of words associated with that terminal.

Intuitively, the θ-marking rule combines a θ-assignment such as THEME : 1, with a referent such as $\mathbf{person}_1$ to form the θ-mapping THEME : $\mathbf{person}_1$. The θ-map for an utterance will contain a number of such θ-mappings, one for each θ-assignment in the θ-grid of each word in the utterance. A word or node with a non-empty θ-grid is called a *θ-assigner*. θ-theory stipulates that each θ-assigner must *discharge* its θ-grid. Discharging a θ-grid involves discharging each of its constituent θ-assignments. Discharging a θ-assignment (i.e. assigning a θ-role) is done by *θ-marking* the appropriate complement of the θ-assigner involved. This involves pairing the referent of a particular word in that complement with the θ-role specified by that θ-assignment and adding the resulting θ-mapping to the θ-map for the utterance. The complement of the θ-assigner thus θ-marked is called a *θ-recipient*. θ-recipients are said to *receive* the given θ-role.

The θ-marking rule incorporates the following constraints.

[17] I use the terms *antecedent* and *bound* here is a much more restricted way than is common in the linguistic literature. KENUNIA does not incorporate any binding theory. The terms are used solely to denote the relation between a moved node and a trace created by that movement.

[18] KENUNIA currently supports only one type of trace. KENUNIA does not currently support parasitic gaps, PRO, pure variables, or operator-variable structures.

(a) θ-marking is performed at D-structure. This standard assumption has two implications. First, θ-assigners which have moved must discharge their θ-grids from their position at D-structure.[19] In other words, antecedents don't discharge their θ-grids in situ. Instead they discharge their θ-grids from the location of their bound traces. Second, since θ-recipients receive their θ-role in their D-structure position, traces which are θ-marked pass on that θ-role to their antecedent.

(b) θ-assigners must discharge their θ-grids. In other words, if a node assigns a θ-role to its internal argument, for example, then there must be an internal argument to receive that θ-role.

(c) Complements of nodes labeled with non-functional categories must be θ-marked. This constraint is commonly called the *θ-criterion*. In KENUNIA, functional categories are taken to be those which c-select, namely D, I, and C.

(d) The θ-map constructed for an utterance must contain at least one θ-mapping.

The θ-marking rule can be stated more formally as follows. The *ultimate antecedent* of a node α is

- α itself if α is not a trace or
- the ultimate antecedent of the antecedent of α if α is a trace.

The *ultimate referent* of a node α is

- the ultimate referent of the antecedent of α if α is a trace,
- the ultimate referent of the complement of α if α is a head-complement structure and the category of α is a c-selecting category,
- the ultimate referent of the head of α if α is either an adjunction structure or a head-complement structure where the category of α is not a c-selecting category, or
- the referent of α if α is a terminal and not a trace.

Every non-antecedent node α whose ultimate antecedent is a terminal must discharge the θ-grid associated with that ultimate antecedent. If the θ-grid for the ultimate antecedent of α contains the θ-assignment $\rho : i$ then find the node β such that

- β dominates α,
- the bar-level of β is i,
- β is not the head of an adjunction structure, and
- no node which dominates α and is dominated by β is a complement or adjunct,

and form the θ-mapping $\rho : \mu$ where μ is the ultimate referent of the complement of β.

4. Case theory
KENUNIA incorporates a variant of the case filter which states that overt maximal D nodes can only appear in one of three places: the complement of an I^1 node, the complement of a P^0 node and the complement of a V^0 node if the V^0 node assigns a θ-role to its external argument. This latter restriction is a formulation of Burzio's generalization. The above formulation of the case filter assumes that $M = 2$.

[19] As stated previously, KENUNIA does not create an explicit D-structure representation. KENUNIA's implementation of θ-marking, however, operates as if such a representation existed by utilizing movement relations in the S-structure representation to guide the θ-marking process.

5. Monosemy constraint
   A *lexicon* maps word strings to a unique category, bar-level, referent, and θ-grid. The category, bar-level, referent, and θ-grid of terminal nodes (except for traces) must be the those projected by the lexicon for the substring associated with the terminal node.

4.3.3 Search Strategy

KENUNIA uses a variant of the weaker, revised search strategy used by DAVRA. In this strategy, all language-specific knowledge is maintained as part of a single *language model*. This language model contains information both about the lexicon as well as syntactic parameter settings. The language model consists of a set of propositions. There are six types of proposition, illustrated by the following examples.

1. category(*slide*) = V
2. bar-level(*slide*) = 0
3. referent(*slide*) = $\perp$
4. θ-grid(*slide*) = {THEME : 1}
5. [I^0 initial]
6. [adjoin I^0 left]

The first four propositions indicate components of the lexical entry for the word *slide*. Note that the category, bar-level, referent, and θ-grid for a word are represented as four independent propositions in the language model. The last two propositions indicate parameter settings; in this case the statement that the language is head-initial for inflection nodes and that adjuncts adjoin to the left of I^0 nodes.

At all times, KENUNIA maintains a single set of such propositions that represent the current cumulative hypothesis about the language being learned. The eventual goal is for the initial language model to consist of the empty set of propositions and to have KENUNIA acquire all six types of propositions representing both parameter settings and the syntactic and semantic properties of words. The current implementation, however, learns only parameter settings and syntactic categories. Thus, KENUNIA is provided with an initial language model containing propositions detailing the referents and θ-grids for all words, both overt and empty, that appear in the corpus. KENUNIA then extends this language model with propositions detailing the categories and bar-levels of those words, as well as the syntactic parameter settings.

KENUNIA extends the language model by processing the corpus on an utterance by utterance basis. Each utterance is processed and then discarded. No information, except for the cumulative language model, is retained after processing an input utterance, other than a set of nogoods to be described shortly. When processing an input utterance, KENUNIA simply tries to find a superset of the current language model that allows the input utterance to be parsed. This superset must be consistent in that it cannot assign a parameter two different settings, nor can it assign a word two different categories, bar-levels, referents, or θ-grids. This latter restriction is an embodiment of the monosemy constraint. If KENUNIA is successful in finding a consistent superset of the language model capable of parsing the input utterance, this superset is adopted as the new language model, and processing continues with the next utterance in the corpus.

So far, the strategy employed by KENUNIA is identical to the revised strategy used by DAVRA. The strategies diverge however, when KENUNIA is unable to find a consistent extension of the language model. In this situation, DAVRA would compute a minimal nogood. A nogood is a subset of the current language model that is inconsistent, i.e. one that cannot be extended to parse the current input. A nogood is

minimal if it has no proper subset which is a nogood. It turns out that the process used by DAVRA for computing a minimal nogood is intractable. DAVRA repeatedly tries to remove individual propositions from the nogood, one by one, testing the resulting set for consistency. Although only a linear number of such consistency tests are performed, they are performed on successively smaller sets of propositions. The smaller the language model, the more freedom the parser has in making choices to try to extend that language model to see if it is consistent with the current input. Experience has shown that a parser can work efficiently with either an empty language model, or one which is almost fully specified. In the former case, an empty language model places little restriction on finding a consistent extension and thus one will almost always be found. In the latter case, a highly constrained language model will focus the search and yield very few intermediate analyses. A small but non-empty language model, however, produces a larger number of analyses that must be checked for consistency. For this reason, the strategy used by DAVRA for computing minimal nogoods turns out to be intractable in practice. Therefore, KENUNIA uses nogoods that are not necessarily minimal. When the current language model cannot be extended to parse the input utterance, KENUNIA forms a nogood that contains the following propositions.

- all of the syntactic parameters
- all category and bar-level propositions for words appearing in the current input utterance
- all category and bar-level propositions for zeros in the current language model

This nogood, while not minimal, is nonetheless a subset of the current language model and is easy to compute.

KENUNIA uses nogoods thus constructed in two ways. First, the nogood is saved to prevent repeatedly hypothesizing the same language model. Whenever KENUNIA attempts to extend a language model, the extension is checked to ensure that it is not a superset of some previously constructed nogood. Extensions that are supersets of some nogood are discarded since they are inconsistent with prior input. Note that KENUNIA does not retain the prior input itself to perform this check of consistency. Only the nogood, the inconsistent language model, is retained to prevent looping. Second, one proposition is selected arbitrarily from the newly constructed nogood. This proposition is removed from the current language model and a new attempt is made to extend the resulting language model to parse the current input utterance.

4.3.4 The Parser

A key step of the above learning strategy is determining whether the current language model is consistent with the next input utterance. This requires determining whether the language model, either as it stands, or possibly extended, can parse the input utterance. KENUNIA uses a parser whose architecture is similar to that described by Fong. The parser consists of a cascade of modules. The first module generates potential S-structure representations corresponding to the input utterance. Each subsequent module can either filter out structures which violate some principle, or can adorn a structure with additional information such as θ-markings or movement relations. Since such augmentation of structure can be nondeterministic, the number of structures passed from module to module can both grow as a result of structure augmentation, and shrink as a result of filtering. The particular cascade of modules used in KENUNIA is illustrated in figure 4.9. Note that $\overline{\mathrm{X}}$ theory must come first since it is the initial generator. θ-theory must come after Move-α since θ-marking is performed at D-structure. θ-theory uses the movement relations produced by Move-α to reconstruct the D-structure representation. The case filter depends on Burzio's generalization which requires determining the θ-grid of a head. Since a head trace inherits its θ-grid from its antecedent, the case filter must come after Move-α as well. Since the

Figure 4.9: The cascade of modules used in KENUNIA's parser.

case filter only rejects structures and doesn't nondeterministically adorn them, it is more efficient to place it before θ-theory. Thus the cascade order for the parsing modules is fixed.

The variant of $\overline{\mathrm{X}}$ theory incorporated into KENUNIA generates infinitely many potential $\overline{\mathrm{X}}$ structures corresponding to any given input string. This is because such $\overline{\mathrm{X}}$ structures can repeatedly cascade empty terminals. KENUNIA might therefore never terminate trying to parse an utterance which could not be parsed with a given language model. Solving this problem in general requires induction. Lacking the ability to inductively prove that no element of such an infinite set of S-structures meets the subsequent constraints, or some meta-level knowledge which would bound the size of the S-structure representation by a known function of the length of the underlying utterance, KENUNIA instead sets a limit k on the number of empty terminals that can be included in a generated S-structure. This single limit k applies collectively to both traces and zeros. The implementation allows the limit k to be adjusted. Preliminary experimentation with different values for k indicate that performance degrades severely when $k > 3$. All results reported in this chapter, therefore assume that $k = 3$. KENUNIA uses an iterative deepening strategy when searching for S-structures which meet the constraints, first enumerating those structures which do not contain any empty terminals, then those which contain one empty terminal, and so forth, terminating after enumerating structures which contain k empty terminals. Thus while several alternate analyses for an utterance may meet the constraints imposed by the linguistic theory, KENUNIA always adopts the analysis with the minimal number of empty terminals. It is this analysis which contributes the necessary extensions to the language model in the search process described previously. There may however, be several alternate minimal analyses. In this case, an arbitrary one is chosen to extend the language model.

The $\overline{\mathrm{X}}$ theory module operates essentially as a context-free parser. KENUNIA generates a context-free grammar corresponding to an instantiation of the aforementioned variant of $\overline{\mathrm{X}}$ theory with the parameter settings in the current language model. For example, the grammar would contain the rule

$$\mathrm{D}^1 \rightarrow \mathrm{D}^0\ \mathrm{N}^M$$

if the language model contained the parameter [D^0 initial].[20] Alternatively, it would contain the rule

$$\mathrm{D}^1 \rightarrow \mathrm{N}^M\ \mathrm{D}^0$$

if the language model contained the parameter [D^0 final]. Given such a context-free grammar, the $\overline{\mathrm{X}}$ theory module uses a variant of the CKY algorithm to generate S-structures. The particular memoization strategy used allows each variant structure to be retrieved in constant time once the well-formed substring table has been constructed in $O(n^3)$ time.

One feature which distinguishes this parser from the parser described by Fong is that it can operate with an incomplete language model. The learning algorithm in which it is embedded must determine whether a given language model can be extended to parse a given utterance, and if so, what the necessary extension is. If, for example, the language model does not set either the [D^0 initial] or the [D^0 final] parameter, then the grammar can simply contain both of the above rules. Since however, any given language must set the parameter one way or the other, a hypothetical analysis for an utterance could never be correct if one subphrase was generated by one setting, and another by the opposite setting.

[20] KENUNIA doesn't actually generate a context-free grammar; rather the parser directly uses the parameter settings. The operation of the parser is most easily explained, however, as if it utilized an intermediate grammar.

This requires that the $\overline{\mathrm{X}}$ theory module check the output of the CKY parser to guarantee that each structure produced is generated by a consistent set of parameter settings. The necessary extensions to the language model can be recovered by examining a structure output by the final module in the cascade. The language model might also contain incomplete word-to-category and word-to-bar-level mappings. These are handled by treating such words as lexically ambiguous in the CKY algorithm. Here again, since KENUNIA must ultimately enforce the monosemy constraint, a hypothetical analysis for an utterance could never be correct if some word appeared more than once in that utterance with different category or bar-level assignments. The $\overline{\mathrm{X}}$ theory module must check the structures produced for such inconsistencies as well.

The cascaded parser architecture has the property that the $\overline{\mathrm{X}}$ theory module produces numerous structures that are ultimately filtered out by later modules in the cascade. Since asymptotically, the processing time is proportional to the number of intermediate structures generated, it is useful to fold as much of the constraint imposed by the later modules into the CKY-based structure generator. There is a limit to how much can be done along these lines however. Much of the constraint offered by the latter modules depends on non-local structural information. By its very nature, a context-free parser can enforce only local constraints on the structures it generates. There are however, two components of θ-theory which are essentially local and thus can be folded into the context-free structure generator. These are the θ-criterion and the requirement that all nodes discharge their θ-grid. Coupled with the c-selection requirements, these two components can be reformulated as the following pair of constraints. A node X^i must have a complement if both $i = 0$ and X c-selects, or if the θ-grid of the ultimate head of the node contains a θ-assignment with complement index i. Likewise, a node X^i must not have a complement if the θ-grid of the ultimate head of the node does not contain a θ-assignment with complement index i and X does not c-select. These constraints can be encoded by adding features $\pm\theta_i$ to the categories X^i in the context-free grammar. For example, the grammar would contain the rules

$$\begin{array}{l} \mathrm{V}^1[+\theta_0] \rightarrow \mathrm{V}^0[+\theta_0]\ \mathrm{D}^M \\ \mathrm{V}^1[-\theta_0] \rightarrow \mathrm{V}^0[-\theta_0] \end{array}$$

but not the rules

$$\begin{array}{l} \mathrm{V}^1[-\theta_0] \rightarrow \mathrm{V}^0[-\theta_0]\ \mathrm{D}^M \\ \mathrm{V}^1[+\theta_0] \rightarrow \mathrm{V}^0[+\theta_0]. \end{array}$$

Ground context-free rules can be generated by enumerating instances of such rule schemas, for all possible unspecified feature assignments, subject to the constraint that the feature assignments of a node must match those of its head.

So far, only the above constraints have been folded into the context-free CKY-based structure generator. There would be substantial algorithmic benefit if all of the remaining modules could be folded in as well. If this could be accomplished then there would never be any need to enumerate the structures generated by the context-free grammar, since the parser as a whole is used only as a recognizer, to determine whether an utterance is consistent with a given language model. Such recognition could be performed in polynomial time, irrespective of whether the language model was complete or incomplete, notwithstanding the need for consistency checks on the generated structures as discussed previously. This would allow efficient computation of minimal nogoods since with a CKY-based recognizer there would be no performance penalty for smaller language models over larger ones. Even if some per-structure filtering was required, as is the case for consistency checks, folding more into the generator, enabling it to producing fewer structures which violate subsequent filters, makes the process of computing smaller nogoods more feasible.

4.3.5 Additional Restrictions

Even after folding parts of θ-theory into the context-free S-structure generator, the resulting generator can still produce a large number of intermediate structures. A manageable number of structures is produced when the language model is complete. In such cases, the linguistic theory overgenerates only slightly, with subsequent modules filtering out practically all of the structures generated. Smaller language models however, generate an astronomical number of intermediate structures. While the linguistic theory may, in principle, be able to filter out all such intermediate structures, it has never succeeded in doing so in practice. Thus for pragmatic reasons, some additional restrictions are adopted that further constrain the structures generated. Both $\overline{\mathrm{X}}$ theory and Move-α are restricted. Most of the restrictions on $\overline{\mathrm{X}}$ theory apply to adjunction. These include the following.

1. The bar-level of the head of an adjunction structure must be the same as the bar-level of its adjunct. In other words, a node can adjoin only to a node of the same bar-level.
2. Minimal nodes that are the head of an adjunction structure must bear the category label I. In other words, the only minimal node that can be adjoined to is I^0.
3. Minimal adjunct nodes must bear the category label V. In other words, the only minimal node that can be an adjunct is V^0.
4. Maximal nodes that are the head of an adjunction structure must be labeled either N or V. In other words, the only maximal nodes that can be adjoined to are NP and VP.
5. Maximal adjunct nodes must be labeled either P or C. In other words, the only maximal nodes that can be adjuncts are PP and CP.

Two further restrictions apply to $\overline{\mathrm{X}}$ theory that do not relate to adjunction.

1. Complements of nodes bearing bar-level 1 must bear the category label D. In other words, specifiers must be DPs.
2. The root must bear the category label C. In other words, utterances must be CPs and not other maximal nodes such as DPs or PPs.
3. Terminals must be either empty or singleton word strings. KENUNIA cannot currently handle idioms, or terminals that correspond to more that one word.

All of these restrictions are folded into the context-free grammar used by the $\overline{\mathrm{X}}$ structure generator. With these restrictions, the number of intermediate structures generated is far more manageable. Additionally, several restrictions are imposed on Move-α.

1. Minimal antecedents must bear the category label V. In other words, the only minimal node that can move is V^0.
2. Maximal antecedents must not bear a c-selected category label. In other words, c-selected nodes such as NP, VP, and IP don't move.

Fong's parser implicitly adopts these very same restrictions, with the exception that adjunction to IP is allowed.[21] None of these restrictions seem very principled. Furthermore, some of them appear to be downright wrong. They were chosen since they are the tightest such restrictions which still allow the corpus in figure 4.8 to be parsed. The need for these restrictions is a severe weak link in the current theory. Incorporating these restrictions was dictated by pragmatic expedience, the advantage of getting the system to work at all, before getting it to work cleanly. Replacing these *ad hoc* restrictions with more principled ones remains a prime area for future work.

[21] These restrictions only hold for that portion of Fong's parser which is comparable to KENUNIA. In Fong's parser these restrictions do hold of LF movement, adjectives, adverbs, and I-lowering.

cup:	$\mathbf{object}_1\{\}$
-ed:	$\perp\{\}$
john:	$\mathbf{person}_1\{\}$
slide:	$\perp\{\text{THEME} : 1\}$
that:	$\perp\{\}$
$\emptyset$:	$\perp\{\}$
face:	$\perp\{\text{PATIENT} : 1, \text{GOAL} : 0\}$
from:	$\perp\{\text{SOURCE} : 0\}$
bill:	$\mathbf{person}_3\{\}$
the:	$\perp\{\}$
mary:	$\mathbf{person}_2\{\}$
to:	$\perp\{\text{GOAL} : 0\}$
run:	$\perp\{\text{THEME} : 1\}$
roll:	$\perp\{\text{THEME} : 1\}$

Figure 4.10: KENUNIA is given these mappings from words and zeros to their referents and θ-grids as prior language-specific knowledge before processing the corpus from figure 4.8.

4.3.6 Kenunia in Operation

Appendix B illustrates KENUNIA's application of the above strategy in processing the corpus from figure 4.8. For this run, KENUNIA was also given an initial lexicon mapping the words in the corpus, as well as the inventory of zeros, to their referents and θ-grids. This initial lexicon is illustrated in figure 4.10. The initial lexicon did not include any category or bar-level information, nor was KENUNIA given any syntactic parameter settings.

Like the revised DAVRA strategy, KENUNIA processes a corpus repeatedly to make up for the lack of a larger corpus. KENUNIA makes two passes over the corpus from figure 4.8 before converging on a language model that survives the third pass without need for revision.

This process can be summarized as follows.[22] Starting with an empty language model, KENUNIA succeeds in processing the utterance *John rolled* forming the incorrect though nonetheless valid structure illustrated on page 235 in appendix B. In doing so, KENUNIA assumes that *John* is a DP, *roll* is an I^0, the *-ed* morpheme is a VP, and the zero lexeme is a C^0. KENUNIA also assumes that the language is I^0 initial, I^1 final, and C^0 final. KENUNIA continues processing further input utterances through page 242, successfully extending the language model for each utterance. Though many of the assumptions made are incorrect, they are consistent with both the linguistic theory and the portion of the corpus seen so far. When processing the utterance *John faced Mary*, however, KENUNIA is not able to find a consistent extension of the language model capable of parsing this utterance. This is illustrated on page 243. At this point no single proposition can be retracted from the language model to make it consistent with the current utterance. It is possible however, to derive a consistent language model by retracting both the assumption that the category of the *-ed* morpheme is V, as well as the assumption that its bar-level is 2. After retracting these assumptions, KENUNIA is able to process this utterance by assuming that *-ed* is an I^0. This analysis is illustrated on page 244. Note that in order to make this analysis, KENUNIA had to posit a structure that included both V-to-I movement as well as subject raising from SPEC of V to SPEC of I. Analysis of previous input did not include such movement. There is nothing magical about this transition. KENUNIA did not discover the concept of movement. The potential for movement

[22]Note that in appendix B, the symbol $\emptyset$ denotes a zero, t denotes a trace, X denotes an undetermined category, and n denotes an undetermined bar-level.

was latent all the time in the linguistic theory with which she was innately endowed. She simply did not have to invoke that potential until the current utterance, for simpler analyses (i.e. ones with fewer empty terminals) sufficed to explain the prior utterances of the corpus.

After successfully processing the previous utterance with the revised language model, KENUNIA begins processing the corpus again, since the corpus has been exhausted. KENUNIA now encounters problems trying to process the utterance *John rolled* (page 245). This time however, a single retraction suffices to allow KENUNIA to continue. She retracts the assumption that *roll* is labeled I, replacing it with the assumption that it is labeled V (page 246). KENUNIA is then able to successfully parse a few more utterances until she encounters the utterance *Bill ran to Mary* on page 250. This requires her to retract the assumption that *run* is labeled I and decide instead that *run* is labeled V (page 251). After one more retraction, labeling *slide* as V instead of I on pages 254 and 255, KENUNIA is able to make one complete pass through the corpus without further revision, and thus converges on the lexicon and parameter settings illustrated in figure 4.11. This language model is consistent with both the corpus and the linguistic theory. Processing the corpus to produce this language model requires about an hour of elapsed time on a Sun SPARCstation 2^{TM} computer.

As with the revised version of DAVRA, the method described above can determine only that this is one possible consistent language model, not that it is the only such language model. These methods can be extended to determine whether the solution is unique by using the same techniques that were described for DAVRA. Furthermore, like the revised version of DAVRA, the rate of convergence of the search strategy used by KENUNIA is dependent on the order in which utterances are processed. Future work will attempt to quantify the sensitivity of the search strategy to corpus ordering.

From figure 4.11 one can see that KENUNIA has arrived at the correct category and bar-level assignments for all of the words in the corpus except *cup*. KENUNIA assigns *cup* the correct category N, but incorrectly assigns it bar-level 2 instead of 0. One can easily see that the linguistic theory incorporated into KENUNIA is not able to force a word to be labeled X^0 instead of X^2 without seeing that word appear with either a complement or specifier. Since *cup* has an empty θ-grid, it cannot take a complement or specifier, for that would violate the θ-criterion. Thus KENUNIA could never uniquely determine the bar-level of nouns like *cup*. This is a shortcoming of the KENUNIA linguistic theory for which I do not yet have a viable solution.

KENUNIA likewise makes a number of incorrect parameter setting decisions. She sets [V^0 final] and [C^0 final]. The former occurs because in the current corpus verbs always raise to adjoin to I.[23] There is thus no evidence in S-structure as to the original position of the verb. I do not yet have a viable solution to this problem. The latter occurs because the corpus does not contain any overt complementizers. With only zero complementizers, it is equally plausible to postulate that the zero complementizer follows an utterance as it is to postulate that it precedes the utterance. Encountering utterances with overt complementizers should remedy this problem.

KENUNIA is still very much work in progress. Three areas need further work. First, as mentioned before, a number of *ad hoc* restrictions were adopted as part of the linguistic theory to reduce the number of intermediate structures generated. KENUNIA does not work without such restrictions. A goal of prime importance is replacing those restrictions with ones which are more soundly motivated, or perhaps eliminating them entirely by using alternative parsing algorithms. Second, one of the original goals behind KENUNIA was to extend DAVRA to account for learning in the presence of movement and empty categories. This goal has been partially achieved since KENUNIA analyzes the corpus in figure 4.8 in terms of V-to-I movement and raising of VP-internal subjects to SPEC of IP. Nonetheless, this success is partially gratuitous since such movement is theory-internal. An immediate goal is to exhibit learning in the presence of less controversial forms of movement, such as Wh- and NP-movement. Doing so would be a major advance since no current theory can explain how children learn word meanings in the presence

[23] The methods suggested in Lightfoot (1991) work only when the corpus contains some utterances with verbs in their original position.

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Figure 4.11: The parameter settings and lexicon derived by KENUNIA for the corpus in figure 4.8. KENUNIA derived only the category and bar-level information in the lexicon. The referent and θ-grid information was given to KENUNIA as prior language-specific input.

of such movement. Wh-movement, in particular, is prevalent in parental input to children. Longer-term goals along these lines would be to explain the acquisition of numerous phenomena associated with the interaction between the verbal and inflectional systems. These include verb-second/verb-final phenomena, subject-aux-inversion, diathesis alternations (in particular the passive alternation), and the unergative/unaccusative distinction. Finally, KENUNIA does not yet achieve the level of performance that has been demonstrated with DAVRA. DAVRA learns three things—parameter settings, word-to-category mappings, and word-to-meaning mappings—with no such prior information. Furthermore, DAVRA does so for very small corpora in both English and Japanese. KENUNIA on the other hand, learns only two things: parameter settings and word-to-category mappings. KENUNIA must be given word-to-meaning mappings as prior language-specific input. Furthermore, KENUNIA has been demonstrated only on a very small English corpus. A goal of prime importance is to fully replicate the capabilities of DAVRA within the more comprehensive linguistic framework of KENUNIA.

Chapter 5

Conclusion

Part I of this thesis has addressed the question: *What procedure might children employ to learn their native language without any access to previously acquired language-specific knowledge?* I have advocated cross-situational learning as a general framework for answering this question. In chapter 3, I have demonstrated how cross-situational learning can be more powerful than trigger-based learning and can bootstrap from an empty language model. Furthermore, in chapter 4, I have demonstrated three implemented systems based on this framework that are capable of acquiring very small language fragments. MAIMRA learns both word-to-category and word-to-meaning mappings, for a very small fragment of English, given prior access only to grammar. DAVRA learns both word-to-category and word-to-meaning mappings, as well as the grammar, for very small fragments of both English and Japanese. KENUNIA learns word-to-category mappings along with the grammar, for a very small fragment of English, given prior access only to word-to-meaning mappings. Each of these systems learns from a corpus, containing positive-only examples, pairing linguistic information with a representation of its non-linguistic context. In MAIMRA and DAVRA, both word and utterance meanings are represented as Jackendovian conceptual structures. In KENUNIA, θ-theory replaces these conceptual structures as the framework for representing semantic information. All three systems are capable of learning despite referential uncertainty in the mapping of utterances to their associated meaning.

5.1 Related Work

A number of other researchers have attempted to give procedural accounts of how children might acquire language. These accounts differ from the one given here in a number of ways. Some advance trigger-based learning—unambiguously augmenting one's language model with information gleaned from isolated utterances—rather than the cross-situational approach presented here. Most explain only part of the acquisition process, for instance, acquiring word-to-meaning mappings but not word-to-category mappings and grammar, or vice versa, assuming that the learner possesses some prior language-specific knowledge. Furthermore, most do not deal with the problem of referential uncertainty. I will discuss some related work in detail below. Other important related work which I will not have the opportunity to discuss includes Granger (1977), Anderson (1981), Selfridge (1981), Berwick (1979, 1982, 1983), and Suppes et al. (1991).

5.1.1 Semantic Bootstrapping

Grimshaw (1979, 1981) and Pinker (1984) have proposed an approach which has been termed the *semantic bootstrapping hypothesis*. According to this approach, the child is assumed to first learn the

meanings of individual words by an unspecified prior process. Thus at the onset of semantic bootstrapping, a child can already map, for instance, *John* to **John**, *see* to SEE, and *Mary* to **Mary**. Furthermore, the semantic bootstrapping hypothesis assumes that the child's innate linguistic knowledge contains a universal default mapping between semantic concept classes and their syntactic realization. This knowledge includes, for instance, the fact that THINGS are realized as nouns and EVENTS are realized as verbs. Such language-universal default mappings are termed *canonical structure realizations*. Using such knowledge, the child can infer that *John* and *Mary* are nouns, and *saw* is a verb, from the observation that **John** and **Mary** are THINGS, and SEE is an EVENT. Furthermore, upon hearing an utterance such as *John saw Mary*, a child can infer that the language she is hearing admits utterances of the form noun-verb-noun.

Pinker (p. 38) illustrates the above strategy via the following example. For simplicity, suppose that universal grammar was described by the following grammar schema.

$$\begin{aligned} \mathrm{S} &\rightarrow \{\mathrm{NP}, \mathrm{VP}\} \\ \mathrm{NP} &\rightarrow \{(\mathrm{DET}), \mathrm{N}\} \\ \mathrm{VP} &\rightarrow \{\mathrm{NP}, \mathrm{V}\} \end{aligned}$$

This is a grammar schema in the sense that the order of the constituents in the right hand sides of the rules is not specified—the learner must figure out the correct order for the language being learned. Furthermore, suppose that the above grammar schema was innate. Upon hearing the utterance *The boy threw rocks*, the learner could form the following analysis

and in doing so determine incorrect constituent order parameters and word-to-category mappings for English. If however, the learner knew that *boy* and *rocks* were nouns, *threw* was a verb, and *the* was a determiner, presumably by applying canonical structure realization rules to their known meanings, she could determine that only the following structure is possible

allowing her to infer the correct constituent order parameters for English.

The above example works however, only with the oversimplified grammar schema. If one adopts a more comprehensive theory of universal grammar, the learner might not be able to uniquely determine the constituent order parameter settings, even given complete word-to-category mappings for every word

in the input. Take for instance, the linguistic theory which was described in section 4.3. Under this theory, the above utterance allows eight different analyses, where the three parameters [V^0 initial/final], [V^1 initial/final], and [C^0 initial/final] each vary independently. One such analysis is shown below.

Whether semantic bootstrapping is a viable acquisition theory is a question which must be asked independently for each linguistic theory proposed. The ability for semantic bootstrapping to uniquely constrain potential analyses and determine parameter settings decreases as the linguistic theory becomes richer and allows more variance between languages. Thus it is unclear whether semantic bootstrapping will explain acquisition under the correct linguistic theory, whenever that is discovered.

The semantic bootstrapping hypothesis makes two crucial assumptions. First, word meanings are acquired by an unspecified process prior to the acquisition of syntax. This implies that the process used to acquire word meanings, whatever it is, cannot make use of syntactic information, since such information is acquired only later. Furthermore, semantic bootstrapping is not a complete account of language acquisition, since it does not offer an explanation of how that prior task is accomplished. It explains only how language-specific syntax is acquired, not how word-to-meaning mappings are acquired. Second, semantic bootstrapping assumes that the learner uses a trigger-based strategy to acquire language-specific information from isolated situations. Only those situations that uniquely determine language-specific choices drive the language acquisition process. The above example was a failed attempt at showing how semantic bootstrapping made such situations more predominant, constraining otherwise ambiguous situations to be determinate. Furthermore, the assumption that word meanings are acquired prior to syntax was motivated specifically as a method for constraining ambiguous situations. This thesis suggests a different approach whereby the learner can acquire partial knowledge from ambiguous situations and combine such partial knowledge across situations to infer unique solutions that could not be determined from individual situations alone. This cross-situational approach thus also alleviates the need to assume prior knowledge, since all such knowledge can be acquired simultaneously by the same mechanism.

5.1.2 Syntactic Bootstrapping

In a series of papers (Gleitman 1990, Fisher et al. 1991), Gleitman and her colleagues have proposed an alternate learning strategy that has become known as *syntactic bootstrapping*. In contrast to semantic bootstrapping, where knowledge of word meanings guides the acquisition of syntax, syntactic bootstrapping assumes essentially the reverse, that knowledge of the syntactic structures within which words appear guides the search for possible meanings. This alternate strategy is best illustrated by the following example. Suppose a child heard the utterance *John threw the ball to Mary* in the context where she observed John throwing a ball to Mary. Furthermore, suppose that the child already knew that *John*, *ball*, and *Mary* were nouns meaning **John**, **ball**, and **Mary** respectively, that *to* was a preposition meaning TO(x), and that *the* was a determiner denoting a definite reference operator. In this circumstance, the child lacks only the category and meaning of *threw*. Finally, suppose that the child can form a parse tree for the utterance. Gleitman (1990) and Fisher et al. (1991) suggest that such a parse tree can be constructed using prosodic information available in parental speech to children.[1] In this situation, the child can infer that *throw* must mean 'throw' since that is the only meaning consistent with both the non-linguistic observation, as well as the utterance, given the partial information already known about the meaning and syntax of that utterance.

The key idea here is that the syntactic information in the utterance acts as a filter on potential word-to-meaning mappings for the unknown verb *threw*. At the time the utterance was heard, other things may have been happening or true in the world. John may have been wearing a red shirt and Mary could have been walking home from school. Either of these could be the meaning of some potential utterance in that situation. Thus *a priori*, a novel verb heard in this context could mean 'wear' or 'walk'. Yet the learner could infer that *threw* could not mean 'wear' or 'walk' since neither of these could consistently fit into the utterance template *John x the ball to Mary*, given both the known meanings of the remaining words in the utterance, as well as its structure.

As stated above, this strategy differs little from that proposed by Granger (1977) where the meaning of a single novel word can be determined from context. Gleitman however, takes the above strategy a step further. She claims that the structure of an utterance alone can narrow the possible word-to-meaning mappings for a verb in that utterance, even without knowledge of the meanings of the remaining words. Suppose that a child observed John pushing a cup off the table causing it to fall. In this situation, an utterance can potentially refer to either the pushing event or the falling event. She claims that a child hearing *John pushed the cup* would be able to infer that *pushed* refers to the pushing event and not the falling event since structurally, the utterance contains two noun phrases, and the argument structure of PUSH(x, y), but not FALL(x), is compatible with that structure. Similarly, a child hearing *The cup fell* could determine that *fell* refers to the falling event, and not the pushing event, since its syntactic structure is compatible with FALL(x), but not PUSH(x, y). A child could make such inferences even without knowing the meaning of *John* and *cup*, so long as she could determine the structure of the utterance, using say prosodic information, and determine that *John* and *the cup* were noun phrases, using other syntactic principles.

Gleitman carries this argument even further. In the above examples, structural information was used only as a filter, to select the correct interpretation from several possible interpretations of a given non-linguistic observation. She suggests however, that a verb's subcategorization frame gives substantial clues as to its meaning, independent of non-linguistic context. For example, the fact that the verb *explain* can take a sentential complement, as in *John explained that he was late for school*, indicates that it is a

[1] While they suggest that prosodic information alone can be used to construct the parse, they also assume that the child knows the syntactic category of the nouns and prepositions in the utterance. Since such category information can clearly aid the parsing process, I see no reason why they adopt the stronger claim of parsing using *only* prosodic information, given that they in any case assume the availability of further information. It would seem more felicitous to assume that the child can construct a parse tree using whatever information she has available, whether that be syntactic category information, prosodic information, or both.

verb of cognition-perception. A given verb may admit several different subcategorization frames, each further limiting its potential meaning. For example, the verb *explain* can also appear with a direct object and a destination, as in *John explained the facts to Mary*, indicating that it is also a verb of transfer. Taken together, these two utterances strongly limit the possible meaning for *explain*.

As outlined above, syntactic bootstrapping actually comprises two distinct strategies. They can be summarized by the following two hypotheses.

1. Children can determine the meaning of an unknown verb in an utterance by first determining the structure of that utterance using prosodic information alone, and then selecting as the correct verb meaning the one that allows that structure to have an interpretation consistent with non-linguistic context, given prior knowledge of the categories and meanings of the remaining words in the utterance.

2. Children can constrain the possible meanings of an unknown verb by finding those meanings that are compatible with each of the different subcategorization frames heard for that verb.

These two hypotheses may be combined to yield a single more comprehensive strategy. Both of these hypotheses, however, make two crucial assumptions. First, they assume the availability of prior language-specific information in the form of the word-to-meaning mappings, or at least word-to-category mappings, for the nouns and prepositions that appear as arguments to the unknown verb. Second, though not explicitly stated in their work, their methods appear to rely on the ability for prosodic parsing to determine a unique structure for each utterance. This thesis describes techniques for learning even without making the limiting assumptions of unambiguous parsing and prior language-specific knowledge.

The techniques described in this thesis could be extended to take prosodic information as input along with word strings. This would in essence form a synthesis of the ideas presented in this thesis with those advocated by Gleitman and her colleagues. One must be careful to include only those prosodic distinctions which are demonstrated to exist in the input, and which can be detected by children. This would include less information than say, a full syntactic analysis of the type performed by KENUNIA. Even though such prosodic information might be ambiguous and partial, the strategies described in this thesis could be used to find a language model which could consistently map the word strings to their meanings, subject to the constraints implied by the prosodic information. Such prosodic information would only ease the learning task when compared with the results presented in this thesis. If prosodic information was only partially available, or even totally absent, performance of this extended technique would degrade gracefully to the performance of the techniques discussed in this thesis. In order to experimentally verify this claim, one must formulate a representation for prosodic information, along with an appropriate linguistic theory constraining the possible syntactic analyses consistent with prosodic information specified in that representation. Such an experiment awaits future research.

5.1.3 Degree 0+ Learning

Lightfoot (1991) proposes a theory of how children determine parameter settings within a framework of universal grammar. His central claim is that children use primarily unembedded material as evidence for the parameter setting process. If this claim is true, a child must have access to sufficient structural information about the input utterances in order to differentiate embedded from unembedded material. Deriving such structural information requires that the learner determine constituent order prior to other parameter settings. Realizing this, Lightfoot suggests that children have access to syntactic category information before the onset of parameter setting and utilize a strategy whereby they wait for input utterances which are simple enough to uniquely determine the setting of some parameter.

(3) a. NP → Specifier N′
N′ → (Adj)[N or N′] PP
(7) a. XP → {Specifier, X′}
b. X′ → {X or X′, (YP)}
(8) a. the house
b. students of linguistics, belief that Susan left

Under (7), the linear order of constituents constitutes a parameter that is set on exposure to some trigger. The English-speaking child hears phrases like (8a) and, after some development, analyzes them as consisting of two words, one of a closed-class (*the*) and the other of an open class (*house*); in light of this and in light of the parameter in (7a), the child adopts the first rule of (3a). Likewise, exposure to phrases like (8b) suffices to set the parameter in (7b), such that the second rule of (3a) emerges. [...]

Consider, for a moment, the development that must take place before these parameters can be set. children acquire the sounds of their languages and come to use *men* as a word and a noun with the meaning roughly of the plural of 'man'. This is a nontrivial process, and many people have explained how it happens. Having established that *men* is a noun, children later acquire the constituent structure of *men from the city*, if I am right, by setting the parameters in (7) and projecting to NP accordingly via N′, yielding

$$[_{\text{NP}}\ \text{Spec}\ [_{\text{N}'}\ [_{\text{N}'}\ [_{\text{N}}\ \text{men}]][_{\text{PP}}\ \text{from the city}]]].$$

Lebeaux (1988) discusses this aspect of language acquisition very interestingly. In setting these particular parameters, children operate with partially formed representations that include $[_{\text{N}}$ men], $[_{\text{P}}$ from], $[_{\text{Spec}}$ the], and $[_{\text{N}}$ city]. They are operating not with "raw data" or mere words but with partially analyzed structures.

Men from the city and similar expressions occur in the child's environment with an appropriate frequency, and, given a partially formed grammar whereby *men* and *city* are classified as nouns, a child can assign a projection conforming to (7).

[pp. 6–7]

Lightfoot's proposal is thus very similar to Pinker's in this regard. It tacitly assumes that children determine constituent order from isolated utterances which uniquely determine that order. It uses a trigger-based approach in contrast to the cross-situational strategy advocated in this thesis. It is unclear whether Lightfoot's central claims about degree 0+ learnability are compatible with a cross-situational learning strategy. Such investigation merits future work.

5.1.4 Salveter

Salveter (1979, 1982) describes a system called MORAN, which like MAIMRA and DAVRA, learns word meanings from correlated linguistic and non-linguistic input. MORAN is presented with a sequence of utterances. Each utterance is paired with a sequence of two scenes described by a conjunction of atomic formula. Each utterance describes the state change occurring between the two scenes with which it is paired. The utterances are presented to MORAN in a preprocessed case frame format, not as word strings. From each utterance/scene-description pair in isolation, MORAN infers what Salveter calls a *conceptual meaning structure* (CMS) which attempts to capture the essence of the meaning of the verb in that utterance. This CMS is a subset of the two scenes that identifies the portion of the scenes referred to by the utterance. In this CMS the arguments of the atomic formula that are linked to noun phrases are replaced by variables labeled with the syntactic positions those noun phrases fill in the utterance. The process of inferring CMSs is reminiscent of the fracturing operation performed by MAIMRA and

DAVRA, whereby verb meanings are constructed by extracting out arguments from whole utterance meanings. MORAN's variant of this operation is much simpler than the analogous operation performed by MAIMRA and DAVRA since the linguistic input comes to MORAN preparsed. This preprocessed input implicitly relies on prior language-specific knowledge of both the grammar and the syntactic categories of the words in the utterance. MORAN does not model the acquisition of grammar or syntactic category information, and furthermore does not deal with any ambiguity that might arise from the parsing process. Additionally, MORAN does not deal with referential uncertainty in the corpus. Furthermore, the corpus presented to MORAN relies on a subtle implicit link between the objects in the world and linguistic tokens used to refer to these objects. Part of the difficulty faced by MAIMRA and DAVRA is discerning that a linguistic token such as *John* refers to a conceptual structure fragment such as **John**. MORAN is given that information *a priori* due to the lack of a formal distinction between the notion of a linguistic token and a conceptual structure expression. Given this information, the fracturing process becomes trivial. MORAN therefore, does not exhibit the cross-situational behavior attributed to MAIMRA and DAVRA, and in fact, learns every verb meaning from just a single utterance. This seems very implausible as a model of child language acquisition. In contrast to MAIMRA and DAVRA, however, MORAN is able to learn polysemous senses for verbs; one for each utterance provided for a given verb. MORAN focuses on extracting out the common substructure for polysemous meanings attempting to maximize commonality between different word senses and build a catalog of higher-level conceptual building blocks, a task not attempted by the techniques discussed in this thesis.

5.1.5 Pustejovsky

Pustejovsky (1987, 1988) describes a system called TULLY, which also operates in a fashion similar to MAIMRA, DAVRA, and MORAN, learning word meanings from pairs of linguistic and non-linguistic input. Like MORAN, TULLY is given parsed utterances as input. Each utterance is associated with a predicate calculus description of three parts of a single event described by that utterance: its beginning, middle, and end. From this input, TULLY derives a *thematic mapping index*, a data structure representing the θ-roles borne by each of the arguments to the main predicate. TULLY is thus similar to KENUNIA except that TULLY derives the θ-grids which KENUNIA currently must be given as prior language-specific knowledge. Like MORAN, the task faced by TULLY is much simpler than that faced by MAIMRA, DAVRA, or KENUNIA, since TULLY is presented with unambiguous parsed input, is given the correspondence between nouns and their referents, and does not have to deal with referential uncertainty since it is given the correspondence between a single utterance and the semantic representation of the event described by that utterance. TULLY does not learn language-specific syntactic information or word-to-category mappings. Furthermore, TULLY implausibly learns verb meanings from isolated utterances without any cross-situational processing. Multiple utterances for the same verb cause TULLY to generalize to the least common generalization of the individual utterances. TULLY however, goes beyond KENUNIA in trying to account for the acquisition of a variety of markedness features for θ-roles including [$\pm$motion], [$\pm$abstract], [$\pm$direct], [$\pm$partitive], and [$\pm$animate].

5.1.6 Rayner et al.

Rayner et al. (1988) describe a system that uses cross-situational techniques to determine the syntactic category of each word in a corpus of utterances. They observe that while in the original formulation, a definite clause grammar (Pereira and Warren 1980) normally defines a two-argument predicate `parser(Utterance,Tree)` with the lexicon represented directly in the clauses of the grammar, an alternate formulation would allow the lexicon to be represented explicitly as an additional argument to the parser relation, yielding a three argument predicate `parser(Utterance,Tree,Lexicon)`. This three argument relation can be used to learn syntactic category information by a technique summarized in

```
?- Lexicon = [entry(the,_),
              entry(cup,_),
              entry(slid,_),
              entry(from,_),
              entry(john,_),
              entry(to,_),
              entry(mary,_),
              entry(bill,_)],
   parser([the,cup,slid,from,john,to,mary],_,Lexicon),
   parser([the,cup,slid,from,mary,to,bill],_,Lexicon),
   parser([the,cup,slid,from,bill,to,john],_,Lexicon).
```

⇒

```
Lexicon = [entry(the,det),
           entry(cup,n),
           entry(slid,v),
           entry(from,p),
           entry(john,n),
           entry(to,p),
           entry(mary,n),
           entry(bill,n)].
```

Figure 5.1: The technique used by Rayner et al. (1988) to acquire syntactic category information from a corpus of utterances.

figure 5.1. Here, a query is formed containing a conjunction of calls to the parser, one for each utterance in the corpus. All of the calls share a common `Lexicon`, while in each call, the `Tree` is left unbound. The `Lexicon` is initialized with an entry for each word appearing in the corpus where the syntactic category of each such initial entry is left unbound. The purpose of this initial lexicon is to enforce the monosemy constraint that each word in the corpus be assigned a unique syntactic category. The result of issuing the query in the above example is a lexicon, with instantiated syntactic categories for each lexical entry, such that with that lexicon, all of the words in the corpus can be parsed. Note that there could be several such lexicons, each produced by backtracking.

Rayner et al. use a strong cross-situational strategy which is equivalent to the strategy used in section 3.2. The PROLOG program from figure 5.1 is a direct embodiment of the architecture depicted in figure 2.2. Part I extends the work of Rayner et al. in a number of important ways. First, the system described by Rayner et al. learns only word-to-category mappings from a corpus consisting only of linguistic input. MAIMRA and DAVRA learn word-to-meaning mappings in addition to word-to-category mappings by correlating the non-linguistic context with the linguistic input. Second, like MAIMRA, the system described by Rayner et al. is given a fixed language-specific grammar as input. DAVRA and KENUNIA learn language-specific grammatical information along with the lexicon. Third, like the first implementation of DAVRA, the system described by Rayner et al. keeps the whole corpus in memory throughout the learning process, using a simple chronological backtracking scheme to search for a lexicon consistent with the entire corpus. MAIMRA explores ways of representing the consistent language models using disjunctive lexicon formulae so that the corpus need not be retained in memory to support strong cross-situational learning. The revised implementation of DAVRA, along with KENUNIA, explore weaker learning strategies which also do not retain the corpus in memory. Nonetheless, the work of Rayner et al. was strong early motivation for the work described in this thesis.

5.1.7 Feldman

Feldman et al. (1990) have proposed a miniature language acquisition task as a touchstone problem for cognitive science. This task is similar in many ways to the language learning task described in part I of this thesis, combined with the visual perception task described in part II of this thesis. The proposed task is to construct a computer system with the following capacity.

> The system is given examples of pictures paired with true statements about those pictures in an arbitrary language.

> The system is to learn the relevant portion of the language well enough so that given a new sentence of that language, it can tell whether or not the sentence is true of the accompanying picture.

Feldman et al. go on to specify an instance of this general task, called the L_0 problem, where the pictures are constrained to contain only geometric figures of limited variation and the language fragment is constrained to describe only a limited number of spatial relations between those figures.

Feldman and his colleagues have explored a number of approaches to solving the L_0 problem. Weber and Stolcke (1990) describe a traditional symbolic approach where syntactic knowledge is represented as a unification grammar and semantic information is represented in first-order logic. This system however, does not learn. It is simply a query processor for L_0 as restricted to English. Stolcke (1990) describes a system which does learn to solve the L_0 task. This system is based on simple recurrent neural networks. The linguistic input to their system consists of a sequence of sentences such as *A light circle touches a small square*. These sentences are composed out of a vocabulary containing nineteen words. The words are presented one-by-one to the network, being represented as orthogonal 19-bit feature vectors. The non-linguistic input paired with each sentence consists of a semantic representation of a picture associated with that sentence. This semantic representation is encoded as a 22-bit feature vector of the following form.

$$\overbrace{\underbrace{\text{T L R A B}}_{\text{relation}}\ \underbrace{\text{F}}_{\text{mod}}}^{\text{Predicate}}\ \overbrace{\underbrace{\text{C S T}}_{\text{shape}}\ \underbrace{\text{S M L}}_{\text{size}}\ \underbrace{\text{D L}}_{\text{shade}}}^{\text{Argument 1}}\ \overbrace{\underbrace{\text{C S T}}_{\text{shape}}\ \underbrace{\text{S M L}}_{\text{size}}\ \underbrace{\text{D L}}_{\text{shade}}}^{\text{Argument 2}}$$

Once trained, the network acts as a map between a sentence and its semantic representation. The words of the sentence are presented to the network one-by-one. The semantic representation appears at the output of the network after the final word has been presented as input. The network thus includes some feedback to model the stored state during sentence processing. The network is trained using back-propagation while being presented with positive-only instances of sentences paired with their correct semantic representation. Thus their system does not admit referential uncertainty. The fragment of L_0 that Stolcke considers allows a total of 5052 distinct sentences. Of these, 353 were used as training sentences and the remainder as test sentences. Stolcke does not report the percentage of test sentences which his system is correctly able to process, except for stating that the training set contained 61 out of all 81 possible 'simple NP sentences' and that the system generalized correctly to the remaining 20 simple NP sentences. Weber (1991) and Stolcke (1991) describe more recent continuation of this work.

5.2 Discussion

An ultimate process account of child language acquisition must meet two criteria. It must be able to acquire any language which children can acquire, and it must be able to do so for any corpus on which a child would be successful. It would be very hard to prove that any given algorithm met these two universal criteria since we lack information which would allow us to perform such universal quantification. We have little information that circumscribes the child-learnable languages, or the situations which support that learnability. Rather than a formal proof of adequacy, a more reasonable approach would be to amass quantitative evidence that a given algorithm can acquire many different languages given a variety of corpora in those languages. This thesis takes only a first, exceedingly modest, step in that direction, with the demonstration that DAVRA can process very small fragments of both English and Japanese. The longer-term goal of this research is to extend this ability to process larger corpora in different languages. Larger corpora are needed to guarantee that the algorithms scale. Ideally, such corpora should consist of transcripts of actual parental speech to children, instead of the synthetic text currently used.

Successfully processing large natural corpora requires surmounting a number of hurdles in addition to the problem of developing a syntactic theory capable of accounting for the linguistic phenomena in the corpus. One technical difficulty is that the learning strategy proposed here requires non-linguistic annotation for the linguistic input. (Remember, "You can't learn a language simply by listening to the radio.") Available transcriptions do not come with such annotations, at least not annotations in the correct form or which contain the information needed to put it in the correct form. There is a way around this problem. One could use an available dictionary to parse the corpus under a fixed set of parameter settings. Pseudo-semantic information can then be derived from the resulting parse trees. The parse trees themselves can be taken as meaning expressions in a MAIMRA/DAVRA framework. Alternatively, one could construct a KENUNIA style θ-map by applying θ-theory in reverse. Each noun could be given a random token as its referent. Other terminals would be given $\perp$ as their referent. The θ-criterion requires that complements of non-functional categories be θ-marked. For each such complement configuration, a θ-mapping is constructed matching a randomly chosen θ-role to the ultimate referent of the complement. In both of these cases, noise would then be added to model referential uncertainty. For the MAIMRA/DAVRA framework, the correct meaning expression would be added to a set of random alternate expressions, possibly derived as perturbations of the correct meaning. For the KENUNIA framework, several other random θ-mappings could be added to the θ-map. The learning algorithm would then be applied to this corpus, without access to the dictionary and parameter settings used in its construction. The algorithm would be deemed successful if it could accurately reconstruct the dictionary and parameter settings. This technique for pseudo-semantic annotation has an added benefit. By varying the amount of noise added to the non-linguistic input one could analytically determine the sensitivity of the learning algorithms to such noise. Such sensitivity predictions could be compared with actual sensitivity measurements performed on children as an experimental test of predictions made by the theory.

A much more serious hurdle remains, however, before the above experiment could be attempted. The cross-situational learning strategy advocated in this thesis requires that the learner find a single grammar and lexicon that can consistently explain an entire corpus. This would be virtually impossible for natural corpora for three reasons. First, natural corpora contain ungrammatical input. Even ignoring input that is truly ungrammatical, the current state of the art in linguistic theory is not capable of accounting for many phenomena occurring in natural text. While such text is grammatical in principle, it must be treated as ungrammatical relative to our meager linguistic theories. Any strict cross-situational learning strategy would fail to find a language model consistent with a corpus that contained ungrammatical input. Children however, can learn from input a substantial fraction of which is ungrammatical. Second, a key assumption made by each of the systems discussed in part I of this thesis was the monosemy constraint, the requirement that each word map to a unique category and meaning. This assumption is clearly false. Polysemy runs rampant in human language. Here again, a strict cross-situational strategy would fail to find a consistent language model when presented with a corpus that could only be explained by a polysemous lexicon. Children however, have no difficulty learning polysemous words. A final hurdle involves referential uncertainty. What if the set of meanings conjectured by the learner as a possible meaning of some observed utterance does not contain the correct meaning? This could happen if the correct meaning of some utterance is not readily apparent from its non-linguistic context, or if the learner incorrectly discards the correct meaning, by some measure of salience, to reduce the referential uncertainty and make cross-situational learning more tractable. In this situation again, the learner, not knowing that no possible meaning was hypothesized for the utterance, would fail to find a consistent language model.

Each of these three problems is symptomatic of a single more general problem: noise in the input. Such noise can be dealt with using a variety of techniques. One way would be to assign weights to different lexical entries and parameter settings, making the decision between alternative lexical entries and parameter settings a graded one, rather than an absolute one. A scheme could be adopted for

increasing the weights of those alternatives that correctly explain some input while decreasing the weights of those alternatives that fail to explain some input, ultimately choosing those alternatives with the higher weight. In the language acquisition literature, such weights are often confused with probabilities. While weights might have a probabilistic interpretation, they need not have one.

There are alternatives to weights. One could instead find a language model that minimized violations of the linguistic theory. This offers a spectrum of alternative ways of counting violations. At one end of the spectrum, the linguistic theory can be treated as a black box, either capable or incapable of parsing an utterance given a language model. With such a theory, the learner would simply minimize the number of utterances which could not be parsed. This might not work if the linguistic theory was so poor that it could parse relatively few utterances in the corpus. A more general approach, still using an encapsulated linguistic theory, would be to allow utterances to be parsed with minor perturbations of the language model and choose the language model which allowed the corpus to be parsed with the minimal total associated cost. An even more general approach would be to have the parser produce a quality measure as output. Successful parses would have a high quality measure while unsuccessful parses would still have a non-zero quality measure if they could 'almost' be parsed. The quality measure could be based on which components of a modular grammatical theory were violated. In this case, the learner would choose the model which maximized the total quality of the parsed corpus.

While these approaches can deal with all forms of noise, it seems unreasonable to consider polysemy as noise. A similar but more plausible strategy could be used to support polysemy. The language model could be extended to allow polysemous lexical entries. The cost of a language model could be defined so that it measured the amount of polysemy in the lexicon. The learner could then find the lowest cost language model, i.e. the one with least polysemy, that could still consistently account for the corpus. While all of the above approaches are conceptually straightforward, substantial details remain to be worked out. This is left for future research.

Part II

Grounding Language in Perception

Chapter 6

Introduction

Part II of this thesis advances a theory of event perception. When people observe the world they can generally determine whether certain events have happened. Furthermore, they can describe those events using language. For instance, after seeing John throw a ball to Mary, the observer can say that the event described by the utterance *John threw the ball to Mary* has happened, along with perhaps events described by other utterances. Part II of this thesis suggests a mechanism to describe how event perception may work. This mechanism has been partially implemented in a computer program called ABIGAIL. ABIGAIL watches a computer-generated animated stick-figure movie and constructs descriptions of the events that occur in that movie. The input to ABIGAIL consists solely of the positions, orientations, shapes, and sizes of the line segments and circles which constitute the image at each frame during the movie. Figure 6.1 illustrates one frame of a movie presented to ABIGAIL. From this input, ABIGAIL segments the image into objects, each object comprised of several line segments and circles, and delineates the events in which those objects participate.

At the highest level, ABIGAIL can be described as a program that takes an utterance and a movie segment as input, and determines whether that utterance describes an event that occurred during that movie segment.

$$\textsc{Abigail}(u, m) \rightarrow \{\mathbf{true}, \mathbf{false}\}$$

Alternatively, ABIGAIL can be thought of as a program that takes a movie segment as input, and produces utterances that describe the events which occurred during that segment.

$$\textsc{Abigail}(m) \rightarrow \{u\}$$

ABIGAIL does not, however, directly relate utterances to movies. An intermediate semantic representation mediates between an utterance and a movie. For example, the semantic representation for the utterance *John threw the ball to Mary* might be CAUSE(**John**, GO(**ball**, TO(**Mary**))). The intermediate semantic representation connects two halves of ABIGAIL. One half relates the semantic representation to the movie while the other half relates it to an utterance. The general architecture is depicted in figure 6.2. In this architecture, the box labeled 'perception' relates semantic descriptions to movies. It can be thought of either as a predicate

$$\mathbf{perception}(s, m) \rightarrow \{\mathbf{true}, \mathbf{false}\}$$

that determines whether the event described by some semantic expression s occurred during the movie segment m, or alternatively as a function

$$\mathbf{perception}(m) \rightarrow \{s\}$$

Figure 6.1: A typical frame from a movie which is shown to ABIGAIL. The objects in the frame, such as tables and chairs, are constructed solely from line segments and circles.

Figure 6.2: A depiction of the architecture of ABIGAIL's language faculty. It contains three processing modules: a parser, a linker, and a perceptual component, that mutually constrain five representations: the input utterance, the syntax of that utterance, the meaning of that utterance, the visual perception of events in the world, and a language model comprising a grammar and a lexicon. The lexicon in turn maps words to their syntactic category and meaning. Given observations and a lexicon as input, this architecture can produce as output, utterances which explain those observations. The long-term objective is to combine ABIGAIL's perceptual component with the language learning techniques described in part I of this thesis to provide a comprehensive model of language acquisition. As a language acquisition device, when given pairs of observations and utterances which explain those observations as input, this architecture will produce as output, a language model for the language in which those utterances were phrased. Part I of this thesis elaborates on this language acquisition process.

that produces a set of semantic expressions describing those events which occurred during the movie segment. The two remaining boxes in figure 6.2 relate the semantic representation to an utterance.

The architecture depicted in figure 6.2 is a very general mechanism for grounding language in perception. As discussed on page 27, it can support the comprehension, generation, and acquisition of language. Part I of this thesis focussed on using this architecture to support language acquisition. It described the parser and linker modules in detail as they related to the language acquisition task. Part II of this thesis will focus solely on the perception module, i.e. mechanisms for producing semantic descriptions of events from (simulated) visual input. The two halves of this thesis discuss the two halves of this architecture independently. The reason for this is that the two halves have not yet been integrated into a single implementation. This integration awaits further research.

After displaying the architecture in figure 6.2, a natural first question that arises is: *What is an appropriate intermediate semantic representation?* Semantic representations are normally taken to encode the meaning of an utterance. Chapter 7 argues that the notions of *support*, *contact*, and *attachment* are central to defining the meanings of simple spatial motion verbs such as *throw*, *pick up*, *put*, and *walk*. For instance, throwing involves moving one's hand while grasping an object (attachment), resulting in the unsupported motion of that object. Chapter 7 further motivates the need for including the notions of support, contact, and attachment as part of a semantic representation scheme by demonstrating the central role these notions play in numerous spatial motion verbs. Definitions for these verbs are presented in a novel representation scheme that incorporates these notions. These definitions are compared with those proposed by other researchers which do not incorporate such notions. I claim that incorporating

the notions of support, contact, and attachment allows formulating more precise definitions of these verbs.

If one accepts the argument that the semantic representation should incorporate the notions of support, contact, and attachment, a second question arises: *How does one perceive support, contact, and attachment relationships?* An answer to this question is necessary in order to construct the perception box from figure 6.2. Chapter 8 offers a unified answer to that question: *counterfactual simulation*. An object is supported if it does not fall when one imagines the short-term future. Likewise, one object supports another object if the latter is supported but loses that support when one imagines the short-term future of a world without the former object. When one object supports another they must be in contact with each other. Furthermore, two objects are assumed to be attached to each other if such an attachment must be hypothesized to explain the fact that one object supports the other. Chapter 8 elaborates on these ideas. A simplified version of these ideas has been implemented in ABIGAIL. ABIGAIL uses counterfactual simulation to determine the attachment relations between the line segments and circles which constitute each frame of the movie she watches. This allows her to aggregate the line segments and circle into objects. She then uses counterfactual simulation to determine support, contact, and attachment relations between those objects. Chapter 8 also discusses some experiments performed by Freyd et al. (1988) which give evidence that human visual perception operates in an analogous fashion.

If one accepts the claim that support, contact, and attachment relations are recovered by counterfactual simulation, a third question then arises: *What is the nature of the mechanism used to perform counterfactual simulation?* Nominally, the simulator predicts the behavior of machine-like mechanisms, parts connected by joints, under the influence of forces such as gravity. Chapter 9 argues however, that traditional approaches to kinematic simulation, namely those based on numerical integration, are inappropriate as cognitive models of the human imagination capacity since the traditional approaches take physical accuracy to be primary and collision detection to be secondary. In contrast, human visual perception appears to take certain naive physical notions such as substantiality, the constraint that solid objects can't pass through one another, and continuity, the constraint that objects must follow continuous paths during motion, to be primary. Chapter 9 presents a kinematic simulator for the micro-world of line segments and circles which takes substantiality and continuity, along with gravity, to be primary. This simulator directly encodes such principles allowing it to quickly predict in a single step, for instance, that an object will fall precisely the distance required for it to come in contact with the object beneath it. Traditional simulators based on numerical integration would require many small perturbations to make such a prediction. While such simulators are more accurate than the simulator described here, and can simulate a larger class of mechanisms, the simulator described in chapter 9 is much faster and better suited to the task of discerning support, contact, and attachment relations. Chapter 9 also discusses some experiments performed by Baillargeon et al. (1985), Baillargeon (1986, 1987), and Spelke (1988) which give evidence that young infants are sensitive to violations of naive physical constraints such as substantiality and continuity. The remainder of this chapter describes the event perception task faced by ABIGAIL since this task motivates the formulation of the algorithms discussed later in part II of this thesis.

6.1 The Event Perception Task

ABIGAIL is shown a computer-generated animation depicting objects such as tables, chairs, boxes, balls, and people. During the movie, the objects participate in events. The people walk, pick up, and put down objects, and so forth. The task faced by ABIGAIL is to determine which events occur and when they happened. For instance, after a movie segment depicting John walking to the table, she is to produce a representation of the utterance *John walked to the table*. For simplicity, the movie shown to ABIGAIL is a stick figure animation, constructed solely from line segments and circles. These line segments and

circles, collectively called *figures*, constitute the lowest level structure of the image. Higher-level *objects*, such as tables, chairs, and people, are constructed out of collections of figures. Figure 6.1 shows a typical frame from one of the movies which is shown to ABIGAIL.

The movie shown to ABIGAIL consists of a sequence of such frames containing objects built out of figures. As the movie progresses, the objects move about and participate in various *events*. ABIGAIL is not given any explicit information about the non-atomic entities in the movie. She is not told which collections of figures constitute objects nor is she told which events they participate in. Furthermore, she is not even told what types of objects exists in the world or what types of events can occur. The only input that ABIGAIL receives is the position, orientation, shape, and size of the figures in each movie frame.

ABIGAIL faces a two-stage task. First, she must recover a description of the objects and events occurring in the movie, solely from information about the constituent figures. Second, she must form a mapping between the recovered object and event representations, and the linguistic utterances which describe those events. To date, only part of the first task has been accomplished. The second task has not been attempted. Part II of this thesis therefore, addresses only the first task. It proposes a novel approach to the task of event perception and presents, in detail, the mechanisms underlying this approach. As discussed in chapter 1, the long-term goal of this research is to use the object and event representations recovered by ABIGAIL as the non-linguistic input to language acquisition models such as those described in part I of this thesis. Linking models of language acquisition to models of event perception would allow a comprehensive study of the acquisition of word meanings in a way which is not possible without perceptual grounding of those word meanings.

The perceptual mechanisms used by ABIGAIL to recover object and event descriptions are very general. Unlike some prior approaches, they do not incorporate any knowledge that is specific to any class of objects or events. Thus, they do not contain models of particular objects such as tables or particular events such as walking. The intention is that the same unaltered perceptual mechanism be capable of recovering reasonable object and event descriptions from any movie constructed out of line segments and circles.

In order to verify whether ABIGAIL's unaltered perceptual mechanisms are indeed capable of analyzing any movie, a simple movie construction tool was created to facilitate the generation of numerous movies with which to test ABIGAIL. This tool takes a *script* and generates the positions, orientations, shapes, and sizes of the figures at each frame during the movie. While the script itself delineates objects and events, the perceptual mechanisms of ABIGAIL have no access to the representation of objects and events in the script and must recover the object and event information solely from the positions, orientations, shapes, and sizes of the figures in the movie generated from the script.

A sample movie script is shown in figure 6.3. This script generates a movie consisting of 1063 frames, the first of which is depicted in figure 6.1. Each frame is constructed from 43 figures: 5 circles and 38 line segments. These figures form caricatures of 7 objects: a table, two chairs, a box, a ball, a man, and a woman. The script of this movie is simple and fairly boring. The man, John, walks over to the table and picks up the ball. He turns around and walks back to his original position. He then turns around again, walks back to the table, puts the ball down on the table, turns around, and walks back to his original position. The woman, Mary, then performs a similar task. Finally, John walks toward the table, picks up the ball, carries it over to Mary, and gives it to her. He then turns around and walks back to his place, after which Mary walks toward the table, puts the ball on the table, and returns to her place. Figure 6.4 depicts the general sequence of events in this movie by showing a selection of several key frames from the movie.

The original expectation was that ABIGAIL would be able to successfully process numerous movies. That goal was overly ambitious. Most of the development of ABIGAIL was driven by only one movie, the one generated by the script in figure 6.3 and depicted in figure 6.4. In fact, due to computer processing limitations and to the current incomplete state of ABIGAIL's implementation, only a portion of that

```
(define-movie movie1 ((table (make-instance
                              'table :name 'table :x 16.0 :y 0.0 :world world))
                      (chair1 (make-instance
                               'chair :name 'chair1 :x 12.0 :y 0.0 :world world))
                      (chair2 (make-instance
                               'chair
                               :name 'chair2 :x 20.0 :y 0.0 :direction -1.0 :world world))
                      (box (make-instance 'box :name 'box :x 18.0 :y 2.525 :world world))
                      (ball (make-instance
                             'ball :name 'ball :x 14.0 :y 3.0 :world world))
                      (john (make-instance
                             'man :name 'john :x 3.0 :y 0.0 :world world))
                      (mary (make-instance
                             'woman
                             :name 'mary :x 30.0 :y 0.0 :direction -1.0 :world world)))
 (walk-to john (x (center ball)))
 (pick-up (left-hand john) ball)
 (about-face john)
 (walk-n-steps john 4)
 (walk-to john (x (center table)))
 (put-down (left-hand john)
           (x (center table))
           (+ (y (point1 (top table))) (size (circle ball))))
 (about-face john)
 (walk-n-steps john 4)
 (about-face john)
 (walk-to mary (x (center ball)))
 (pick-up (left-hand mary) ball)
 (about-face mary)
 (walk-n-steps mary 5)
 (walk-to mary (x (center table)))
 (put-down (left-hand mary)
           (x (center table))
           (+ (y (point1 (top table))) (size (circle ball))))
 (about-face mary)
 (walk-n-steps mary 5)
 (about-face mary)
 (walk-to john (x (center ball)))
 (pick-up (right-hand john) ball)
 (walk-to john (x (center mary)))
 (give (right-hand john) (left-hand mary))
 (about-face john)
 (walk-n-steps john 9)
 (walk-to mary (x (center table)))
 (put-down (left-hand mary)
           (x (center table))
           (+ (y (point1 (top table))) (size (circle ball))))
 (about-face mary)
 (walk-n-steps mary 5)
 (about-face mary))
```

Figure 6.3: A script used to generate a movie to be watched by ABIGAIL. The first frame of this movie is shown in figure 6.1. The general sequence of events in this movie is depicted by the selection of frames in figure 6.4.

Figure 6.4: Several key frames depicting the general sequence of events from the movie used to drive the development of ABIGAIL. The script used to generate this movie is given in figure 6.3. Frame 0 is shown in greater detail in figure 6.1.

movie has been successfully processed and analyzed by ABIGAIL. Future work will attempt to extend the results described in this thesis by running ABIGAIL on other movies.

6.2 Outline

The remainder of part II of this thesis contains four chapters. Chapter 7 advances that claim that the notions of support, contact, and attachment play a pivotal role in defining the prototypical meanings of simple spatial motion verbs. It surveys past attempts at defining the meanings of many such verbs, finding these attempts inadequate. An alternative representation scheme is put forth which highlights the notions of support, contact and attachment. Chapter 8 proposes a computational mechanism, implemented in ABIGAIL, for perceiving support, contact, and attachment relations. It advances the claim that such relations are not recovered by static analysis of images but rather require counterfactual simulation. Chapter 9 suggests that the simulation performed as part of event perception differs from traditional kinematic simulation in that it takes the naive physical notions of substantiality, continuity, gravity, and ground plane to be primary, and physical accuracy and coverage to be secondary. It describes in detail, the novel kinematic simulator that acts as ABIGAIL's imagination capacity. Chapter 10 discusses related work and concludes with an outline of potential future work.

Chapter 7

Lexical Semantics

Part II of this thesis advances a theory of event perception. It proposes a mechanism for how people visually recognize the occurrence of events described by simple spatial motion verbs such as *throw*, *walk*, *pick up*, and *put*. The proposed recognition process is decompositional. Each event type is successively broken down into more basic notions that ultimately can be grounded in perception. For instance, a throwing event comprises two constituent events: moving one's hand while grasping an object, followed by the unsupported motion of that object. The words *grasping* and *unsupported* play a pivotal role in this description of throwing. An event would not typically be described as throwing if it did not involve the grasping and releasing of an object along with the resulting unsupported motion. Many prior approaches to defining the meaning of the word *throw* (e.g. Miller 1972, Schank 1973, Jackendoff 1983, and Pinker 1989), however, do not highlight this pivotal role. In this chapter, I advance the claim that the notions of *support*, *contact*, and *attachment* are central to describing many common spatial motion events. Accurately delineating the occurrence of such events from non-occurrences hinges on the ability of perceiving support, contact, and attachment relationships between objects in the world. In chapters 8 and 9, I offer a theory of how to ground the perception of these relations.

A central assumption of this work is that perception is intimately tied to language. We use words and utterances to describe events that we perceive. The meaning of a word is typically thought of as conditions on its appropriate use. It thus seems natural to relate the meaning of a word such as *throw* to a procedure for detecting throwing events. Many schemes have been proposed for representing the meanings of words and utterances (cf. Miller, Schank, Jackendoff, and Pinker). I will show that these schemes cannot be taken as procedures for recognizing the events that they attempt to describe because they lack the notions of support, contact, and attachment. Accordingly, I propose a different representation scheme that incorporates these notions into definitions of word meanings. The central focus of this work is the ability for recognizing events by grounding the notions of support, contact, and attachment. Therefore, the representation scheme developed here exaggerates the role played by these notions.

For the remainder of this chapter, I will discuss the meanings of a number of spatial motion verbs. I will show how prior definitions proposed for these verbs cannot be used as event recognition procedures. For each verb I will then propose an alternate definition that highlights the role played by the notions of support, contact, and attachment in characterizing the events described by that verb.

Consider the word *throw*. The Random House dictionary (Stein et al. 1975) offers the following definition for *throw*.

> **throw** *v.t.* **1.** to propel or cast in any way esp. to project or propel from the hand by a sudden forward motion or straightening of the arm and wrist

This definition comprises two parts: a general condition and a more prototypical situation. Both of these, however, admit events which one would not normally consider to be throwing events, for instance, rolling a bowling ball down a bowling lane. A sign at a bowling alley that said 'Please do not throw balls down the alley' does not consider rolling a bowling ball to be throwing. The difference lies in whether or not the resulting motion is unsupported.

Miller (p. 355) offers the following definition for *throw*.

to apply force by hand to cause to begin to travel through air

At first glance, it appears that Miller is attempting to capture the notion of support through the statement **through air**. We might take the statement **through air** not as literally meaning 'through air', which would admit supported motion through the air, but as a gloss for unsupported motion. But elsewhere Miller groups **through air** along with **through water** and **on land** as the *medium* of motion. Miller defines *swim* as **to travel through water** (p. 351) and *walk* as **to travel on land by foot** (p. 345). Furthermore, as we shall see, the glosses given by Miller for other words whose definitions require the notion of support do not incorporate the **through air** primitive.

Schank offers the following two definitions for *throw*.

The first describes throwing as propelling an object Z on a path from the agent X to the destination Y. The second appears to add the statement that Z must actually reach Y to be thrown *to* its destination. Neither of these definitions mention the unsupported nature of the resulting motion.

Jackendoff (p. 175) offers the following gloss for the statement *Beth threw the ball out the window.*

$$\text{CAUSE}(\textbf{Beth}, \text{GO}(\textbf{ball}, \text{OUT}(\textbf{window})))$$

While in this example, the unsupported nature of the resulting motion is implied by the fact that the ball is being thrown out the window, nothing in the representation conveys this information. If one takes $\text{CAUSE}(x, \text{GO}(y, z))$ as the meaning of *throw*, this definition admits many non-throwing events.

Pinker (p. 218) offers the following definition for the word *throw* via the gloss for the statement *Bob threw the box to Bill.*[1]

[1]For typographical reasons, I have omitted the time-line component of Pinker's representations. It is not relevant to the current discussion. The method for annotating effect and for/to branches is altered somewhat as well, again for typographical reasons.

This gloss encapsulates the distinction between throwing and non-throwing events in the manner attribute "throwing". Since this is an uninterpreted symbol, it offers little help in building a procedure for recognizing throwing events.

In short, none of the representation schemes proposed by Miller, Schank, Jackendoff and Pinker contain a primitive for describing support. Thus in these schemes, one could not reformulate better definitions around the notion of support without adding such a primitive. Pinker (p. 201) gives the following definition for the word *support*

but does not recognize the need to incorporate this structure as part of the definitions of other words which depend on support.

The definitions for *throw* given by Schank, Jackendoff, and Pinker also do not mention the role played by one's hand in throwing an object. Numerous non-throwing events such as kicking, or bumping into an object causing it to fall, would satisfy the above definitions even though they are not prototypical throwing events. Random House and Miller attempt to capture this requirement via the statements 'from the hand' or **by hand**. Even these do not express the notion that prototypical throwing involves grasping an object and subsequently releasing it. Combined with not specifying unsupported motion, not specifying this grasping-releasing transition allows all of the definitions for *throw* given by Miller, Schank, Jackendoff, and Pinker to admit many non-throwing events such as pushings, pullings, and carryings. In fact, even the Random House definition would suffer from this problem were it not for the words 'or cast' appended to 'propel' in its definition for *throw*.

In contrast, I propose the following alternative definition for *throw*.

```
(define throw (x y)
 (exists (i j)
  (and (during i (move (hand x)))
       (during i (move y))
       (during i (contacts (hand x) y))
       (during i (attached (hand x) y))
       (during j (not (contacts (hand x) y)))
       (during j (not (attached (hand x) y)))
       (during j (move y))
       (during j (not (supported y)))
       (= (end i) (beginning j)))))
```

Informally, this states that a throwing event comprises two consecutive time intervals i and j, where during i, both x's hand and y are moving, and x's hand is in contact with and attached to y, while during j, x's hand is no longer in contact with and attached to y, and y is in unsupported motion. Note that this definition incorporates the grasping and releasing action of the agent followed by the unsupported motion motion of the patient, aspects of throwing not captured by the definitions advanced by Miller, Schank, Jackendoff, and Pinker. I will not formally define the notation used for defining words. In fact, I have taken some liberty with the notation, sacrificing precision in favor of expository simplicity. What I hope to convey, however, is the belief that if one could ground the notions of support, contact, and attachment, in addition to movement, one could use the above definition as a procedure for perceiving throwing events.

I should stress that I do not advance such a definition as embodying the necessary and sufficient conditions for the use of the word *throw*. Even ignoring metaphorical and idiomatic uses, the word *throw* can be extended to a variety of situations. The above definition attempts to describe only prototypical throwing events. It is a well-known philosophical quagmire to attempt to formally circumscribe the meaning of a word or even to characterize prototypical events and their extensions. To avoid such difficulties, I will simply say that the definitions presented here try to capture our intuitive notions of the events they describe, better than prior representations. I offer no way to substantiate this claim except for the projected eventual success in using these definitions as part of an implemented computer program to accurately differentiate occurrences from non-occurrences of the events they describe in animated movies. Since the implementation of that program is still underway, I can only hope to convince the reader that the mechanisms I propose in part II of this thesis show some actual promise of achieving these aims. One should note that neither Miller, Schank, Jackendoff, nor Pinker offer any better substantiation of their respective representation schemes.

I also want to point out a number of issues pertaining to the above definition and others like it. First, it does not specify precisely *when* the throwing event occurred. For most verbs like *throw*, it is unclear whether the actual event described spanned both i and j, just i or j, some portion of either i or j, or just the transition between i and j. The notation intentionally leaves this question unanswered in the absence of suitable criteria for determining the appropriate solution. The intention is to interpret the notation as stating that the event occurred sometime during the interval spanning i and j given that the criteria for i and j are met. Second, the definition does not express certain other notions that we intuitively believe to be part of throwing events. For instance, x's hand imparting force to y during i, or that force causing the unsupported motion during j. Clearly notions such as force application and causality play an important role in the meaning of most spatial motion verbs. I leave such notions out of definitions simply because I do not yet know how to perceptually ground them. Section 10.2 will offer some speculation on how the methods described in part II of this thesis can be extended to support perception of force application and causality, allowing such notions to be included in revised definitions for verbs like *throw*. Finally, the above definition contains redundant information. Stating that x's hand is attached to y during i implies that it contacts y during that interval as well. Likewise, stating that x's hand is moving

during i, while it is attached to y, implies that y must also be moving during that interval. Furthermore, stating that y is unsupported during j implies that x's hand is neither in contact with, nor attached to, y during that interval. I include such redundant information for two reasons. First, it may allow more robust detection of events given unreliable primitives. Second, the redundant prototypical definition is more suitable for extension to non-prototypical situations. For example, throwing that does not involve unsupported motion of an object still involves the release of that object at some point during its motion. Perhaps some variant of structure mapping (Gentner 1983, Falkenhainer et al. 1989) applied to such redundant definitions can form a basis for generalizing prototype definitions to idiomatic, metaphorical, and other extended uses (cf. Lakoff 1987).

Putting these and many other subtleties aside then, let us examine some other verbs for which support, contact, and attachment play an important role. Consider the verbs *fall*, *drop*, *bounce*, and *jump*. Miller (p. 357) gives the following definitions for these words.

fall:	**to travel downward**
drop:	**to cause to travel downward**
bounce:	**to travel up and down**
jump:	**to travel over**

These definitions seem not to accurately capture the meanings of these words since they lack the notion of support, contact, and attachment. Falling is unsupported motion. One is not falling when one is walking down stairs. Dropping must result in falling. One is not dropping a tea cup when one is gently placing it onto its saucer. Furthermore, not just any causation of falling counts as dropping. Pushing or knocking an object off a ledge is not dropping that object. Dropping an object requires that the agent previously grasp, or at least support, that object prior to its falling. Bouncing seems to involve temporary contact more than up-and-down motion. One can bounce a ball horizontally against a wall. Furthermore, not all up-and-down motion is bouncing. A book is not bouncing when one picks it up and puts it down somewhere else. Jumping too, seems to involve support, in particular a self-induced state change from being supported to being unsupported, typically incorporating upward motion. One need not travel over something to successfully jump.

Schank gives the following definitions for *fall* and *drop*.

X *fall*: $\text{nf} \Longleftrightarrow \text{PROPEL} \xleftarrow{\text{O}} \text{Z} \xleftarrow{\text{D}}$ ground

$\text{X} \overset{t_F}{\Longleftrightarrow} \text{GRASP} \xleftarrow{\text{O}} \text{Z}$

$\Downarrow$ r

X *drop* Z: $\text{nf} \Longleftrightarrow \text{PROPEL} \xleftarrow{\text{O}} \text{Z} \xleftarrow{\text{D}}$ ground

These require only that 'nf', the natural force of gravity, propel an object toward the ground, and do not require the object to be unsupported. They admit a situation where one is lowering a bucket into a well as a case where one dropped the bucket and it is falling.

In contrast, I propose the following definitions for the verbs *fall*, *drop*, *bounce*, and *jump*.

```
(define fall (x)
 (exists (i)
  (and (during i (not (supported x)))
       (during i (move-down x)))))
```

```
(define drop (x y)
 (exists (i j)
  (and (during i (contacts (hand x) y))
       (during i (attached (hand x) y))
       (during i (supports x y))
       (during i (supported y))
       (during j (not (contacts (hand x) y)))
       (during j (not (attached (hand x) y)))
       (during j (not (supports x y)))
       (during j (not (supported y)))
       (during j (move-down y))
       (= (end i) (beginning j)))))

(define bounce (x)
 (exists (i j k y)
  (and (during i (not (contacts x y)))
       (during j (contacts x y))
       (during k (not (contacts x y)))
       (= (end i) (beginning j))
       (= (end j) (beginning k))
       (short j))))

(define jump (x)
 (exists (i j)
  (and (during i (supported x))
       (during j (not (supported x)))
       (during j (moving-up x))
       (= (end i) (beginning j)))))
```

Intuitively, these definitions state that *falling* involves unsupported downward motion, that *dropping* involves releasing a previously grasped object allowing it to fall, that *bouncing* involves temporary contact and that *jumping* involves the transition from being supported to unsupported upward motion. Again, they are not meant as necessary and sufficient conditions on the use of these words, only as descriptions of prototypical events. More importantly, they can be used as procedures for recognizing occurrences of the events they describe.

There seems to be no single unified notion of support. The intuitive concept of support breaks down into at least three variant notions, each corresponding to a different way an object can fall. An object can fall straight downward, fall over pivoting about a point beneath its center-of-mass, or slide down an inclined plane. Whether or not an object is supported in one way, preventing one type of falling, may be independent of whether it is supported in a different way. Figure 7.1 illustrates several different potential support situations for an object. In figure 7.1(a), the object is totally unsupported and will fall down. In figure 7.1(b), the object is prevented from falling down but will fall over. In figure 7.1(c), the object is prevented from falling down but can either fall over or slide. In figure 7.1(d), the object will neither fall down nor fall over but will slide. In figure 7.1(e), the object is totally supported and will not fall down, fall over, or slide. Difference in type of support appears to play a role in verb meaning. For instance, throwing seems to require that an object be able to fall down, or at least fall over, as in *The wrestler threw his opponent to the floor*. An event is not throwing if it results in unsupported sliding motion. Similarly, falling, dropping, and jumping most prototypically involve the ability to fall down but may be extended to cases of falling over and perhaps even to sliding. Other verbs are sensitive to this distinction in different ways. For instance, the verb *lean on* can be used only to describe situations

Figure 7.1: The different varieties of support relationships. In (a), the object is totally unsupported and will fall down. In (b), the object is prevented from falling down but will fall over. In (c), the object is prevented from falling down but can either fall over or slide. In (d), the object will neither fall down nor fall over but will slide. In (e), the object is totally supported and will not fall down, fall over, or slide.

where one object prevents another from falling over, and not when one object prevents another from falling down. One is not leaning on the floor when one is standing on it.

Consider now the verb *put*. Miller (p. 359) defines *put* as **to cause to travel**. Jackendoff (p. 179) offers

$$\text{CAUSE}(\textbf{man}, \text{GO}(\textbf{book}, \text{TO ON}(\textbf{table})))$$

as the meaning of *The man put the book on the table*. Pinker (p. 180) gives the following fragment of a definition for *put*.

All of these definitions involve causing an object to move to a destination. Such a definition is overly general. Jackendoff's expression would be true of an event where the man knocked the book off the shelf onto the table, yet one would not say that he put the book there. *Put* seems to require the ability to control the precise final destination of an object. One does not usually have such control when one throws or kicks an object, so one doesn't use the word *put* to describe such situations. One way to achieve greater positional control is by grasping or otherwise supporting an object while moving it. Furthermore, positional control is achieved only if the object is supported at the end of the *put* event. This support must come from something other than the hand which moved it. Otherwise, it has not yet reached its final destination. These aspects of *put*, at least, can be captured using the machinery described here with the following definition.

```
(define put (x y)
 (exists (i j z)
  (and (during i (move (hand x)))
       (during i (contacts (hand x) y))
       (during i (attached (hand x) y))
       (during i (supports x y))
       (during i (move y))
       (during j (not (move y)))
       (during j (supported y))
       (during j (supports z y))
       (not (equal z (hand x)))
       (= (end i) (beginning j)))))
```

Similarly, the prototypical event described by *pick up* can expressed as essentially the inverse operation.

```
(define pick-up (x y)
 (exists (i j z)
  (and (during i (supported y))
       (during i (supports z y))
       (during i (contacts z y))
       (during j (move (hand x)))
       (during j (contacts (hand x) y))
       (during j (attached (hand x) y))
       (during j (supports x y))
       (during j (move y))
       (not (equal z (hand x)))
       (= (end i) (beginning j)))))
```

Many other simple spatial motion verbs also apparently involve support. Consider *carry* and *raise*. Miller (p. 355) defines these words as follows.

carry: **to cause to travel with self**
raise: **to cause to travel up**

Jackendoff (p. 184) defines *raise* as

$$\text{CAUSE}(x, \text{GO}(y, [_{\text{Path}}\ \text{UPWARD}, z])).$$

One would say *Larry Bird raised the ball into the basket* to describe a layup but not a jump shot even though he has caused upward motion of the basketball in either case. One must be continually supporting an object, perhaps indirectly, to be raising it. This holds true even more so for the verb *lift*. Likewise, one is not carrying a baby stroller when one is pushing or pulling it, even though one is causing it to travel with oneself.[2] The statement *Don't drag that box, carry it!* would be infelicitous if the prototypical carrying event admitted dragging. Accordingly, I offer the following alternate definitions for *carry* and *raise*.

```
(define carry (x y)
 (exists (i)
  (and (during i (move x))
       (during i (move y))
       (during i (supports x y)))))
```

[2]The *Halakhic* concept of הוצאה not withstanding.

```
(define raise (x y)
 (exists (i)
  (and (during i (supports x y))
       (during i (move-up y)))))
```

The verbs described so far highlight the need for support in their definition. Support is not the only crucial component of verb meaning. Contact and attachment also play a pivotal role. This is illustrated in the simple verbs *slide* and *roll*. Pinker (p. 182) offers the following representations for the intransitive use of *roll*.

The uninterpreted manner attribute offers no guidance as to the perceptual mechanisms needed to detect rolling and thus to define the meaning of the word *roll*. A proper definition of rolling can be based on a definition of sliding since rolling occurs when sliding doesn't. One object slides against another object if they are in continual contact and one point of one object contacts different points of the other object at different instants. Although the notion of one object sliding against another can be represented in the notation used here, by reducing it to primitives that return the points of contact between objects, I prefer instead to treat `slide-against` as a primitive notion much like support, contact, and attachment. I conjecture that the human visual apparatus contains innate machinery for detecting sliding motion and suggest that experiments like those performed by Freyd and Spelke, to be described in sections 8.3 and 9.5, could be used to determine the validity of this claim. Given the primitive notion `slide-against`, one could then define the intransitive verb *slide* as follows.

```
(define slide (x) (exists (i y) (during i (slide-against x y))))
```

Rolling motion can then be described as occurring in any situation where an object is rotating while it is in contact with another object without sliding against that object.

```
(define roll (x)
 (exists (i y)
  (and (during i (not (slide-against x y)))
       (during i (rotate x))
       (during i (contacts x y)))))
```

Accurately representing the transitive uses of *slide* and *roll*, however, requires the notion of causality. Since this thesis does not offer a theory for grounding the perception of causality, I will not attempt to formulate definitions for these transitive uses. It is interesting to note, however, that despite this inability for describing causality, many verbs described so far are nonetheless causal verbs. They can be described fairly accurately without recourse to causality due to the availability of other cues such as support, contact, and attachment.

So far, the primary use of the notion of attachment has been to describe grasping. Levin (1985, 1987) suggests that there is an entire class of verbs of attachment including *attach*, *fasten*, *bolt*, *glue*, *nail*, *staple*, ... I want to suggest another potential role attachment might play in verb meaning beyond the class of these kind of attachment verbs. Two other verb classes suggested by Levin include verbs of creation and verbs of destruction. The typical way of representing such verbs is via a change in the state of existence of some object. Thus Schank proposes the following definitions for *make* and *break*.

X *make* Z:

$$\begin{array}{c} X \Longleftrightarrow DO \\ \Uparrow r \\ Z \overset{ts}{\Longleftrightarrow} BE \end{array}$$

X *break*:

$$\begin{array}{c} X \Longleftrightarrow DO \\ \Uparrow r \\ Z \Longleftrightarrow broken \end{array}$$

To the same end, Jackendoff proposes the existential field and primitives like GO_{Exist} and [EX]. Similarly, Pinker offers the following definitions for *make* (p. 223)[3]

make:

and *break* (p. 206).

break:

Like uninterpreted manner attributes, a symbol like [EX] offers little guidance in grounding the concepts of creation and destruction. While I do not suggest that we are anywhere close to being able to fully ground these concepts, the notion of attachment may allow a modest start in the right direction. Objects are constructed from components that are typically attached to each other to form the aggregate parent object. One can make an object by forming attachments between appropriate components. One can break an object by severing those attachments. Chapter 8 describes how ABIGAIL models objects as collections of attached line segments and circles. Attachments between line segments and circles can be made and broken during the course of the movie. ABIGAIL can track the formation and dissolution of attachment relationships dynamically during event perception. This is how ABIGAIL can detect graspings and releasings. This same mechanism can be used to determine that a new object has been constructed

[3] I have omitted the benefactive component of Pinker's original definition as it is tangential to the current discussion. Pinker also phrased the original definition as a gloss for the utterance *Bob made a hat*. I have replaced the tokens (Bob) and (hat) from the original gloss with the variables X and Y.

out of some components, or that an object has been broken into its pieces. Such low-level notions may form the basis of more complete explanations for creation and destruction by way of a long chain of analogical reasoning. Whether such speculation leads anywhere remains for future research.

As a final example, I will present the definition of a verb that is seemingly perceptually much more complex. Schank gives the following definition for *walk*.

This definition, however, admits running, hopping, skipping, jumping, skating, and bicycling events. We can consider walking to involve a sequence of steps. Each step involves lifting up some foot off the ground and placing it back on the ground.

```
(define step (x)
 (exists (i j k y)
  (and (during i (contacts y ground))
       (during j (not (contacts y ground)))
       (during k (contacts y ground))
       (equal y (foot x))
       (= (end i) (beginning j))
       (= (end j) (beginning k)))))
```

In addition to stepping, walking involves motion. Furthermore, two conditions can be added to distinguish walking from running, hopping, skipping, and jumping on one hand, and skating on the other. One stipulates that at all times during walking, at least one foot must be on the ground. The second stipulates that no sliding takes place.

```
(define walk (x)
 (exists (i)
  (and (during i (repeat (step x)))
       (during i (move x))
       (during i
        (exists (y)
         (and (equal y (foot x))
              (contacts y ground))))
       (during i
        (not (exists (y)
              (and (equal y (foot x))
                   (slide-against y ground)))))))
```

Taken together, this is a fairly accurate description of walking.

All of the discussion so far has focussed on using semantic representations for event perception. The ultimate goal of this research, however, is to link language with perception using the architecture from

figure 6.2. For a semantic representation to act as an appropriate bridge between the linguistic and non-linguistic halves of this architecture, it must simultaneously meet criteria imposed by both halves. Linguistic processing imposes a strong constraint not addressed so far. It must be possible to specify a way for combining representations of the meanings of words to form the representation of the meaning of an utterance comprising those words. Such a process in called a *linking rule*. The choice of linking rule depends on the representation used. A linking rule appropriate for one representation might not be suitable for another. Jackendoff, Pinker, and Dorr (1990a, 1990b) adopt a substitution-based linking rule. With this rule, word meanings are taken to be expressions with variables acting as place holders for a word's arguments. The meaning of a phrase is composed by taking some constituent in that phrase as the head and substituting the meanings of the remaining constituents for variables in the head's meaning. Figure 7.2, illustrates an example application of this linking rule. This rule can be thought of simply as β-substitution, one of the rewrite rules introduced as part of the λ-calculus. While such a linking rule is suitable for Jackendovian representations and its derivatives used by Pinker and Dorr, it is unsuitable for the representation proposed here. This can be illustrated by the following example. Consider the utterance *John dropped the book on the floor*. For simplicity, let's take the meanings of *John*, *the book*, and *the floor* to be `john`, `book`, and `floor` respectively. Earlier, I took the meaning of *drop* to be as follows.

```
(define drop (x y)
 (exists (i j)
  (and (during i (contacts (hand x) y))
       (during i (attached (hand x) y))
       (during i (supports x y))
       (during i (supported y))
       (during j (not (contacts (hand x) y)))
       (during j (not (attached (hand x) y)))
       (during j (not (supports x y)))
       (during j (not (supported y)))
       (during j (move-down y))
       (= (end i) (beginning j)))))
```

While one could apply simple substitution to link `john` with x and `book` with y, that technique will not work with the prepositional phrase *on the floor* in the above utterance. The desired expression to represent the meaning of the entire utterance would look something like the following.

```
(exists (i j K)
 (and (during i (contacts (hand john) book))
      (during i (attached (hand john) book))
      (during i (supports john book))
      (during i (supported book))
      (during j (not (contacts (hand john) book)))
      (during j (not (attached (hand john) book)))
      (during j (not (supports john book)))
      (during j (not (supported book)))
      (during j (move-down book))
      (DURING K (CONTACTS BOOK FLOOR))
      (DURING K (SUPPORTS FLOOR BOOK))
      (DURING K (SUPPORTED BOOK))
      (= (end i) (beginning j))
      (= (END J) (BEGINNING K))))
```

Figure 7.2: A derivation of the meaning of the utterance *John slid the cup from Mary to Bill* from the meanings of its constituent words using the linking rule proposed by Jackendoff.

While it is unclear what to take precisely as the meaning of the preposition *on*, what it does structurally in the above example is contribute a new interval k to the existential quantifier, some added conjuncts describing support and contact relationships between the book and the floor, and an added conjunct to temporally constrain the new interval relative to prior intervals. These additions appear in upper case in the above semantic representation. Whatever we take as the meaning of *on the floor*, it is not a piece of structure that is substituted for a single variable in some other structure. Furthermore, the new structure contributed by *on the floor* must itself have variables which are linked to elements such as `book` from the structure to which it is linked. Thus substitution-based linking rules are not suitable for the type of representation discussed here.

There is much talk in the linguistic literature about linking rules which are claimed to be innate and universal (cf. Pinker 1989). Such claims can be valid only if the actual semantic representation used by the brain is of the form that allows such linking rules to apply. These claims must be revised if it turns out that the semantic representation must be more like that discussed here. Consider the following example. A common claim is that a universal linking rule stipulates that agents are subjects. An additional claim is that the first argument to the CAUSE primitive is an agent (cf. Jackendoff 1990). Using extensions that will be described in section 10.2, the primitive notion `(supports` x y`)` can be viewed as something like `(cause` x `(supported` y`))`. In this case, x would be an agent and thus would be a subject. Consider however, the utterance *John leaned on the pole*. In the representation considered here, this would correspond to `(supports pole john)`, or equivalently `(cause pole (supported john))`. This would require *pole* to be an agent and thus a subject, contrary to English usage. Thus the claimed universal linking rule and the semantic representation considered here are incompatible. The universal linking rule can be valid only if we find a compatible representation which also allows grounding meaning in perception.

Borchardt (1984) recognizes the need to incorporate the notions of support, contact, and attachment into procedures for recognizing simple spatial motion events. He describes a system that recognizes such events in a simulated micro-world containing a robot hand and several objects. That system receives the changing coordinates of those objects as input. Figure 7.3 illustrates several event recognition procedures

suggested by Borchardt for that micro-world. While his definitions and notation differ in specific details from the definitions and notation suggested here, we share the same intent of describing spatial motion events using the notions of support, contact, and attachment. The major difference is that Borchardt's system receives the changing support, contact, and attachment relationships between objects as input, while ABIGAIL infers such relationships from lower-level perceptual input.

To summarize, this chapter has advanced the claim that the notions of support, contact, and attachment play a central role in defining the meanings of numerous simple spatial motion verbs. These notions are necessary to construct procedures which can differentiate between occurrences and non-occurrences of prototypical events which these verbs describe. I have shown how prior lexical semantic representations lack the ability for representing these notions, and are thus incapable of making the requisite distinctions. Furthermore, I have proposed an alternate representation which not only incorporates these notions into verb definitions, but does so in a prominent fashion. This new representation is useful only if one can show how to ground the notions of support, contact, and attachment in visual perception. The next two chapters will propose a theory of how such grounding may work.

```
(defun slide (a b)                                                    p. 99
 (and (dsupport table a)
      (translate a)
      (not (roll a))))

(defun roll (a b)                                                     p. 99
 (and (dsupport a)
      (dsupport a)
      (translate a)
      (or (isa a ball)
          (and (isa a cylinder)
               (perpendicular i (heading a i) (orientation a p i))))))

(defun fall (a)                                                       p. 99
 (and (< (ddt (position a z)) -10)
      (not (exists i hand (control i a)))))

(defun bounce (a b)                                                   p. 105
 (and (moveaway a b)
      (hit a b justbefore (start (moveaway a b)))
      (< (abs (ddt (velocity b))) 3)))

(defun control (a b)                                                  p. 108
 (and (not (dsupport table b))
      (or (hold a b)
          (support a b)
          (exists i object (and (hold a i) (support i b))))))

(defun raise (a b)                                                    p. 108
 (and (control a b) (< (ddt (position b z)) -0.5)))

(defun pickup (a b)                                                   p. 110
 (and (movefingers a)
      (not (control a b))
      (at (ever (control a b)
                (start (and (movefingers a) (not (control a b)))))
          (next (stop (movefingers a))))))

(defun setdown (a b)                                                  p. 110
 (and (movefingers a)
      (control a b)
      (at (ever (not (control a b))
                (start (and (movefingers a) (control a b))))
          (next (stop (movefingers a))))))

(defun drop (a b)                                                     p. 110
 (and (fall b) (justbefore (control a b) (start (fall b)))))
```

Figure 7.3: A selection of representations of verbs used by Borchardt to detect occurrences of events described by those verbs in a simulated blocks world with a robot arm. The page numbers indicate where the representation appeared in Borchardt (1984).

Chapter 8

Event Perception

In chapter 7, I argued that the notions of support, contact, and attachment play a central role in defining the meanings of numerous spatial motion verbs. If this is true, the ability to perceive occurrences of events described by those verbs rests on the ability to perceive these support, contact, and attachment relations. In this chapter I advance a theory of how this might be accomplished. The central claim of this chapter is that support, contact, and attachment relations can be recovered using *counterfactual simulation*, imagining the short-term future of a potentially modified image under the effects of gravity and other physical forces. For instance, one determines that an object is unsupported if one imagines it falling. Likewise, one determines that an object A supports an object B if B is supported, but falls when one imagines a world without A. An object A is attached to another object B if one must hypothesize such an attachment to explain the fact that one object supports the other. A similar, though slightly more complex, mechanism is used to detect contact relationships. All of the mechanisms rely on a modular *imagination capacity*. This capacity takes the representation of a possibly modified image as input, and predicts the short-term consequences of such modifications, determining whether some predicate P holds in any of the series of images depicting the short-term future. The imagination capacity is modular in the sense that the same unaltered mechanism is used for a variety of purposes, varying only the predicate P and the initial image model between calls. To predict the future, the imagination capacity embodies physical knowledge of how objects behave under the influence of physical forces such as gravity. For reasons to be discussed in chapter 9, such knowledge is naive and yields predictions that differ substantially from those that accurate physical modeling would produce. Section 10.2 speculates about how the imagination capacity might also contain naive psychological knowledge modeling the mental state of agents in the world, and how such knowledge might form the basis of the perception of causality. Chapter 9 discusses the details of the mechanism behind the imagination capacity. This chapter first presents a computational model of how such a capacity can be used to perceive support, contact, and attachment relations, as well as experimental evidence that suggests that such mechanisms might form the basis of human perception of these notions.

Certain notions seem to pervade human perception of the world. We know that solid objects cannot pass through one another. This has been termed the *substantiality* constraint. We know that objects do not disappear and then later reappear elsewhere. When an object moves from one location to another, it follows a continuous path between those two locations. This has been termed the *continuity* constraint. We know that unsupported objects fall and that the ground acts as universal support for all objects. I will refer to these latter two facets of human perception as *gravity* and *ground plane*. Section 9.5 will review experiments performed by Spelke (1988) and her colleagues that give evidence that at least two of the above notions are present in humans from very early infancy, namely substantiality and continuity. This chapter, along with chapter 9, argues that substantiality, continuity, gravity, and ground plane are

central notions that govern the operation of an imagination capacity which is used to recover support, contact, and attachment relations from visual input. Recovery of these relations in turn, forms the basis of event perception and the grounding of language in perception.

8.1 The Ontology of Abigail's Micro-World

Before presenting the details of a computational model of event perception, it is necessary to describe the ontology which ABIGAIL uses to interpret the images she is given as input.

The real world behaves according to the laws of physics. Beyond these laws, people project an ontology onto the world. It may be a matter of debate as to which facets of our perceived world should be attributed to physics, and which to our conceptualization of it, but such philosophical questions do not concern us here. In either case, our world contains, among other things, solid objects. These objects have mass. They are located and oriented in three-dimensional cartesian space. Solid objects obey the principles of substantiality, continuity, gravity, and ground plane, that is, solid objects do not pass through one another, they follow a continuous path through space when moving between two points, they fall unless they are supported, and they are universally supported by the ground. Subject to these constraints (and perhaps others), solid objects can change their position and orientation, they can touch one another, they can be fastened to one another, they can be broken into pieces, and those pieces eventually refastened to form either the same object, or different objects. Complex objects can be constructed out of parts which have been fastened together. The relative motion of such parts can be constrained to greater or lesser degrees.

The aforementioned story is a small but important fragment of human world ontology. On this view, we all share roughly the same conceptual framework, around which much of language is structured. The non-metaphoric meanings of many simple spatial motion verbs depend on this shared ontology. For example, the verb *sit* incorporates, among other things, the notion of support, which in turn is built on the notions of gravity and substantiality. But this alone does not suffice. *Sit* also incorporates the notion that our body has limbs as parts, that these limbs are joined to our torso, that these joints impose certain constraints on the relative motion of our body parts, and these constraints allow us to assume certain postures which facilitate the support of our body. Furthermore, many nouns such as *chair* derive at least part of their meaning from the role they play in events referred to by words like *sit*. So a chair must facilitate support of the body in the sitting posture. A little introspection will reveal that the aforementioned fragment is a necessary, and perhaps almost sufficient, ontology for describing numerous word meanings, including those discussed in chapter 7.

Like the real world, ABIGAIL's micro-world has an ontology, though this ontology is derived mostly via projection of ABIGAIL's perceptual processes onto a world governed by very few physical laws. This ontology is analogous to that of the real world though it differs in some of the details. ABIGAIL's micro-world contains objects that have mass, and are located and oriented in a $2\frac{1}{2}$-dimensional cartesian space. These objects obey substantiality, continuity, gravity, and ground plane. They can move, touch, support, and be fastened to one another. They can break into pieces and those pieces refastened. The relative motion of pieces fastened together can be constrained so that an object constructed out of parts can have a posture which can potentially change over time. Most of the words discussed in chapter 7 can be interpreted relative to the alternate ontology of ABIGAIL's micro-world, rather than the real world. Such a re-interpretation maintains the general conceptual organization of the lexicon in that a person would use the same word *sit* to describe analogous events in the movie and the real word. Furthermore, the ontological analysis projected by ABIGAIL onto a sitting event in the movie is identical to the analysis projected by a person watching a sitting event in the real world, even though the low-level primitives out of which those analyses are constructed differ. This allows ABIGAIL's micro-world to act as a simplified though non-trivial testbed for exploring the relationship between language and perception.

The aforementioned ontology is not implemented in ABIGAIL as explicit declarative knowledge. Instead, it is embedded procedurally in an *imagination capacity* to be described in chapter 9. The event perception mechanisms described in this chapter, and ultimately any language processing component which these mechanisms drive, rely on this ontology through the imagination capacity. Although the ontology possessed by humans differs from this artificial ontology in its details, if the general framework for event perception incorporated into ABIGAIL is reflective of actual human event perception, then human event perception too must ultimately rely on a world ontology. I should stress that I remain agnostic on the issue of whether such an ontology—and the mechanisms for its use—are innate or acquired. Nothing in this thesis depends on the outcome of that debate. All that is assumed is that the ontology and mechanisms for its use are in place prior to the onset of any linguistic ability based on the link between linguistic and perceptual processes. A particular consequence of this assumption is the requirement that the ontology and mechanisms for its use be in place prior to the onset of language acquisition, since the models described in part I of this thesis rely on associating each input utterance with semantic information denoting the potential meanings of that utterance recovered from the non-linguistic context.

This ontology may be represented redundantly, and differently, at multiple cognitive levels. I find no reason to assume that this ontology is represented uniformly in the brain at a single cognitive level. The representation used for imagination, a low-level process, might differ from representations at higher levels. The ontology used for low-level imagination during visual perception may differ both in its implementation, as well as its predictive force, from any other ontology we possess, in particular that which we discover through introspection. Different ontologies may be acquired via different means at different times. Furthermore, it is plausible for some to be innate while others are acquired. To me, in fact, this seems to be the most likely scenario.

8.1.1 Figures

At the lowest level, the world that ABIGAIL perceives is constructed from *figures*. I will denote figures with the (possibly subscripted) symbols f and g. In the current implementation, figures have one of two *shapes*, namely *line segments* and *circles*. Conceivably, ABIGAIL could be extended to support additional shapes, such as conic section arcs and polynomial arcs, though the complexity of the implementation would grow substantially without increasing the conceptual coverage of the theory.[1]

At each movie frame ABIGAIL is provided with the position, orientation, shape, and size of every figure. Positions are points in the cartesian plane of the movie screen. I assume that the camera does not move. Thus an object is stationary if and only if the coordinates of the positions of its figures do not change. The (possibly subscripted) symbols p and q will denote points. Each point p has two coordinates, $x(p)$ and $y(p)$.

The *position* of a figure f is specified by two points, $p(f)$ and $q(f)$. For line segments, these are its two endpoints. For circles, $p(f)$ is its center while $q(f)$ is a point on its perimeter. The orientation and size of figures are derived from these points. Given two points, p and q, the orientation of the line from p to q in given by[2]

$$\theta(p,q) \triangleq \tan^{-1} \frac{y(q) - y(p)}{x(q) - x(p)}.$$

The *orientation* of a figure, whether it be a line segment or a circle, is an angle $\theta(f) \triangleq \theta(f)$.[3] Throughout the implementation of ABIGAIL, all angles θ, including the orientations of figures, are normalized so

[1] In retrospect, even allowing circles unduly complicated the implementation effort. Little would be lost by allowing only line segments, and modeling circles as polygons.

[2] Actually, the COMMON LISP function `(atan (- (y q) (y p)) (- (x q) (x p)))` is used to handle orientation in all four quadrants and the case where θ is $\frac{\pi}{2}$.

[3] This implies the somewhat unrealistic assumption that circles have a perceivable orientation. The reason for this simplification will be discussed on page 127.

that $-\pi < \theta \leq \pi$. Note that the leftward orientation is normalized to $+\pi$ and not $-\pi$. The reason for this will be discussed on page 165. Axes of translation will be specified as orientations. Given the orientation θ of an axis of translation, translation along the axis in the opposite direction is accomplished via a translation with the orientation $\theta+\pi$, suitably normalized. In a similar fashion, amounts of rotation about pivot points will be specified via angles. If θ denotes an amount of rotation in one direction then $-\theta$ denotes the amount of rotation in the opposite direction.

I will denote the distance between two points p and q as $\Delta(p,q)$.

$$\Delta(p,q) \triangleq \sqrt{(x(p)-x(q))^2+(y(p)-y(q))^2}$$

The *size* of a line segment is its length, the distance $\Delta(p(f),q(f))$ between its two endpoints. The size of a circle is its perimeter: $\pi\Delta(p(f),q(f))^2$. Figures also have a *mass*, denoted $m(f)$, which is taken to be equal to their size. Figures have a *center-of-mass*. The values $x(f)$ and $y(f)$ denote the coordinates of the center-of-mass of a figure f. The center-of-mass of a line segment is its midpoint.

$$\begin{aligned} x(f) &= \frac{x(p(f))+x(q(f))}{2} \\ y(f) &= \frac{y(p(f))+y(q(f))}{2} \end{aligned}$$

The center-of-mass of a circle is its center: $x(f)=x(p(f))$, $y(f)=y(p(f))$.

I also define the notion of the *displacement* between a point and a figure, denoted $\delta(p,f)$. This will play a role in defining joint parameters in the next section. If f is a line segment, then

$$\delta(p,f) \triangleq \frac{\Delta(p,p(f))}{\Delta(p(f),q(f))}.$$

Such a displacement is called a *translational displacement*. Since displacements are used only for points forming joints between figures, the point p will always lie on f and the displacement will always be between zero and one inclusively. If f is a circle, then $\delta(p,f) \triangleq \theta(p(f),p)-\theta(f)$. Such a displacement is called an *rotational displacement* and will always be normalized so that $-\pi < \delta(p,f) \leq \pi$.

8.1.2 Limitations and Simplifying Assumptions

At every movie frame, ABIGAIL is presented with a set $\mathcal{F}_i$ of figures that appear in frame i. Several simplifying assumptions are made with respect to the sets $\mathcal{F}_i$.

1. Each figure in every frame corresponds to exactly one figure in both the preceding and following frame.

2. ABIGAIL is given this correspondence.

3. The shape of each corresponding figure does not change from frame to frame.

4. ABIGAIL is given the correspondence between the endpoints of corresponding line segments in successive frames. In other words, ABIGAIL is given the distinction between a line segment whose endpoints are (p,q) and one whose endpoints are (q,p). This allows ABIGAIL to assign an unambiguous orientation to every line segment.

5. ABIGAIL can perceive two concentric equiradial circles as separate figures even though they overlap. ABIGAIL can also perceive two collinear intersecting line segments as separate figures. This means, for instance, that when a knee is straightened so that the thigh and calf are collinear, they are still perceived by ABIGAIL as distinct line segments even though they may be depicted graphically as a single line segment.

Collectively, these simplifying assumptions[4] imply that ABIGAIL need only maintain a single set $\mathcal{F}$ of figures invariant over time. Only the coordinates of the points of the figures can change from frame to frame. These assumptions also imply several restrictions on ABIGAIL's ontology. First, individual figures are never created, destroyed, split, fused, or bent. This is not a severe restriction since figures are only the atomic elements out of which objects are constructed. Objects, being sets of figures, can nonetheless be created, destroyed, split, fused, or bent by changing the attachment relationships between the figures constituting those objects. Second, figures cannot appear or disappear. They can never enter or leave the field of view and are never occluded. Since objects are composed of figures, this implies that objects, as well, never enter or leave the field of view. While from a very early age, infants possess the notion of object permanence, such a notion has not yet been incorporated into ABIGAIL. This severe restriction will not be addressed in this thesis. Finally, these assumptions imply that ABIGAIL is given the continued identity of objects over time.

Object perception can be broken down into three distinct tasks: segmentation, classification, and identification. *Segmentation* is the process of grouping figures together into objects. *Classification* is the process of assigning a type to an object based on its relation to similar objects. *Identification* is the process of tracking the identity of an object—determining that some object is the same as one previously seen. This thesis currently addresses only segmentation. The per-frame analysis discussed in section 8.2.1 is a novel approach to image segmentation based on naive physical knowledge. Extending this approach to address object classification and identification is an area left for future research.

It is possible to relax the assumptions that ABIGAIL be provided with the figure and endpoint correspondences (assumptions 2 and 4 from above), and have her recover such correspondences herself, provided that such correspondences do exist to be recovered and the remaining assumptions still hold. One way to extend ABIGAIL to recover the figure and endpoint correspondences would be to choose a matching that paired only objects of the same shape, and choose the matching that minimized the sum of the distances between the points of the paired figures. If the frame rate is high enough relative to object velocities, a simple greedy optimization algorithm, perhaps with some hillclimbing, should suffice. This approach would be a simple first step at addressing object identification. It has not been attempted since it is tangential to the main focus of this work.

Many of ABIGAIL's perceptual mechanisms are phrased in terms of the notions intersect, touch, and overlap. Two figures *intersect* if they share a common point. Two line segments *touch* if they intersect at a single point and that intersection point is coincident with an endpoint of one of the line segments. Two circles touch if they intersect at a single point. A line segment and circle touch either if the line segment is tangent to the circle, or one of the two possible intersection points is coincident with an endpoint of the line segment. Two figures *overlap* if they intersect but do not touch, except that a line segment and a circle can both overlap and touch if one intersection point is coincident with an endpoint of the line segment while the other is not. Figure 8.1 gives a pictorial depiction of these notions and enumerates the different possible relations between two figures. The left hand column depicts the possible relations between two line segments. The center column depicts the possible relations between a line segment and a circle. The right hand column depicts the possible relations between two circles. Cases (a) through (h) depict touching relations. Cases (i) through (k) depict overlap relations. Case (l) depicts the only instance where two figures can both touch and overlap simultaneously.

For reasons which will be discussed is section 9.3.4, these notions of intersect, touch, and overlap must be made 'fuzzy'. In this fuzzy definition of intersection, two figures intersect if the closest distance between a point on one and a point on the other is within some tolerance. The midpoint between those two closest points is taken to be the intersection point for determining the touch and overlap relations if the two figures do not actually intersect. Finally, two points are taken to be coincident if the distance between them is within some tolerance.

[4]For efficiency reasons, the current implementation of ABIGAIL adds the additional assumption that the size of corresponding figures is invariant across frames though this assumption is not fundamental and easily lifted.

Figure 8.1: The possible ways in which two figures can touch or overlap. Cases (a) through (h) depict instances of touching. Cases (i) through (k) depict instances of overlapping. Case (l) depicts the one instance which can involve both touching and overlapping between the same two figures.

These are not meant to be taken as definitions of the words *intersect*, *touch*, and *overlap*. Rather they are low-level perceptual notions out of which higher-level definitions of these words, and others, can be constructed.

8.1.3 Joints

Part of ABIGAIL's ontology is the knowledge that figures can be joined, fastened, or attached together. A *joint* is a constraint that two figures intersect. I will denote a joint with the (possibly subscripted) symbol j. The two figures joined by a joint j are denoted $f(j)$ and $g(j)$.

Joints can optionally further constrain the relative motion between two figures. Since each figure has three degrees of freedom (the (x, y) position of one endpoint and its orientation), a joint can potentially constraint each of these three degrees of freedom of one figure relative to another it is joined to. Thus a joint may specify three parameters, each of which independently constrains one of the degrees of freedom. Each of these parameters may be either real-valued or `nil`. A `nil` value for a parameter signifies that a joint is *flexible* along that degree of freedom, while a real value specifies that it is *rigid*. Joints can be independently rigid or flexible along each degree of freedom. A rigid *rotation* parameter $\theta(j)$ constrains the angle between the orientations of the two joined figures to be equal to the parameter setting: $\theta(j) = \theta(g(j)) - \theta(f(j))$. The remaining two joint parameters are the *displacement* parameters $\delta_f(j)$ and $\delta_g(j)$ which partially constrain the displacement of the intersection point relative to each figure. Since the two figures of a joint must intersect, one can denote their intersection point as $p(j)$. If $\delta_f(j)$ is rigid then the constraint $\delta_f(j) = \delta(p(j), f(j))$ is enforced. Likewise, if $\delta_g(j)$ is rigid then the constraint $\delta_g(j) = \delta(p(j), g(j))$ is enforced.[5] Note that giving circles orientations allows defining the concept of rotational displacement. Without such a concept, fixing the relative positions of two joints, each joining a different line segment to the same circle, would require a complex constraint specification between all three figures. With the notion of rotational displacement, the displacement of each line segment relative to the circle can be fixed independently as a constraint between two figures.

Since two figures may have more than one intersection point, I add an additional simplifying assumption about joints to allow unambiguous determination of the intersection point $p(j)$. I require that at least one of the displacement parameters of each joint be rigid. Subject to this constraint, the intersection point can be found by using whichever of the following formulas is applicable. If $\delta_f(j)$ is rigid and $f(j)$ is a line segment then

$$\begin{aligned} x(p(j)) &\triangleq x(p(f(j))) + \delta_f(j) \times (x(q(f(j))) - x(p(f(j)))) \\ y(p(j)) &\triangleq y(p(f(j))) + \delta_f(j) \times (y(q(f(j))) - y(p(f(j)))). \end{aligned}$$

If $\delta_f(j)$ is rigid and $f(j)$ is a circle then

$$\begin{aligned} x(p(j)) &\triangleq x(p(f(j))) + \Delta(p(f(j)), q(f(j))) \cos(\delta_f(j) + \theta(f(j))) \\ y(p(j)) &\triangleq y(p(f(j))) + \Delta(p(f(j)), q(f(j))) \sin(\delta_f(j) + \theta(f(j))). \end{aligned}$$

If $\delta_g(j)$ is rigid and $g(j)$ is a line segment then

$$\begin{aligned} x(p(j)) &\triangleq x(p(g(j))) + \delta_g(j) \times (x(q(g(j))) - x(p(g(j)))) \\ y(p(j)) &\triangleq y(p(g(j))) + \delta_g(j) \times (y(q(g(j))) - y(p(g(j)))). \end{aligned}$$

[5] Due to roundoff problems, a fuzzy notion of equality must be used to enforce joint parameters. The fuzzy comparison of angles must take normalization into account. This requires equating $-\pi + \epsilon$ to $\pi - \epsilon$.

If $\delta_g(j)$ is rigid and $g(j)$ is a circle then

$$\begin{aligned} x(p(j)) &\triangleq x(p(g(j))) + \Delta(p(g(j)), q(g(j)))\cos(\delta_g(j) + \theta(g(j))) \\ y(p(j)) &\triangleq y(p(g(j))) + \Delta(p(g(j)), q(g(j)))\sin(\delta_g(j) + \theta(g(j))). \end{aligned}$$

As part of her pre-linguistic endowment ABIGAIL knows that figures can be fastened by joints and that joints have the aforementioned properties. Furthermore, she knows how these properties affect the motion of joined figures under the effects of gravity and related naive physical constraints. This knowledge is embodied in an imagination capacity which will be discussed in chapter 9. However, her perceptual processes do not allow her to directly perceive the existence of joints in the movie she is watching. As perceptual input, she is given only the positions, orientations, shapes, and sizes of figures in each movie frame. She is not told which figures are joined and how they are joined. She must infer this information from the image figure data alone. Furthermore, which figures are joined and the parameters of those joints may change over time. Joints may be broken, as happens when a leg is removed from the table. New joints may be formed, as would happen if a table was built by attaching its legs to the table top. Rigid joint parameters may become flexible and flexible joint parameters may become rigid. At all times ABIGAIL maintains a *joint model*, a set of joints J and their parameters, that she currently believes to reflect what is happening in the movie. The process by which she updates this joint model will be described in section 8.2.1.

8.1.4 Layers

ABIGAIL's micro-world is nominally two-dimensional. The movie input has only x and y coordinates. A two-dimensional world, however, is very constraining. If one wants to model the substantiality constraint in such a world, the movement of objects world be severely restricted. For instance, in the movie described in section 6.1, John would not be able to walk, as he does, from one side of the table to the other, for in doing so, he would violate substantiality. People, have no difficulty understanding that movie even though they too, perceive only a two-dimensional image. That is because human world ontology is three-dimensional and human perception understands two-dimensional depictions of a three-dimensional world. So a human watching the movie described in section 6.1 would assume that John walked either in front of the table, or behind it, as he passed from one side to the other.

I want to be able to model such a capacity in ABIGAIL as well. Thus part of ABIGAIL's pre-linguistic endowment is the knowledge that each figure in the world resides on some *layer*. Two figures may either be on the same layer or on different layers. I will denote the fact that two figures f and g are on the same layer by the assertion $f \bowtie g$, and the fact that they are on different layers by the assertion $f \not\bowtie g$. These *layer assertions* affect whether the substantiality constraint holds between a pair of figures. Two figures which are on the same layer must not overlap. The substantiality constraint does not apply to figures on different layers.

Just like for joints, ABIGAIL is not given layer assertions as direct input. She must infer which figures are on the same layer, and which are on different layers, solely from image figure data. Again, much in the same way that joint parameters change during the course of a movie, figures can move from layer to layer as the movie progresses. Thus which layer assertions are true may change over time. ABIGAIL maintains a *layer model* which consists of a set L of layer assertions that reflects her current understanding of the movie. The process by which she updates this layer model will be discussed in section 8.2.1.

ABIGAIL treats layer assertions as an equivalence relation. The $\bowtie$ relation embodied in L is thus reflexive, symmetric, and transitive. The layer model must also be consistent. It cannot imply that two figures be both on the same layer, and on different layers, simultaneously. Furthermore, if the layer model neither implies that two figures are on the same layer nor that they are on different layers,

ABIGAIL will assume that they are on different layers by default. Layer assertions are a weak form of information about the third dimension. In particular, there is no notion of one figure being in front of or behind another figure, nor is there a notion of two figures being on adjacent layers. No further knowledge implied by our intuitive notion of 'layer' is modeled beyond layer equivalence.

8.2 Perceptual Processes

Having presented the ontology which ABIGAIL projects onto the world, it is now possible to describe the process by which she perceives support, contact, and attachment relations between objects in the movie. Recall that ABIGAIL has no prior knowledge about the types or delineation of objects in the world. She interprets any set of figures connected by joints as an object. To do so, she must know which figures are joined. Not being given that information as input, her first task is to form a model of the image that describes which figures are joined. Since the attachment status of figures may change from frame to frame as the movie unfolds, she must repeat the analysis which derives the joint model as part of the processing for each new frame. The ontology which ABIGAIL projects onto an image includes a layer model in addition to a joint model. Since ABIGAIL is given only two-dimensional information as input, she must infer information about the third dimension in the form of layer assertions in the layer model. Again, since figures can move from layer to layer during the course of the movie, ABIGAIL must update both the layer and joint models on a per-frame basis. Thus ABIGAIL performs two stages of processing for each frame. In the first stage she updates the joint and layer models for the image. The derived joint model delineates the objects which appear in the image. In the second stage she uses the derived joint and layer models to recover support, contact, and attachment relations between the perceived objects. The architecture used by ABIGAIL to process each movie frame is depicted in figure 8.2. The architecture takes as input, the positions, orientations, shapes, and sizes of the figures constituting the image, along with a joint and layer model for the image. The architecture updates this joint and layer model, groups the figures into objects, and recovers support, contact, and attachment relations between those objects. Central to the event perception architecture is an imagination capacity which encodes naive physical knowledge such as the substantiality, continuity, gravity, and ground plane constraints.

8.2.1 Deriving the Joint and Layer Models

As ABIGAIL watches the movie, she continually maintains both a joint model J and a layer model L. At the start of the movie, these models are empty, containing no joints and no layer assertions. After each frame of the movie, ABIGAIL looks for evidence in the most recent frame that the joint and layer models should be changed. Most of the evidence requires that ABIGAIL hypothesize potential changes and then imagine the effect of these changes on the world. ABIGAIL assumes that the world is for the most part stable. Objects are typically supported. She considers an unstable world with unsupported objects to be less likely than a stable one. If the world is unstable when imagined without making the hypothesized changes, then these hypothesized changes are adopted as permanent changes to the joint and layer models. This facet of ABIGAIL's perceptual mechanism is not justified by any experimental evidence from human perception but simply appears to work well in practice.

ABIGAIL's preference for a stable world requires that, to the extent possible, all objects be supported. There are two ways to prevent an object from falling. One is for it to be joined to some other supported figure. The other is for it to be supported by another figure. One figure can support another figure only if they are on the same layer, since support happens as a consequence of the need to avoid substantiality violations and substantiality holds only between two figures on the same layer.

ABIGAIL's imagination capacity is embodied in a kinematic simulator. This simulator can predict how a set of figures will behave under the effect of gravity, given particular joint and layer models, such

Figure 8.2: The event perception architecture incorporated into ABIGAIL. The architecture takes as input, the positions, orientations, shapes, and sizes of the figures constituting the image, along with a joint and layer model for the image. The architecture updates this joint and layer model, groups the figures into objects, and recovers support, contact, and attachment relations between those objects. Central to the event perception architecture is an imagination capacity which encodes naive physical knowledge such as the substantiality, continuity, gravity, and ground plane constraints.

that naive physical constraints such as substantiality are upheld. This imagination capacity, denoted as $I(\mathcal{F}, J, L)$ will be described in detail in chapter 9. The processes described here treat this capacity as modular. Any simulation mechanism that accurately models gravity and substantiality will do. The event perception processes simply call $I(\mathcal{F}, J, L)$ with different values of $\mathcal{F}$, J, and L, asking different questions of the predicted future, in the process of updating the joint and layer models and recovering support relations.[6]

ABIGAIL can change the joint and layers models in six different ways to keep those models synchronized with the world. She can

- add a layer assertion to L,
- remove a layer assertion from L,
- add a joint to J,
- remove a joint from J,
- promote a parameter of some joint $j \in J$ from flexible to rigid,
- demote a parameter of some joint $j \in J$ from rigid to flexible,

or perform any simultaneous combination of the above changes. Each type of change is motivated by particular evidence in the most recent movie frame, potentially mediated by the imagination process.

ABIGAIL makes three types of changes to the layer model on the basis of evidence gained from watching each movie frame. The process can be stated informally as follows. She will add an assertion that two figures are on different layers whenever they overlap, since if they were not on different layers, substantiality would be violated. She will add an assertion that two figures are on the same layer whenever one of the figures must support the other in order to preserve the stability of the image. Finally, whenever newer layer assertions contradict older layer assertions, the older ones are removed from the layer model giving preference to newer evidence. For example, when presented with the image from figure 6.1, ABIGAIL will infer that the ball and the table top are on the same layer since the ball would fall if it was not supported by the table top.

The process of updating the layer model can be stated more precisely as follows. A layer model consists of an ordered set L of layer assertions. Initially, at the start of the movie, this set is empty. The *closure* of a layer model is the layer model augmented with all of the layer assertions entailed by the equality axioms. A layer model is *consistent* if its closure does not simultaneously imply that two figures are on the same, as well as different, layers. ABIGAIL never replaces the layer model with its closure. She always maintains the distinction between layer assertions that have been added to the model as a result of direct evidence, in contrast to those which have been derived by closure. A *maximal consistent subset* of a layer model L is a consistent subset L' of L such that any other subset L'' of L that is a superset of L' is inconsistent. The *lexicographic maximal consistent subset* of a layer model L is the particular maximal consistent subset of L returned by the following procedure.

```
procedure MAXIMAL CONSISTENT SUBSET(L)
    L'←{};
    for a ∈ L
    do if L' ∪ {a} is consistent
        then L'←L' ∪ {a} fi od;
    return L' end
```

[6] As discussed in chapter 9, the imagination capacity $I(\mathcal{F}, J, L, P)$ takes a predicate P as its fourth parameter. In informal presentations, it is simpler to omit this parameter and use the English gloss 'P occurs during $I(\mathcal{F}, J, L)$' in place of $I(\mathcal{F}, J, L, P)$.

The above procedure may not find the largest possible maximal consistent subset. That problem has been shown to be NP-hard by Wolfram (1986). Using the above heuristic has proven adequate in practice.

Given the above procedure we can now define the process used to update the layer model. We define $L_{\not\bowtie}$ to be the set of all different-layer assertions $f \not\bowtie g$, where f and g overlap in the most recent movie frame. These are layer assertions which must be added to the layer model in order not to violate substantiality. We define $L_{\bowtie}$ to be the set of all same-layer assertions $f \bowtie g$, where f and g touch in the most recent movie frame. These are *hypothesized* layer assertions which could potentially account for support relationships needed to preserve stability. $L_{\bowtie}$ contains assertions only between figures which touch since only such assertions could potentially contribute to support relationships. The layer model updating procedure makes permanent only those hypothesized same-layer assertions that actually do prevent figures from falling under imagination. The layer model updating procedure is as follows.[7]

procedure UPDATE LAYER MODEL
  for $f \bowtie g \in L_{\bowtie}$
  do **if** neither $\hat{f}$ nor $\hat{g}$ move during
    $I(\mathcal{F}, J,$ MAXIMAL CONSISTENT SUBSET$(L_{\not\bowtie} \cup (L_{\bowtie} - \{f \bowtie g\}) \cup L))$
    then $L_{\bowtie} \leftarrow L_{\bowtie} - \{f \bowtie g\}$ **fi** **od**;
  $L \leftarrow$ MAXIMAL CONSISTENT SUBSET$(L_{\not\bowtie} \cup L_{\bowtie} \cup L)$ **end**

The process of updating the joint model is conceptually very similar to updating the layer model. The algorithm is illustrated in figure 8.3. First, remove all joints j from J where $f(j)$ does not intersect $g(j)$ in the most recent frame (lines 2 and 3). Second, demote any rigid parameter of any joint $j \in J$ when the constraint implied by that parameter is violated (lines 4 through 9). Third, remove all joints j from J where both $\delta_f(j)$ and $\delta_g(j)$ are flexible (lines 10 and 11). This is to enforce the constraint from page 127 that every joint have at least one rigid displacement parameter. Fourth, find a minimal set of parameter promotions and new joints that preserve the stability of the image (lines 12 through 33). To do this we form the set J' of all joints j' where $f(j')$ intersects $g(j')$ in the most recent movie frame (lines 12 through 20). Those joints in J' which appear in J have their parameters initialized to the same values as their counterparts in J, while any new joints have their parameters initialized to be flexible. We then promote all of the flexible parameters in J' to have the rigid values that they have in the most recent movie frame. One by one we temporarily demote each of the parameters just promoted and imagine the world (lines 21 through 33). If when demoting a parameter of a joint j', the constraint specified by the original rigid parameter is not violated during the imagined outcome of that demotion, then that demotion is preserved. Otherwise, the parameter is promoted back to the rigid value it has in the most recent movie frame. After trying to demote each of the newly promoted joint parameters, remove all joints j' from J' where both $\delta_f(j')$ and $\delta_g(j')$ are flexible (lines 34 and 35) and replace J with J' (line 36).[8]

Recall that an object can be supported in two ways, either by being joined to another object or by resting on top of another object on the same layer. ABIGAIL gives preference to the latter explanation. Whenever the stability of an image can be explained by hypothesizing either a joint between two figures or a same-layer assertion between those two figures, the same-layer assertion will be preferred. Thus for the image in figure 6.1, ABIGAIL infers that the ball is resting on top of the table, by virtue of the fact that they are on the same layer, and not attached to the side of the table. If ABIGAIL did not maintain

[7] The notation $\hat{f}$ used here and in figure 8.3 is described on page 160.

[8] Only a simplified version of this algorithm is currently implemented. First, the implemented version does not consider promoting existing flexible joints to explain the stability of an image. Only newly created rigid joints can offer such support. Second, newly added joints are always rigid. They are demoted to be flexible only when they move. Thus rather than finding a minimal set of promotions which make the image stable, the current implementation finds a minimal set of new rigid joints to stabilize the image.

```
procedure UPDATE JOINT MODEL
    for j ∈ J
    do if f(j) does not intersect g(j) then J←J − {j} fi od;
    for j ∈ J
    do if θ(j) ≠ nil ∧ θ(j) ≠ θ(g(j)) − θ(f(j)) then θ(j)←nil fi od;
    for j ∈ J
    do if δ_f(j) ≠ nil ∧ δ_f(j) ≠ δ(p(j), f(j)) then δ_f(j)←nil fi od;
    for j ∈ J
    do if δ_g(j) ≠ nil ∧ δ_g(j) ≠ δ(p(j), g(j)) then δ_g(j)←nil fi od;
    for j ∈ J
    do if δ_f(j) = nil ∧ δ_g(j) = nil then J←J − {j} fi od;
    J′←{};
    for f ∈ F
    do  for g ∈ F
        do  if f intersects g at p
            then j′ = f ↔ g;
                θ(j′)←θ(g) − θ(f);
                δ_f(j′)←δ(p, f);
                δ_g(j′)←δ(p, g);
                J′←J′ ∪ {j′} fi od od;
    for j′ ∈ J′
    do  j←nil;
        for j″ ∈ J
        do if f(j″) = f(j′) ∧ g(j″) = g(j′) then j←j″ fi od;
        θ←θ(j′); θ(j′)←nil;
        if (j ≠ nil ∧ θ(j) ≠ nil) ∨ θ̂(g(j′)) − θ̂(f(j′)) ≠ θ during I(F, J′, L)
        then θ(j′)←θ fi;
        δ_f←δ_f(j′); δ_f(j′)←nil; p←p(j′);
        if (j ≠ nil ∧ δ_f(j) ≠ nil) ∨ δ̂_f(p, f(j′)) ≠ δ_f during I(F, J′, L)
        then δ_f(j′)←δ_f fi;
        δ_g←δ_g(j′); δ_g(j′)←nil;
        if (j ≠ nil ∧ δ_g(j) ≠ nil) ∨ δ̂_g(p, g(j′)) ≠ δ_g during I(F, J′, L)
        then δ_g(j′)←δ_g fi od;
    for j′ ∈ J′
    do if δ_f(j′) = nil ∧ δ_g(j′) = nil then J′←J′ − {j′} fi od;
    J←J′ end
```

Figure 8.3: The algorithm for updating the joint model. ABIGAIL performs this procedure as part of her processing of each frame in the movie she watches.

this preference she would never form same-layer judgments, since any time a same-layer assertion can be used to provide support, a joint can be used as well. The fact that the converse is not true allows her to hypothesize joints when an object would slide off another object even if they were on the same layer.

The joint and layers models must be updated simultaneously by a tandem process rather than independently. If the joint model was updated before the layer model there would be no way to enforce the aforementioned preference for same-layer support over joint support. On the other hand, the layer model cannot be created before the joint model. When processing the first image, starting out with an empty joint model, ABIGAIL could not infer any layer information, since a layer model alone is insufficient to explain support. Without any joints, no set of layer assertions can improve the stability of an image. Thus the processes of updating the joint and layers models are interleaved, finding the least cost combination of same-layer assertions and joint promotions which improve the stability of the image. When computing the cost of such a combination, same-layer assertions have lower cost than promotions of existing joints, which in turn have lower cost than creation of new joints.

The method used by ABIGAIL to construct and update the joint and layer models is best illustrated by way of an example. The following example depicts the actual results generated by ABIGAIL when processing the first twelve frames of the movie described in section 6.1. Figure 8.4 shows these first twelve frames in greater detail. Since frame 0 is the first frame of the movie, ABIGAIL starts out processing this frame with empty joint and layer models. With empty models, the world is completely unstable and collapses into a pile of rubble when the short-term future is imagined. This is depicted by the imagination sequence given in figure 8.5. Accordingly, ABIGAIL hypothesizes the set of joints depicted in figure 8.6 and layer assertions depicted in figure 8.7. A joint is hypothesized between every pair of intersecting figures. A same-layer assertion is hypothesized between every pair of figures that touch. A different-layer assertion is hypothesized between every pair of overlapping figures. Not all of these joints and layer assertions are necessary to explain the stability of the image. By the process described above, ABIGAIL chooses to retain only the starred joints and layer assertions. With this new joint and layer model, the image is stable.[9]

Several things about the derived joint and layers models are worthy of discussion. First, note that the final layer model includes the following assertions[10]

$$\texttt{(circle ball)} \bowtie \texttt{(top table)}$$
$$\texttt{(bottom box)} \bowtie \texttt{(top table)}$$

indicating that ABIGAIL has determined that the ball and the bottom of the box are resting on the table rather than being joined to the table top. Second, the hem of Mary's dress need only be joined to one side of her dress, since one rigid joint is sufficient to support the line segment constituting the hem. Third, the image contains a number of locations where the endpoints of multiple line segments are coincident on the same point. Such a situation arises, for example, where John's legs meet his torso. In this situation, three joints are possible.

$$\texttt{(torso john)} \leftrightarrow \texttt{(right-thigh john)}$$
$$\texttt{(torso john)} \leftrightarrow \texttt{(left-thigh john)}$$
$$\texttt{(right-thigh john)} \leftrightarrow \texttt{(left-thigh john)}$$

All three of these joints are not necessary to achieve a stable image however. Any two of these joints are sufficient, since relative rigidity is transitive. ABIGAIL arbitrarily chooses the last two joints as the ones

[9] Except for the fact that John's and Mary's eyes fall out, since they appear unsupported. This highlights a deficiency in the ontology incorporated into ABIGAIL's perceptual mechanisms. I will not address this anomaly, and methods for dealing with it, in this thesis.

[10] In this and all further discussion, expressions such as `(circle ball)` denote particular figures. These figures are given names to aid in the interpretation of the results produced by ABIGAIL. ABIGAIL does not have access to these names during processing, so that fact that the names of several figures, i.e. `(circle ball)`, `(line-segment1 ball)`, etc. share the component `ball` in common, in no way assists ABIGAIL in her perceptual processing.

Figure 8.4: The first twelve frames of the movie depicted in figure 6.4. The script used to generate this movie is given in figure 6.3.

Figure 8.5: A subsequence of images produced by ABIGAIL while imagining the short-term future of frame 0 from the movie described in section 6.1 with empty joint and layer models.

```
    (hem mary) ↔ (dress2 mary)
*   (hem mary) ↔ (dress1 mary)
    (dress2 mary) ↔ (dress1 mary)
    (dress2 mary) ↔ (torso mary)
    (dress2 mary) ↔ (right-upper-arm mary)
*   (dress2 mary) ↔ (left-upper-arm mary)
    (dress1 mary) ↔ (torso mary)
    (dress1 mary) ↔ (right-upper-arm mary)
*   (dress1 mary) ↔ (left-upper-arm mary)
*   (mouth mary) ↔ (head mary)
*   (head mary) ↔ (torso mary)
    (torso mary) ↔ (right-thigh mary)
*   (torso mary) ↔ (left-thigh mary)
    (torso mary) ↔ (right-upper-arm mary)
*   (torso mary) ↔ (left-upper-arm mary)
*   (right-thigh mary) ↔ (left-thigh mary)
*   (right-thigh mary) ↔ (right-calf mary)
*   (left-thigh mary) ↔ (left-calf mary)
*   (right-upper-arm mary) ↔ (left-upper-arm mary)
*   (right-upper-arm mary) ↔ (right-fore-arm mary)
*   (left-upper-arm mary) ↔ (left-fore-arm mary)
*   (mouth john) ↔ (head john)
*   (head john) ↔ (torso john)
    (torso john) ↔ (right-thigh john)
*   (torso john) ↔ (left-thigh john)
    (torso john) ↔ (right-upper-arm john)
*   (torso john) ↔ (left-upper-arm john)
*   (right-thigh john) ↔ (left-thigh john)
*   (right-thigh john) ↔ (right-calf john)
*   (left-thigh john) ↔ (left-calf john)
*   (right-upper-arm john) ↔ (left-upper-arm john)
*   (right-upper-arm john) ↔ (right-fore-arm john)
*   (left-upper-arm john) ↔ (left-fore-arm john)
    (circle ball) ↔ (line-segment3 ball)
    (circle ball) ↔ (line-segment3 ball)
*   (circle ball) ↔ (line-segment2 ball)
    (circle ball) ↔ (line-segment2 ball)
*   (circle ball) ↔ (line-segment1 ball)
    (circle ball) ↔ (line-segment1 ball)
    (circle ball) ↔ (left-leg table)
    (circle ball) ↔ (top table)
    (circle ball) ↔ (top table)
    (bottom box) ↔ (right-wall box)
    (bottom box) ↔ (left-wall box)
    (bottom box) ↔ (right-leg table)
*   (right-wall box) ↔ (top table)
*   (left-wall box) ↔ (top table)
*   (seat chair2) ↔ (back chair2)
*   (seat chair2) ↔ (front chair2)
*   (seat chair1) ↔ (back chair1)
*   (seat chair1) ↔ (front chair1)
*   (right-leg table) ↔ (top table)
*   (left-leg table) ↔ (top table)
```

Figure 8.6: ABIGAIL hypothesizes these joints when processing frame 0 of the movie depicted in figure 8.4. Since not all of these joints are necessary to explain the stability of the image, ABIGAIL retains only the starred joints.

```
    (hem mary) ⋈ (dress2 mary)
    (hem mary) ⋈ (dress1 mary)
    (dress2 mary) ⋈ (dress1 mary)
    (dress2 mary) ⋈ (torso mary)
    (dress2 mary) ⋈ (right-upper-arm mary)
    (dress2 mary) ⋈ (left-upper-arm mary)
    (dress1 mary) ⋈ (torso mary)
    (dress1 mary) ⋈ (right-upper-arm mary)
    (dress1 mary) ⋈ (left-upper-arm mary)
    (mouth mary) ⋈ (head mary)
    (head mary) ⋈ (torso mary)
    (torso mary) ⋈ (right-thigh mary)
    (torso mary) ⋈ (left-thigh mary)
    (torso mary) ⋈ (right-upper-arm mary)
    (torso mary) ⋈ (left-upper-arm mary)
    (right-thigh mary) ⋈ (left-thigh mary)
    (right-upper-arm mary) ⋈ (left-upper-arm mary)
    (right-upper-arm mary) ⋈ (right-fore-arm mary)
    (mouth john) ⋈ (head john)
    (head john) ⋈ (torso john)
    (torso john) ⋈ (right-thigh john)
    (torso john) ⋈ (left-thigh john)
    (torso john) ⋈ (right-upper-arm john)
    (torso john) ⋈ (left-upper-arm john)
    (right-thigh john) ⋈ (left-thigh john)
    (right-thigh john) ⋈ (right-calf john)
    (left-thigh john) ⋈ (left-calf john)
    (right-upper-arm john) ⋈ (left-upper-arm john)
    (right-upper-arm john) ⋈ (right-fore-arm john)
*   (circle ball) ⋈ (line-segment3 ball)
    (circle ball) ⋈ (left-leg table)
*   (circle ball) ⋈ (top table)
    (bottom box) ⋈ (right-wall box)
    (bottom box) ⋈ (left-wall box)
    (bottom box) ⋈ (right-leg table)
*   (bottom box) ⋈ (top table)
    (right-wall box) ⋈ (top table)
    (left-wall box) ⋈ (top table)
    (seat chair2) ⋈ (back chair2)
    (seat chair2) ⋈ (front chair2)
    (seat chair1) ⋈ (back chair1)
    (seat chair1) ⋈ (front chair1)
    (right-leg table) ⋈ (top table)
    (left-leg table) ⋈ (top table)
*   (hem mary) ⋪ (right-calf mary)
*   (hem mary) ⋪ (left-calf mary)
*   (dress1 mary) ⋪ (left-fore-arm mary)
*   (torso mary) ⋪ (left-fore-arm mary)
*   (torso john) ⋪ (left-fore-arm john)
*   (line-segment3 ball) ⋪ (line-segment2 ball)
*   (line-segment3 ball) ⋪ (line-segment1 ball)
*   (line-segment2 ball) ⋪ (line-segment1 ball)
```

Figure 8.7: ABIGAIL hypothesizes these layer assertions when processing frame 0 of the movie depicted in figure 8.4. Since not all of these layer assertions are necessary to explain the stability of the image, ABIGAIL retains only the starred layer assertions.

to make part of her joint model.

The joint and layer models constructed by ABIGAIL contain a number of anomalies that point out deficiencies in the perceptual theory. First, note that `(line-segment3 ball)` is not connected to the remaining components of the ball. The intention was that the ball would be composed of four figures, a circle and three line segments. ABIGAIL perceives `(line-segment3 ball)` to be a separate object inside the ball. This is a possible interpretation given her ontology since, being inside the ball, `(line-segment3 ball)` is supported by resting on the interior perimeter of the circle, and thus there is no need to postulate a joint to achieve stability. In fact, given ABIGAIL's preference for support relations over joints, she must come to this analysis. Why then are the remaining two line segments not supported in an equivalent fashion without joints? The answer is simple. For a line segment to be so supported it must be on the same layer as the circle. Since layer equivalence is a transitive relation, all three line segments would have to be on the same layer. They cannot be however, as their intersection would then constitute a substantiality violation. Thus only one line segment can be explained by support. ABIGAIL arbitrarily chooses `(line-segment3 ball)` as that line segment.

The joint and layer models exhibit a second, more serious, anomaly. While ABIGAIL correctly determines that the bottom of the box rests on the table top, she incorrectly decides that the vertical walls of the box are joined to the table top rather than the box bottom. This is a plausible but unintended interpretation. Both interpretations require the same number of joints, thus neither is preferable to the other. One way of driving ABIGAIL to the intended interpretation would be to add an additional level to the preference relation between joint and layer models to prefer one model over another if its joints connected smaller figures rather than larger ones, given that two models otherwise had the same number of joints. I have not tried this heuristic to see if it would work.

At this point ABIGAIL begins processing frame 1. Between frame 0 and frame 1, John lifted his right foot. In doing so he rotated his right knee and thigh joints. Thus the first thing ABIGAIL does is demote the rotation parameters for the joints

`(right-thigh john)` $\leftrightarrow$ `(left-thigh john)`
`(right-thigh john)` $\leftrightarrow$ `(right-calf john)`

from being rigid to being flexible. The resulting image is not stable however. Since John appears to stand on one foot, he falls over when the future is imagined.[11] In the process of falling his right thigh can rotate relative to his torso since that joint is now flexible. ABIGAIL hypothesizes the existence of a new rigid joint, `(torso john)` $\leftrightarrow$ `(right-thigh john)`. While this joint does not prevent John from falling, it does prevent the rotation of his right thigh relative to his torso during that fall. ABIGAIL adopts that joint as part of the updated model since she adopts any joint which prevents the relative rotation of the two figures it would connect.

At this point ABIGAIL begins processing frame 2. Between frame 1 and frame 2, John started moving forward. In doing so he rotated his left knee and thigh joints, causing ABIGAIL to demote the rotation parameters for the joints

`(torso john)` $\leftrightarrow$ `(left-thigh john)`
`(left-thigh john)` $\leftrightarrow$ `(left-calf john)`

from being rigid to being flexible. Between frame 2 and frame 3, John begins moving his right foot forward as well, pivoting his right thigh relative to his torso. This causes ABIGAIL to demote the rotation parameter for the joint `(torso john)` $\leftrightarrow$ `(right-thigh john)`, just created while processing frame 1, from being rigid to being flexible. The model now constructed remains unchanged until frame 7.

[11] I will not show the resulting imagined image since John falls backward out of the field of view due to the fact that his center-of-mass is behind his left foot. Later in the text, I will illustrate the imagined future of frame 11, where John's center-of-mass has shifted so that he falls forward in a visible fashion.

In frame 7, John's knees appear close together as his right leg passes his left leg. This causes ABIGAIL to postulate a spurious joint, `(right-calf john)` $\leftrightarrow$ `(left-calf john)`, between John's two knees. Again, while this joint does not prevent John from falling, it does reduce the movement of his legs during that fall. This reduction in leg movement prompts ABIGAIL to adopt the joint as part of her joint model. This spurious joint is then dropped from the joint model after frame 8, since `(right-calf john)` and `(left-calf john)` no longer intersect. Furthermore, as a result of observing the right leg pass the left leg during its forward motion, ABIGAIL adds the following two assertions to the layer model

`(left-thigh john)` $\not\bowtie$ `(right-calf john)`
`(right-calf john)` $\not\bowtie$ `(left-calf john)`

knowing that otherwise, a substantiality violation would have occurred. At this point, the model remains unchanged through frame 11.

Figure 8.8 depicts the sequence of images produced by ABIGAIL while imagining the short-term future of frame 11. For reasons discussed previously, John's and Mary's eyes fall out in steps 1 and 2. In step 3, John pivots about his left leg until his right foot reaches the floor. In step 4, he pivots about his right foot until his right knee reaches the floor. In step 5, he then pivots about his right knee until both his hand and head reach the floor. This is possible since his right knee has a flexible rotation parameter. Note that his head can appear to pass through the chair since ABIGAIL assumes that objects are on different layers unless she has explicit reason to believe that they are on the same layer. Finally, in step 6, his left calf pivots about his left knee until his left foot reaches the floor. Again, this is possible since his left knee has a flexible rotation parameter.

One can imagine other sources of evidence which can be used to update the joint and layer models. Collisions can be used to determine that two objects are on the same layer, since two objects must be on the same layer in order to collide. A sequence of frames where one object moves toward another object but upon contact (or approximate contact given the finite frame rate) begins moving away from that object, can be interpreted as a collision event, giving evidence that the contacting figures of each object are on the same layer. Such inference could provide information not derivable by the procedure previously described. It is not currently implemented, as determining collisions requires tracking momentum of objects across frames. ABIGAIL currently processes each frame individually.

The continuity constraint offers another source of evidence which can be used to infer that objects are on different layers. Seeing an object totally enclosed by another object in one frame, and then outside that object in the following frame, gives evidence that the two objects are on different layers, even without a directly observed substantiality violation, since there would be no way for that transition to occur, given continuous movement and the substantiality constraint, unless the two objects were on different layers. In contrast to collisions, this would offer little additional inferential power since given a sufficiently high frame rate relative to object velocities, the observer would see an intermediate frame with a direct substantiality violation.

8.2.2 Deriving Support, Contact, and Attachment Relations

ABIGAIL maintains a joint and layer model to reflect her understanding of the movie. These models are continually updated, on a frame-by-frame basis, by the processes described in the previous section. The models form the basis of mechanisms used to derive changing support, contact, and attachment relationships between objects in the movie. It is necessary, however, to first delineate those collections of figures which constitute objects. To this end, ABIGAIL forms the connected components in a graph whose vertices are figures and edges are joints. Each connected component is taken as an object. Not all connected sets of figures constitute objects. Only those which form complete connected components are taken as objects. Once a set of figures is determined to be an object, however, that set retains its status as an independent object, even though it may later be joined to another object. When that

Figure 8.8: The sequence of images produced by ABIGAIL while imagining the short-term future of frame 11 from the movie described in section 6.1.

happens, a part-whole hierarchy is created which represents both the individual parts, as well as the combined whole, as objects. This is needed to model grasping as the formation of a joint between one's hand and the grasped object. The independent identity of both the person grasping an object, as well as the object being grasped, must be maintained, despite the creation of a spurious combined object. Likewise, when a joint is removed from the joint model, an object is broken into parts which are taken as objects. The identity of the original object is retained however. The new parts are thought of both as objects in their own right, as well as parts of an object no longer in existence. ABIGAIL considers an object to exist if the set of figures constituting that object are currently connected. In this way, ABIGAIL can form a primitive model of the words *make* and *break* as the transition of an object from non-existence to existence and vice versa. Furthermore, since ABIGAIL retains the identity of objects no longer in existence, it is possible to model the word *fix* as the transition of an object from existence to non-existence and then back again to existence.

Given the segmentation of an image into objects, the joint and layer models form the basis for detecting contact and attachment relations between those objects. Two objects are attached if the current joint model contains a joint between some figure of one object and some figure of the other object. Two objects are in contact if some figure of one object both touches (in the sense described in figure 8.1), and is on the same layer as, some figure of the other object. Detecting support relations, however, requires further use of the imagination capacity. The lexical semantic representation presented in chapter 7 uses two different support primitives, one to determine whether an object is supported, the other to determine if one object supports another. An object is considered supported if it does not move when the short-term future of the world is imagined. A single call to $I(\mathcal{F}, J, L)$ will suffice to determine those objects which are unsupported.[12] To determine whether an object A supports another object B, ABIGAIL imagines whether B would fall if A were removed. This is done by calling $I(\mathcal{F}-\mathbf{figures}(A), J, L)$ and seeing if B moves. An object A supports another object B only if B is indeed supported. The fact that B falls when A is removed is insufficient to infer that A supports B since B may have fallen even with A still in the image. Here again, a single call to $I(\mathcal{F}-\mathbf{figures}(A), J, L)$ can be used to determine all of the different objects B which are supported by A. Thus for n objects, $n+1$ calls to the imagination capacity I must be performed per frame to determine all support relationships.[13]

The recovery of support, contact, and attachment relations from image sequences is best illustrated by way of several examples. Since the full movie from section 6.1 is fairly complex, I will first illustrate the results produced by ABIGAIL while processing a much shorter and simpler movie. This movie depicts a single object, John, taking two steps forward, turning around, and taking two steps in the other direction. It contains 68 frames, each containing 10 line segments and 2 circles. Figures 8.9 depicts the pivotal frames of this short movie.

ABIGAIL is able to fully process this movie in several minutes of elapsed time on a Symbolics XL1200™ computer, taking several seconds per frame. This is within two orders of magnitude of the processing speed necessary to analyze such a movie in real time. The result of ABIGAIL's analysis is depicted by the event graph illustrated in figure 8.10. Each edge in this graph denotes some collection of perceptual primitives which hold during the interval spanned by that edge. Figures 8.11 and 8.12 enumerate the perceptual primitives associated with each edge in this graph.[14]

[12] Inefficient design of the structure of the current implementation requires $I(\mathcal{F}, J, L)$ to be called independently for each object. Remedying that inefficiency should dramatically improve the performance of the system.

[13] For the same reasons as mentioned before, the current implementation must call I for each pair of objects, thus requiring $n^2 + n$ calls. To mitigate this inefficiency somewhat, the current implementation only discerns direct support, i.e. support relations between objects in contact with each other. Indirect support can be derived by taking the transitive closure of the direct support relation. This efficiency improvement could be combined with the strategy suggested in the text whereby $I(\mathcal{F}-\mathbf{figures}(A), J, L)$ would be called only if A was in contact with some other object.

[14] The perceptual primitives are predicates which hold of objects. As far as ABIGAIL is concerned, objects are simply collections of figures. To make the output more readable, however, objects are printed using notation like `[JOHN]`. This printed notation for objects is derived from the names of the figures comprising the object. Recall that figures are given

Figure 8.9: Several key frames depicting the general sequence of events from a shorter movie used to test ABIGAIL.

Figure 8.10: The event graph depicting the temporal structure of the perceptual primitives recovered by ABIGAIL after processing the short movie from figure 8.9. Each edge denotes some collection of perceptual primitives which hold during the interval spanning that edge. Figures 8.11 and 8.12 enumerate the perceptual primitives associated with each edge in this graph. The edges from 2 to 15, from 18 to 32, from 36 to 49, and from 52 to 66 each correspond to a step taken by John while walking.

In addition to support, contact, and attachment relations, the set of perceptual primitives includes expressions for depicting various kinds of motion, as well as the location of objects and the paths followed by objects during their motion. I will not discuss these primitives in depth as they are tangential to the main focus of this thesis.

At a high level, the correspondence between this event graph and the events in the movie are intuitively obvious. In the movie, John takes four steps while continuously moving. The event graph also depicts four sub-event clusters of the overall motion event. Each cluster further breaks down into a transition between standing on both feet, to moving forward, to again standing on both feet. Note particularly, that John is supported in those situations where he is standing on both feet, namely frames 0, 16, 33, 34, and 50, and not otherwise.[15]

While this event graph bears a global resemblance to the movie, it is not adequate to detect walking

names of the form $(f\ x)$ where x is an 'intuitive' object name given to the figure by the person creating the movie script, and f is an analogous 'intuitive' part name. The printed representation $[c_1, \ldots, c_n]$ delineates the figures which comprise an object by grouping those figures into components c_i based on the intuitive figure name assigned by the script writer. If c_i is a symbol x then it denotes the set of all figures in the image named $(f\ x)$ for some f. If c_i is a pair $(f\ x)$ then it denotes the single figure bearing that name. If c_i is of the form x-`part` then it denotes a set of figures in the image named $(f\ x)$ for any f, where the set contains more than one figure but not all such figures. I should stress that ABIGAIL does *not* use such annotations for anything but printing.

[15] An astute reader may wonder why John doesn't fall even when both feet are on the ground, given that his knee and thigh joints are flexible. The reason for this will be explained in section 9.4.

```
[0,0](PLACE [JOHN-part] PLACE-0)
[0,0](SUPPORTED [JOHN-part])

[0,1](PLACE [(EYE JOHN)] PLACE-1)

[1,67](MOVING [JOHN-part])

[2,2](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[2,2](ROTATING [JOHN-part])

[2,15](MOVING-ROOT [JOHN-part])
[2,15](TRANSLATING [(EYE JOHN)] PLACE-2)
[2,15](MOVING-ROOT [(EYE JOHN)])
[2,15](MOVING [(EYE JOHN)])

[2,67](TRANSLATING [JOHN-part] PLACE-11)

[16,16](SUPPORTED [JOHN-part])

[16,17](PLACE [(EYE JOHN)] PLACE-3)

[18,18](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[18,18](ROTATING [JOHN-part])

[18,32](MOVING-ROOT [JOHN-part])
[18,32](TRANSLATING [(EYE JOHN)] PLACE-4)
[18,32](MOVING-ROOT [(EYE JOHN)])
[18,32](MOVING [(EYE JOHN)])

[33,33](PLACE [(EYE JOHN)] PLACE-5)

[33,34](SUPPORTED [JOHN-part])
```

Figure 8.11: Part I of the perceptual primitives recovered by ABIGAIL after processing the short movie from figure 8.9.

```
[34,34](FLIPPING [JOHN-part])
[34,34](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[34,34](ROTATING-CLOCKWISE [JOHN-part])
[34,34](ROTATING [JOHN-part])
[34,34](MOVING-ROOT [JOHN-part])
[34,34](TRANSLATING [(EYE JOHN)] PLACE-6)
[34,34](ROTATING-COUNTER-CLOCKWISE [(EYE JOHN)])
[34,34](ROTATING [(EYE JOHN)])
[34,34](MOVING-ROOT [(EYE JOHN)])
[34,34](MOVING [(EYE JOHN)])

[35,35](PLACE [(EYE JOHN)] PLACE-7)

[36,36](ROTATING-CLOCKWISE [JOHN-part])
[36,36](ROTATING [JOHN-part])

[36,49](MOVING-ROOT [JOHN-part])
[36,49](TRANSLATING [(EYE JOHN)] PLACE-8)
[36,49](MOVING-ROOT [(EYE JOHN)])
[36,49](MOVING [(EYE JOHN)])

[50,50](SUPPORTED [JOHN-part])

[50,51](PLACE [(EYE JOHN)] PLACE-9)

[52,52](ROTATING-CLOCKWISE [JOHN-part])
[52,52](ROTATING [JOHN-part])

[52,66](MOVING-ROOT [JOHN-part])
[52,66](TRANSLATING [(EYE JOHN)] PLACE-10)
[52,66](MOVING-ROOT [(EYE JOHN)])
[52,66](MOVING [(EYE JOHN)])
```

Figure 8.12: Part II of the perceptual primitives recovered by ABIGAIL after processing the short movie from figure 8.9.

using the definition given in chapter 7.

```
(define step (x)
 (exists (i j k y)
  (and (during i (contacts y ground))
       (during j (not (contacts y ground)))
       (during k (contacts y ground))
       (equal y (foot x))
       (= (end i) (beginning j))
       (= (end j) (beginning k)))))
```

```
(define walk (x)
 (exists (i)
  (and (during i (repeat (step x)))
       (during i (move x))
       (during i
        (exists (y)
         (and (equal y (foot x))
              (contacts y ground))))
       (during i
        (not (exists (y)
              (and (equal y (foot x))
                   (slide-against y ground))))))))
```

Two major things are missing. First, the ground must be reified as an object so that ABIGAIL can detect the changing contact relations between John's feet and the ground. Second, the `slide-against` primitive must be implemented. Future work will address these two issues in the hope that ABIGAIL can detect the occurrence of walking events.

ABIGAIL has processed a sizable portion of the larger movie described in section 6.1. While she cannot yet process the entire movie due to processing time limitations, figure 8.13 depicts an event graph produced for the first 172 frames of that movie. Appendix C enumerates the perceptual primitives associated with the edges in that graph. Producing this event graph required about twelve hours of elapsed time on a Symbolics XL1200™ computer. Comparing this with the time required to process the shorter movie indicates that in practice, the complexity of the event perception procedure depends heavily on the number of figures and objects in the image.[16]

I will not discuss ABIGAIL's analysis of the longer movie in depth except to point out two things. First, one major event that takes place during the first 172 frames is John picking up the ball off the table. The perceptual primitives recovered by ABIGAIL form a solid foundation for recognizing this event. Recall the definition given for *pick up* in chapter 7.

[16]The unreasonable amount of time required to process the longer movie significantly hindered the progress of this research.

Figure 8.13: The event graph depicting the temporal structure of the perceptual primitives recovered by ABIGAIL after processing the first 172 frames of the movie discussed in section 6.1. Appendix C enumerates the perceptual primitives associated with each edge in this graph.

```
(define pick-up (x y)
 (exists (i j z)
  (and (during i (supported y))
       (during i (supports z y))
       (during i (contacts z y))
       (during j (move (hand x)))
       (during j (contacts (hand x) y))
       (during j (attached (hand x) y))
       (during j (supports x y))
       (during j (move y))
       (not (equal z (hand x)))
       (= (end i) (beginning j)))))
```

If we take i to be the interval $[0, 65]$ and j to be the interval $[66, 71]$, the following perceptual primitives taken from appendix C correspond very closely to the above definition.

```
       (during i (supported y))
[0,71](SUPPORTED [BALL-part])
[0,71](SUPPORTED [(LINE-SEGMENT3 BALL)])

       (during i (supports z y))
[0,65](SUPPORTS [TABLE BOX-part] [BALL-part])
[0,71](SUPPORTS [BALL-part] [(LINE-SEGMENT3 BALL)])

       (during i (contacts z y))
[0,65](CONTACTS [TABLE BOX-part] [BALL-part])

       (during j (supports x y))
[66,71](SUPPORTS [JOHN-part] [BALL-part])
[66,71](SUPPORTS [(LINE-SEGMENT3 BALL)] [BALL-part])
[66,71](SUPPORTS [BALL-part JOHN-part] [(LINE-SEGMENT3 BALL)])

       (during j (move y))
[66,71](TRANSLATING [BALL-part] PLACE-19)
[66,71](MOVING-ROOT [BALL-part])
[66,71](MOVING [BALL-part])
[66,71](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-17)
[66,71](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[66,71](MOVING [(LINE-SEGMENT3 BALL)])
```

Note that if an object is supported (by another object) for an interval, say $[0, 71]$, then it is supported for every subinterval of that interval, in particular $[0, 65]$. Given this, ABIGAIL has detected almost all of the prerequisites to recognize a *pick up* event. The only primitives not recognized are the following.

```
       (during j (move (hand x)))
       (during j (contacts (hand x) y))
       (during j (attached (hand x) y))
```

ABIGAIL has in fact detected these prerequisites as well. They just don't appear in the event graph from figure 8.13 as that graph depicts only those primitives which no longer hold after frame 172. John's hand continues to move while grasping the ball well beyond frame 172. The above primitives will become part of the event graph when these actions terminate.

A puzzling thing happens in ABIGAIL's analysis of this movie. ABIGAIL decides that the table is unsupported in frame 172. This is indicated by the fact that event graph contains an edge from frame 0 through frame 171 with the following perceptual primitives.

```
[0,171](SUPPORTED [TABLE BOX-part])
[0,171](SUPPORTED [(BOTTOM BOX)])
[0,171](SUPPORTS [TABLE BOX-part] [(BOTTOM BOX)])
```

Inspection of the movie, however, reveals that the table remains supported throughout the entire movie. What causes ABIGAIL to suddenly decide that the table is unsupported in frame 172? Figure 8.14 depicts the sequence of images that are part of ABIGAIL's imagination of the short-term future for frame 172. In this sequence, John falls over as he is unsupported. In doing so, the ball he is holding knocks against the table. While ABIGAIL knows that John is on a different layer from the table, to allow him to walk across the table without a substantiality violation, she also knows that the ball is on the same layer as the table, since in the past, the table supported the ball. This allows John to raise the table up on one leg by leaning on its edge with the ball. Since ABIGAIL determines that something is unsupported if it moves during imagination, she decides that the table is unsupported. This points out a deficiency in the method used to determine support. An object may be supported even though an unrelated object could knock it over during imagination. Methods to alleviate this problem are beyond the scope of this thesis.

I will discuss one further deficiency in ABIGAIL's mechanism for perceiving support. Recall that ABIGAIL determines that an object A supports an object B if B is supported but loses that support when A is removed. Figure 8.15 depicts a board supported by three tables. Since removing each table individually will not cause the board to fall, ABIGAIL would erroneously conclude that none of the tables support the board. This flaw is easily remedied by having ABIGAIL consider all sets of objects A to see if B falls when the entire set is removed. If so, then either the set can be taken as collectively supporting the object, or support can be attributed to each member of the set individually.

8.3 Experimental Evidence

As discussed previously, a major assumption underlying the design of ABIGAIL is that people continually imagine the short-term future, extrapolating perhaps a second or two into the future, as an ordinary component of visual perception. Freyd and her colleagues have conducted a long series of experiments (Freyd 1983, Freyd and Finke 1984, Finke and Freyd 1985, Freyd and Finke 1985, Finke et al. 1986, Freyd 1987, Freyd and Johnson 1987, Kelly and Freyd 1987) that support this view. These experiments share a common paradigm designed to demonstrate *memory shift*. Subjects are shown a sequence of images which depict one or more objects in motion. They are then shown a test image and asked whether the objects in the test image are in the same position as they were in the final image in the pre-test sequence. Sometimes the objects are indeed in the same position and the correct response is 'same'. Other times however, the objects are displaced along the direction of motion implied by the pre-test image sequence, in either a forward or reverse direction. In this case the correct response is 'different'. Subjects uniformly give more incorrect responses for test images where the objects were displaced further along the path of implied motion than for test images where the objects were displaced in the reverse direction. In fact, for some experiments, subjects were more likely to give a 'same' response for a slight forward displacement than for an image without any displacement. These experiments were repeated, varying a number of parameters. These included the number of pre-test images, the number of moving objects in the image sequence, the length of time each pre-test or test image was displayed, the length of time between the display of each pre-test image or between the display of the final pre-test image and the test image, and whether the images were taken from real photographs or were computer-generated abstractions such as rotating rectangles or moving dots. It appears that subjects' memory of an

Figure 8.14: The sequence of images produced by ABIGAIL while imagining the short-term future of frame 172 from the movie described in section 6.1. ABIGAIL imagines that John will fall and knock over the table. Due to a flaw in the method for determining support, ABIGAIL concludes that the table is unsupported.

Figure 8.15: Three tables collectively supporting a board. ABIGAIL will currently fail to determine that the tables support the board since the board will not fall when each is removed individually.

object's position is shifted reliably as a result of an object's suggested motion. Freyd and her colleagues attribute this memory shift to what they call a mental extrapolation of object movement. Through statistical analysis of the error rates and reaction times for the various experimental tasks, they claim to have demonstrated, among other things, that objects move progressively during extrapolation, that an object's velocity during extrapolation is roughly equivalent to its final velocity implied by the pre-test image sequence, that it takes some time to stop the extrapolation process and the amount of time needed to stop the extrapolation process is proportional to an object's final velocity during the pre-test image sequence. They call this latter phenomenon *representational momentum* due to its similarity to physical momentum.

In many of its details, the extrapolation process uncovered by Freyd and her colleagues differs from the artificial imagination capacity incorporated into ABIGAIL. As I will describe in chapter 9, ABIGAIL's imagination capacity has no notion of velocity or momentum. Nonetheless, I take the results of Freyd and her colleagues as strong encouragement that the approach taken in this thesis is on the right track.

In more recent work, Freyd et al. (1988) report evidence that the human extrapolation process represents forces, such as gravity, in addition to velocities. Furthermore, they report evidence for the representation of forces in equilibrium, even for static images. In particular, their experiments show that subjects who perceive essentially static images with forces in equilibrium, such as one object supporting another, extrapolate motion on the part of the objects in those images when the equilibrium is disturbed, as when the source of support is removed. This is more in line with ABIGAIL's imagination capacity.

The experimental paradigm they used is similar to that used for the memory shift experiments. It is depicted in figure 8.16. Subjects were shown a pre-test sequence of two images followed by a test image. The first image in the pre-test sequence depicted a plant supported either by a stand or by a hook. The plant appeared next to a window to allow subjects to gauge its vertical position. The second image depicted the plant unsupported, with the stand or hook having disappeared. The test image was similar to the second image except that in some instances, the plant was displaced upward or downward from its position in the second image. Subjects viewed each image in the sequence for 250ms, with a 250ms interval between images. They were asked to determine whether the test image depicted the plant in the same position as the second image or whether the test image depicted the plant in a different position. Subjects made more errors determining that the test image differed from the second image when the test image depicted the plant in a lower position than the second image in contrast to when the test image depicted the plant in a higher position. This result can be interpreted as indicating that subjects imagined that the plant fell when its source of support was removed.

ABIGAIL performs an analogous extrapolation when determining support relationships. She continually performs counterfactual analyses determining that an object is supported if it does not fall during extrapolation. A second experiment reported by Freyd et al. (1988) indicates that humans do not perform such analyses in all situations. This experiment is similar to the first experiment except that the plant was also unsupported in the first image, i.e. it was unsupported throughout the image sequence. The image sequence is depicted in figure 8.17. In this experiment, subjects demonstrated no memory shift and thus no tendency to imagine the unsupported plant falling. It appears that a change in support

Figure 8.16: The image sequences shown to subjects as part of an experiment to demonstrate that people represent forces in equilibrium when viewing static images. Reprinted from Freyd et al. (1988).

Figure 8.17: The image sequences shown to subjects as part of an experiment to demonstrate that people don't always represent forces in equilibrium when viewing static images. Reprinted from Freyd et al. (1988).

status is necessary to induce the imagined falling. ABIGAIL's imagination capacity does not accurately reflect this last result.

To summarize, experiments reported by Freyd and her colleagues depict an active perceptual system, forming the basis of our conceptual system, which has as its foundation an imagination capacity which encodes naive physical knowledge. This capacity appears to be in place from very early infancy. This view is most eloquently captured by the following excerpts from Freyd et al. (1988).

> Much of what people encounter in everyday life is static from their point of reference: Cups rest on desks, chairs sit on floors, and books stand on shelves. Perhaps it is the very pervasiveness of static objects and still scenes that has been responsible for psychology's historical focus on the perception of static qualities of the world: shape and form perception, pattern recognition, picture perception, and object recognition. In apparent contrast to this focus, there has been an increasingly popular emphasis on the perception of events, or patterns of change in the world. There is a sense, however, in which the study of event perception (e.g., J. J. Gibson, 1979) has shared some assumptions with the more traditional focus on the perception of static stimuli. In both approaches event and dynamic stimuli have been defined in terms of changes taking place in real time, whereas scenes that are not changing in real time (or are being viewed by an observer who is not moving in real time) have been considered simply static, that is, specifically not dynamic.
>
> This view of static objects and scenes suggests that the perception of a static scene is devoid of information about dynamic qualities of the world (which led J. J. Gibson, 1970, for instance, to consider the perception of static scenes to be a mere laboratory curiosity). But if we take *dynamic* to mean relating to physical force acting on objects with mass, then this view is

incorrect.

[p. 395, emphasis in the original]

Having some sort of access to likely transformations by representing physical forces may help solve a slightly different problem in object recognition: the problem of correctly identifying a particular instantiation, or "token," as a member of a larger class, or type, or object. If part of what one stores in memory about an object type is aspects of its likely behavior when embedded in events, then representing physical forces operating on objects in a particular perceptual situation may help in the process of identification of a particular object token.

[p. 405]

Of course, to go correctly from visual input to a representation of forces, the underlying representation system has to "know" something about physical forces and how they interact with objects for a particular environment, such as the environment encountered on the surface of the planet Earth. Such knowledge may be a function of the inherited or experientially modified representational structure serving perception.

[p. 406]

Indeed, our view suggests that when people are viewing a static scene, lurking behind the surface of consciousness is an inherently dynamic tension resulting from the representation of forces in equilibrium. We see this dynamic tension as contributing to the conscious experience of concreteness in perception and to the memory asymmetries we measure when the equilibrium is disrupted.

[p. 407]

Perhaps we might also be able to determine whether the present findings generalize to physical situations beyond gravity, such as those where pressure (or even electromagnetic force) dominates. However, we suspect that gravity is a better candidate for mental "internalization" than other forces. Shepard (1981, 1984) has argued that the mind has internalized characteristics of the world that have been most pervasive and enduring throughout evolution. Although Shepard's (1981, 1984) list has emphasized kinematic, as opposed to dynamic, transformations, the dynamic aspects of gravity are indeed pervasive and enduring characteristics of the world.

[p. 405]

Although some might accept that the force of gravity and its simple opposing forces (Experiments 1–3) could be represented within the perceptual system, many might argue that the representation of forces active in springs (Experiment 4) implicates real-world learning and thus suggests that the basis of the effect is more central than perceptual. We suggest two responses to this argument: First, perceptual knowledge of springlike behavior may be innately given and not dependent on learning; second, evidence of perceptual learning is not necessarily evidence against modularity. For both of these responses, we question the assumption that the effect in Experiment 4 stems from knowledge of springs per se. It might instead reflect perceptual knowledge of compressible and elastic substances, of which springs are an example. DiSessa (1983) suggested that springiness is a phenomenological primitive. E. J. Gibson, Owsley, Walker, and Megaw-Nyce (1979) found that 3-month-old

> infants extract object rigidity or nonrigidity from motion, suggesting that people distinguish compressible from noncompressible substances at a very early age.
>
> [p. 406][17]

This thesis adopts that above view and takes it as motivation for the design of Abigail's perceptual system.

8.4 Summary

In chapter 7, I argued that the notions of support, contact, and attachment play a central role in the definitions of simple spatial motion verbs such as *throw*, *pick up*, *put*, and *walk*. In this chapter, I presented a theory of how these notions can be grounded in perception via counterfactual simulation. An object is supported if it doesn't fall when the short-term future is imagined. One object supports another object if the second is supported, but loses that support in a world imagined without the first object. Two objects are attached if such attachment is needed to explain the fact that one supports the other. Likewise, two objects must be in contact if one supports the other. A simple formulation of this theory has been implemented as a computer program called ABIGAIL that watches movies constructed out of line segments and circles and produces descriptions of the objects and events depicted in those movies. The events are characterized by the changing status of support, contact, and attachment relations between objects. This chapter has illustrated how such relations could be recovered by using a modular imagination capacity to perform the counterfactual simulations. The next chapter will discuss the inner workings of this imagination capacity in greater detail.

[17] Experiments 1 and 2 correspond to figures 8.16 and 8.17 respectively. Experiment 3 extends experiments 1 and 2 in testing for representation of gravitational forces. Experiment 4 uses a similar experimental setup to test for the representation of forces in a compressed spring as weights are placed on top of the spring and removed from it.

Chapter 9

Naive Physics

Much of ABIGAIL's event perception mechanism, and ultimately the lexical semantic representation she uses to support language acquisition, relies on her capacity for imagining what will happen next in the movie. This imagination capacity is used as part of a continual counterfactual 'what if' analysis to support most of event perception. For example, ABIGAIL infers that two figures are joined if one would fall away from the other were they not joined. Knowing which figures are joined allows her to segment the image into objects comprising sets of figures that are joined together. This ultimately allows the grounding of the lexical semantic primitives `(attached` x y`)` and `(in-existence` x`)`. Furthermore, imagination plays a role in determining support relationships. ABIGAIL infers that two figures are on the same layer if one would fall through the other were they not on the same layer. This is required to ground the lexical semantic primitive `(contacts` x y`)`. Knowing that two figures are on the same layer allows her to determine that one object supports another if the second would fall were the first object removed. This ultimately allows the grounding of the lexical semantic primitives `(supports` x y`)` and `(supported` x`)`.

ABIGAIL's imagination capacity is embodied in a simulator which predicts how a set of figures will behave under the influence of gravity. Gravity will cause the figures to move subject to several constraints.

joint constraints: Figures that are joined must remain joined. The values of rigid joint parameters must be preserved.

substantiality: Two figures which are on the same layer must not overlap.

ground plane: No figure can overlap the line $y = 0$.

Furthermore, each of these constraints is subject to the notion of *continuity*. Not only must all figures uphold the joint, substantiality, and ground plane constraints in their final resting position, they must uphold these constraints continuously at all points along their path of motion. Figure 8.8 on page 141, gives an example of ABIGAIL's imagination capacity in operation.

The problem of simulating the behavior of a set of components under the influence of forces subject to constraints is not new. Much work on this problem has been done in the field of mechanical engineering and robotics where this problem is called *kinematic simulation* of mechanisms. The classical approach to kinematic simulation uses numerical integration.[1] Essentially, it is treated as an n-body problem subject to constraints. Since the constraints are typically complex, it is difficult to derive an analytic, closed-form method of preserving constraints during integration. Accordingly, the common approach

[1] Two notable exceptions to this are the work of Kramer (1990a, 1990b) and Funt (1980). I will discuss this work in section 10.1

is to integrate using a small step size and repeatedly check for constraint violations. Preventing constraint violations is often accomplished by modeling them as additional forces acting on the components. Cremer's thesis (1989) is an example of recent work in kinematic simulation using numerical integration.

The classical approach to kinematic simulation has certain merits. Up to the limits of numerical accuracy, it faithfully models the Newtonian physics of a mechanism. This includes the velocity, momentum, and kinetic energy of its components as well as the magnitude of forces collectively acting on each component. It can handle arbitrary forces as well as arbitrary motion constraints. Except where numerical methods break down at singularities, it accurately predicts the precise motion that components undergo, the paths they follow, and their final resting place when the mechanism reaches equilibrium.

While this classical approach to kinematic simulation is useful in mechanical engineering, it is less suitable as a cognitive model of an innate imagination capacity, if one exists. The approach is both too powerful and at the same time too weak. On one hand, people are not able to accurately predict the precise paths taken by components of complex mechanisms. On the other hand, people do not appear to be performing numerical integration with a small step size. Consider the mechanism shown in figure 9.1. The mechanism consists of a ball attached to a rod which is joined to a stand. The joint is flexible, allowing the rod to pivot and the ball to fall until it hits the table. The classical approach will simulate such a mechanism by small repeated perturbations of the joint angle θ. After each perturbation, a constraint check is performed to verify that the ball does not overlap the table. There is something unsatisfying about this approach. People seem to be able to predict that the rod will pivot precisely the amount needed to bring the ball into contact with the table.

Using a small but nonzero step size has other consequences that conflict with the needs entailed by using a kinematic simulator as part of a model of event perception. One one hand, smaller step sizes slow the numerical integration process. Current kinematic simulators typically operate two to three orders of magnitude *slower* than real time. Event perception however, must perform numerous simulations per frame to support counterfactual analysis. As discussed in chapter 8, to determine support relationships alone, a simulation must be performed for each pair of objects in the image to determine whether one object falls when the other object is removed. To be cognitively plausible, or at least computationally useful for event perception, the simulator incorporated into the imagination capacity must operate two to three orders of magnitude *faster* than real time, not slower. Admittedly, the current implementation is nowhere near that fast. Nonetheless, it does perform hundreds if not thousands of simulations during the five to ten minutes it takes to process each movie frame.

Using a large step size to speed up the classical approach is likewise cognitively implausible. Large step sizes raise the possibility of continuity violations. The configurations before and after an integration step may both satisfy all of the constraints yet there may be no continuous path for the components to take to achieve that perturbation which does not violate some constraint. For example, if the ball in figure 9.1 was smaller and the step size was larger than the diameter of the ball, a classical simulator could err and predict that the ball would fall through the table. While in normal mechanical engineering practice, judicious choice of step size prevents such errors from occurring, there is something unsatisfying about using the classical approach as a cognitive model. Irrespective of their size, people seem able to uniformly predict that objects move along continuous paths until obstructed by obstacles.

The kinematic simulator incorporated into ABIGAIL uses very different methods than classical simulation with the objective of being both more faithful as a cognitive model and fast enough to support event perception. It is motivated by the desire to simulate mechanisms like the one shown in 9.1 in a single step (which it in fact does). To do so, it takes the cognitive notions of substantiality, continuity, gravity, and ground plane to be primary, and Newtonian physical accuracy to be secondary. To simplify the task of enforcing the cognitive constraints, ABIGAIL's imagination capacity ignores many aspects of physical reality and restricts the class of mechanisms it can simulate. First, the simulator ignores the velocity of objects. This implies ignoring the effects of momentum and kinetic energy on object motion.

Figure 9.1: The simulator incorporated into ABIGAIL's imagination capacity can predict in a single step that the joint will pivot exactly the amount needed until the ball lands on the table. Classical kinematic simulators based on numerical integration repeatedly vary the angle θ by a small step size until the ball collides with the table. If the step size is too small the simulation is slow. If the step size is too large the collision might not be detected, resulting in a simulation which violates the substantiality and continuity constraints. ABIGAIL never produces such an anomalous prediction.

Rather than integrating accelerations into velocities and positions, the simulator operates as an optimizer, simply moving objects along paths which reduce their potential energy. Second, for the most part, the simulator ignores the magnitude of forces acting on objects when computing their potential energy. Objects simply move when forces are applied to them, in a direction which decreases their potential energy. They don't move any faster when the force is greater nor do objects necessarily move in a direction which offers the greatest decrease in potential energy. Third, the simulator considers moving only rigid objects, or rigid parts of objects, along linear or circular paths, one at a time, when attempting to reduce their potential energy. Any mechanism which involves either motion along a more complex path or simultaneous motion of multiple objects along different paths cannot be correctly simulated. This precludes simulating mechanisms with closed-loop kinematic chains.[2] While these limitations make this simulator inappropriate for traditional mechanical engineering tasks, at least the first two limitations are inconsequential for the task of modeling the use of imagination to support event perception. The third limitation does, however, cause some problems. These will be discussed in section 9.4.

9.1 Simulation Framework

ABIGAIL simulates the imagined future of an image by moving sets of figures from that image along linear and circular paths which reduce the potential energy of the set of moved figures. The potential energy of a set of figures is simply the sum of the potential energies of each figure in that set. The potential energy of a figure f is taken to be the product of its mass $m(f)$ and the height of its center-of-mass $y(f)$.

ABIGAIL's kinematic simulator is a function $I(\mathcal{F}, J, L, P)$ which takes as input, a set of figures $\mathcal{F}$, along with a joint model J, a layer model L, and a predicate P.[3] Each figure $f \in \mathcal{F}$ has an *observed* position, orientation, shape, and size as derived from the current movie frame. From this input, $I(\mathcal{F}, J, L, P)$ calculates a series of *imagined* positions and orientations for each $f \in \mathcal{F}$.[4] This series of positions and orientations constitutes the motion predicted by ABIGAIL for the figures under the influence of gravity. I will denote the imagined positions and orientations of a figure f as $\hat{x}(p(f))$, $\hat{y}(p(f))$, and $\hat{\theta}(f)$ in contrast to the observed positions $x(p(f))$, $y(p(f))$, and $\theta(f)$. I similarly extend such notation to distances $\hat{\Delta}(p, q)$, displacements $\hat{\delta}(p, f)$, and any other notion ultimately based on coordinates of figure points. During imagination, ABIGAIL applies the predicate P to the imagined positions and orientation of the figures after moving each group of figures. If $P(\hat{\mathcal{F}})$ ever returns **true** then the simulation is halted and $I(\mathcal{F}, J, L, P)$ returns **true**. If $P(\hat{\mathcal{F}})$ never returns **true** and the simulation reaches a state where no further movement is possible, $I(\mathcal{F}, J, L, P)$ returns **false**. Thus $I(\mathcal{F}, J, L, P)$ can be interpreted as asking whether P will happen imminently in the current situation.

During simulation, ABIGAIL will move one set of figures, while leaving the remaining figures stationary. The set of moved figures will be called the *foreground* while the stationary figures will be called the *background*. I will denote the set of foreground figures as F and the set of background figures as G. The sets F and G are disjoint. Their union constitutes the entire set of figures $\mathcal{F}$ being imagined. This might not be equivalent to the entire set of figures in the current movie frame, since ABIGAIL often imagines what would happen if certain figures were missing, as is the case when she tries to determine whether one object supports another by imagining a world without the first object. Two kinds of foreground

[2] A closed-loop kinematic chain is a set of components $\{c_1, \ldots, c_n\}$ where each c_i except c_n is joined to c_{i+1} and c_n is joined back to c_1.

[3] This may appear to be circular since $I(\mathcal{F}, J, L, P)$ takes joint and layer models as input, and in turn, is used to compute joint and layer models according to the process described in section 8.2.1. This circularity is broken by calling $I(\mathcal{F}, J, L, P)$ with empty joint and layer models initially to compute the first joint and layer models, and using the previous models at each frame to compute the updated models. Surprisingly, it usually takes ABIGAIL only a single frame to converge to the correct models.

[4] Independent of the simplifying assumption discussed in section 8.1.2, during imagination the shapes and sizes of figures must remain invariant to avoid producing degenerate predictions.

motion are considered: translating F along a linear axis whose orientation is θ, and rotating F about a pivot point p. The pivot point need not lie on any figure in F. In fact it can be either inside or outside the bounding area of F.

The simulator operates by repeatedly choosing some foreground F, and either translating F along an appropriate axis θ, or rotating F about an appropriate pivot point p, as far as it can, so long as the potential energy of F is continually decreased and the substantiality, ground plane, and joint constraints are not violated. It terminates when it cannot find some foreground it can move to decrease its potential energy. At each step of the simulation there may be several potential motions which could each reduce the potential energy. For the most part, the choice of which one to take is somewhat arbitrary, though there is a partial ordering bias which will be described shortly.

The key facet of this simulation algorithm is that at each step, the foreground is translated or rotated *as far as possible* subject to the requirements that potential energy continually decrease and constraints be maintained. Limiting all motion to be linear or circular, and limiting figure shapes to be line segments and circles, allows closed-form analytic determination of the maximum movement possible during that step. Later in this section, I will discuss this fairly complex closed-form solution.

At each simulation step, ABIGAIL must choose an appropriate foreground F, decide whether to translate or rotate F, and choose an appropriate axis θ for the translation or pivot point p for the rotation. Having made these choices, the maximum movement ϵ is analytically determined. Choosing the type of movement (F, and θ or p), however, involves search. ABIGAIL considers the following six possibilities in order.

Translating an object downwards. In this case F consists of a set of figures connected by joints and $\theta = -\frac{\pi}{2}$. There must by no joint between any foreground and background figures. Thus F must be a connected component in the *connection graph* whose vertices are figures and edges are joints between pairs of figures.

Sliding an object along an inclined surface. In this case F consists of a connected component in the connection graph and θ is either the orientation $\theta(f)$, or the opposite orientation $\theta(f) + \pi$, whichever is negative when normalized, of some line segment f such that either

1. f is in the foreground and is coincident with a line segment g in the background,
2. f is in the background and touches a line segment g in the foreground at an endpoint of g,
3. f is in the background and is tangent to a circle g in the foreground,
4. f is in the foreground and touches a line segment g in the background at an endpoint of g, or
5. f is in the foreground and is tangent to a circle g in the background

as long as $f \bowtie g$. No other translations axes need be considered for this case. Furthermore, neither vertical nor horizontal translation axes need be considered since vertical translation axes fall under the previous case, and horizontal motion will never reduce the potential energy of an object. Figures 9.2(a) through 9.2(e) depict cases 1 through 5 respectively. These cases may at times yield multiple potential sliding axes for a given foreground as demonstrated in figures 9.2(f) through 9.2(h). In figure 9.2(f) these degenerate to the same axis. In both figures 9.2(g) and 9.2(h) only one of the two axes allows unblocked movement. In general, when multiple sliding axes are predicted, they will either be degenerate, or all but one will be blocked.

An object falling over. In this case F consists of a connected component in the connection graph and p is either

1. an endpoint of a line segment from F if the endpoint lies on the ground,
2. an endpoint of a line segment f from F if the endpoint lies on a figure g from G and $f \bowtie g$,

Figure 9.2: Determining the potential axes θ of sliding. A foreground might slide relative to a background along any line segment from the foreground which either is coincident with some line segment, touches the endpoint of some line segment, or is tangent to some circle, in the background, or along any line segment from the background with an analogous relationship to a figure in the foreground. Other axes, including the orientations of unrelated line segments, line segments which touch other figures in ways other than those specified above, or line segments which don't touch across the foreground and background boundary need not be considered.

3. an endpoint of a line segment g from G if the endpoint lies on a figure f from F and $f \bowtie g$,
4. the center of a circle from F if the circle touches the ground,
5. the center of a circle f from F if the circle touches a figure g from G and $f \bowtie g$, or
6. the center of a circle g from G if the circle touches a figure f from F and $f \bowtie g$.

No other pivot points need be considered for this case. Figures 9.3(a) through 9.3(f) depict cases 1 through 6 respectively.

Varying a flexible rotation parameter of a joint. If j is a joint with a flexible rotation parameter that connects two parts of an object that are otherwise unconnected then it is possible to rotate either part about the joint pivot. In this case F can be any connected component in the connection graph computed without j such that F contains either $f(j)$ or $g(j)$. The only pivot point which need be considered is $p(j)$, the point where the two figures are joined. If j is not part of a closed-loop kinematic chain then there will always be exactly two such foregrounds F, one for each subpart connected by j. One subpart will contain $f(j)$ while the other will contain $g(j)$. If j is part of a closed-loop kinematic chain then there will be a single such foreground F containing both $f(j)$ and $g(j)$. ABIGAIL detects this case and simply does not consider rotating about flexible joints in closed-loop kinematic chains. This amounts to treating all closed-loop kinematic chains as rigid bodies.

Varying a flexible translational displacement parameter of a joint. If j is a joint such that $\delta_f(j)$ is flexible and $f(j)$ is a line segment then it is possible to translate either part connected by j along $f(j)$. In this case only the orientation $\theta(f(j))$, or the opposite orientation $\theta(f(j)) + \pi$, need be considered as possible translation axes, whichever is negative when normalized. Likewise, if $\delta_g(j)$ is flexible and $g(j)$ is a line segment then it is possible to translate either part connected by j along $g(j)$. In this case only the orientation $\theta(g(j))$, or the opposite orientation $\theta(g(j)) + \pi$, need be considered as possible translation axes, whichever is negative when normalized. In both cases, the translation is limited to the distance between $p(j)$ and the appropriate endpoint of the line segment along which the translation is taken. The limits imposed by this constraint are computed analytically and combined with the limits implied by the substantiality and ground plane constraints. The foreground F is computed in the same way as for the aforementioned case of varying a flexible rotation parameter and is limited to varying joints which do not participate in closed-loop kinematic chains.

Varying a flexible rotational displacement parameter of a joint. If j is a joint such that $\delta_f(j)$ is flexible and $f(j)$ is a circle then it is possible to rotate either part connected by j about the center of $f(j)$. In this case the only pivot point that need be considered is $p(f(j))$. Likewise, if $\delta_g(j)$ is flexible and $g(j)$ is a circle then it is possible to rotate either part connected by j about the center of $g(j)$. In this case the only pivot point that need be considered is $p(g(j))$. The foreground F is computed in the same way as for the case of varying a rotation parameter and is limited to varying joints which do not participate in closed-loop kinematic chains.

Currently, only the first four cases are implemented. Varying displacement parameters of joints is not implemented though it is not conceptually difficult to do so.

Having chosen a foreground F, and whether to translate F along a chosen axis θ, or to rotate F about a chosen pivot point p, the simulator must now determine ϵ, the amount of the translation or rotation. As mentioned previously, the simulator will always translate or rotate the foreground as far as it will go, in a single analytic step, until one of two conditions occur: either further translation or rotation will no longer decrease the potential energy or a barrier prevents further movement. There are

Figure 9.3: Determining the potential pivot points p about which an object may rotate when falling over. When falling over, an object can pivot only about a point touching the ground or another object. No other pivot points need be considered.

two kinds of barriers: the ground, via the ground plane constraint, and another figure on the same layer, via the substantiality constraint.

Determining when further translation or rotation will no longer decrease the potential energy is easy. For translation along an axis θ, there is no limit. So long as the axis of translation θ is negative when normalized, further downward translation of F will always decrease the potential energy of F. Upward translations where θ is positive need never be considered since they can only increase the potential energy. Likewise, horizontal translations, where $\theta = 0$ or $\theta = \pi$ need not be considered since they will not affect the potential energy.[5] For rotation about a pivot point p, the appropriate limit is the rotation which would bring the center-of-mass of F directly below p. This rotation can be calculated as follows. First compute the center-of-mass of F which I will denote as $p(F)$.

$$\begin{aligned} x(p(F)) &= \frac{\sum_{f \in F} m(f)x(f)}{\sum_{f \in F} m(f)} \\ y(p(F)) &= \frac{\sum_{f \in F} m(f)y(f)}{\sum_{f \in F} m(f)} \end{aligned}$$

Then compute the orientation of the line from the pivot point p to this center-of-mass $p(F)$. This is $\theta(p, p(F))$. The desired rotation limit is $-\frac{\pi}{2} - \theta(p, p(F))$. If this value is zero when normalized then no rotation of F about the pivot point p will reduce the potential energy of F, so such a rotation is not considered. If the value is negative when normalized then only a clockwise rotation can reduce the potential energy of F. If the value is positive but not equal to π when normalized then only a counterclockwise rotation can reduce the potential energy of F. If the value is π when normalized then the choice of rotation direction is indeterminate since either a clockwise or counterclockwise rotation will reduce the potential energy. In this case a counterclockwise rotation is chosen arbitrarily. Furthermore, if the pivot point p is coincident with the center-of-mass $p(F)$ then no rotation of F about the pivot point p will reduce the potential energy of F, so again, such a rotation is not considered. Since the magnitude of a rotation need never be greater than π we can represent clockwise rotations as negative normalized rotations and counterclockwise rotations as positive normalized rotations.

9.2 Translation and Rotation Limits

Determining the translation and rotation limits that result from barriers is more complex. In essence, the following procedures are needed.

- `(aggregate-translation-limit` F G θ`)`
- `(aggregate-clockwise-rotation-limit` F G p`)`
- `(aggregate-counterclockwise-rotation-limit` F G p`)`

These determine the maximum translation or rotation ϵ that can be applied to a foreground F until it collides with either the ground or with the background G. Translating or rotating a foreground F will translate or rotate each figure $f \in F$ along the same axis θ or about the same pivot point p. A foreground F can be translated or rotated until any one of its figures $f \in F$ is either blocked by the ground or by some figure $g \in G$ such that f and g are on the same layer.[6] Being blocked by the

[5] The reason angles are normalized so that a leftward orientation is $+\pi$ and not $-\pi$ is so that only downward translation axes are negative.

[6] Recall that ABIGAIL assumes that two figures are on different layers unless she has explicit reason to believe that they are on the same layer.

ground, i.e. the ground plane constraint, can be handled as a variation of the substantiality constraint by temporarily treating the ground as a sufficiently long line segment that is on the same layer as every figure in the foreground. Thus the above procedures which compute movement limits for a whole foreground can be implemented in terms of procedures which compute limits for individual figures via the following template.[7]

```
(defun aggregate-type-limit (F G θ)
 (iterate outer
          (for f in F)
          (minimize (type-limit f *ground* θ))
          (iterate (for g in G)
                   (when (same-layer? f g)
                    (in outer (minimize (type-limit f g θ)))))))
```

where *type* is either `translation`, `clockwise-rotation` or `counterclockwise-rotation`. To implement the functions

- `translation-limit`,
- `clockwise-rotation-limit`, and
- `counterclockwise-rotation-limit`

which compute movement limits for individual figures, eight major cases must be considered.

1. Translating a line segment f until blocked by a another line segment g.
2. Translating a circle f until blocked by a line segment g.
3. Translating a line segment f until blocked by a circle g.
4. Translating a circle f until blocked by another circle g.
5. Rotating a line segment f until blocked by a another line segment g.
6. Rotating a circle f until blocked by a line segment g.
7. Rotating a line segment f until blocked by a circle g.
8. Rotating a circle f until blocked by another circle g.

Each of these eight cases contains a number of subcases. Many of these cases and subcases compute the amount that f may move until blocked by g by instead computing the amount that g may move in the opposite direction until blocked by f. Translations in the opposite direction involve a translation axis whose orientation is $\theta + \pi$ instead of θ. Rotations in the opposite direction return clockwise limits as counterclockwise ones and vice versa. I will now consider each of these eight major cases individually, along with their subcases.

Translating a line segment f until blocked by another line segment g.

This case contains four subcases, all of which must be considered. The tightest limit returned by any of the subcases is the limit returned by this case.

[7] This code fragment uses the `iterate` macro introduced by Amsterdam (1990).

Figure 9.4: Translating a line segment f until its endpoint $p(f)$ touches another line segment g.

Translating f until its endpoint $p(f)$ touches g.

This subcase is depicted in figure 9.4. Project a ray r from $p(f)$ along the axis θ. This ray will be called a *translation ray*. If r does not intersect g then this subcase does not limit the translation of F along the axis θ. However, if r does intersects g at p_1 then the distance from $p(f)$ to p_1 is a limit on the translation of F along the axis θ. The position of f after the translation is depicted as f_1 in figure 9.4.

This case has a boundary case to consider when the translation ray r intersects g at one of its endpoints. If r intersects $p(g)$ then g limits the translation of f only when $|\theta(f) - \theta(g)| < \frac{\pi}{2}$ when normalized. Likewise, if r intersects $q(g)$ then g limits the translation of f only when $|\theta(f) - \theta(q(g), p(g))| < \frac{\pi}{2}$ when normalized. These boundary cases are illustrated in figure 9.5. In figure 9.5, the endpoint $p(g)$ of line segment g limits the translation of f but not the translation of f'.

Translating f until its endpoint $q(f)$ touches g.

This case is analogous to the first subcase except that the translation ray is projected from $q(f)$ instead of $p(f)$.

Translating f until it touches the endpoint $p(g)$.

This case reduces to the first subcase by translating g in the opposite direction $\theta+\pi$ until $p(g)$ touches f.

Translating f until it touches the endpoint $q(g)$.

This case reduces to the second subcase by translating g in the opposite direction $\theta + \pi$ until $q(g)$ touches f.

Translating a circle f until blocked by a line segment g.

This case contains three subcases, all of which must be considered. The tightest limit returned by any of the subcases is the limit returned by this case.

Translating f until it is tangent to g.

This subcase is depicted in figure 9.6. Construct two lines segments, g_1 and g_2, parallel to and on either side of the line segment g, separated from g by a distance equal to the radius

Figure 9.5: A boundary case of the case depicted in figure 9.4 occurs when the translation ray r intersects an endpoint of g. In this case, the endpoint $p(g)$ of g limits the translation of f but not the translation of f'.

Figure 9.6: Translating a circle f until it is tangent to a line segment g.

of the circle f. The endpoints of g_1 and g_2 are those that result from moving the endpoints of g a distance equal to the radius of f along axes which are perpendicular to g. The line segments g_1 and g_2 are the potential loci of the center of the circle f if it were tangent to g. Project a translation ray r from the center $p(f)$ of the circle along the axis θ. If r does not intersect either g_1 or g_2 then this subcase does not limit the translation of F along the axis θ. However, if r does intersect g_1 at p_1 then the distance from $p(f)$ to p_1 is a limit on the translation of F along the axis θ. Likewise, if r intersects g_2 at p_2 then the distance from $p(f)$ to p_2 is a limit on the translation of F along the axis θ. The position of f after the translation is depicted as f_1 in figure 9.6.

Translating f until it touches the endpoint $p(g)$.
This subcase reduces to the second subcase of the next case by translating the line segment g in the opposite direction $\theta + \pi$ until its endpoint $p(g)$ touches the circle f.

Translating f until it touches the endpoint $q(g)$.
This subcase reduces to the third subcase of the next case by translating the line segment g in the opposite direction $\theta + \pi$ until its endpoint $q(g)$ touches the circle f.

Translating a line segment f until blocked by a circle g.
This case contains three subcases, all of which must be considered. The tightest limit returned by any of the subcases is the limit returned by this case.

Translating f until it is tangent to g.
This subcase reduces to the first subcase of the previous case by translating the circle g in the opposite direction $\theta + \pi$ until it is tangent to the line segment f.

Figure 9.7: Translating a line segment f until its endpoint $p(f)$ touches a circle g.

Translating f until its endpoint $p(f)$ touches g.
This subcase is depicted in figures 9.7 and 9.8. Project a translation ray r from the endpoint $p(f)$ along the axis θ. If r does not intersect the circle g then this subcase does not limit the translation of F along the axis θ. However, if r does intersect g at one point p_1, as it does in figure 9.7, then the distance from $p(f)$ to p_1 is a limit on the translation of F along the axis θ. If r intersects g at two points p_1 and p_2, as it does in figure 9.8, then the shorter of the distances from $p(f)$ to p_1 and from $p(f)$ to p_2 is a limit on the translation of F along the axis θ. The position of f after the translation is depicted as f_1 in figures 9.7 and 9.8.

Translating f until its endpoint $q(f)$ touches g.
This subcase is analogous to the second subcase except that the translation ray is projected from $q(f)$ instead of $p(f)$.

Translating a circle f until blocked by another circle g.
This case contains three disjoint subcases. The applicable subcase can be determined analytically by examining the centers and radii of the circles f and g.[8]

[8] In the anomalous situation where f and g are equiradial and concentric either the second or the third case can be used.

Figure 9.8: Translating a line segment f until its endpoint $p(f)$ touches a circle g.

Figure 9.9: Translating a circle f until blocked by another circle g when f and g are outside each other.

The circles are outside each other.

This subcase is depicted in figure 9.9. In this subcase the circle f is translated until it is tangent to and outside the circle g. Construct a circle g_1, concentric with g, whose radius is the sum of the radii of f and g. Project a translation ray r from the center $p(f)$ of f along the axis θ. If r does not intersect g_1 then this subcase does not limit the translation of F along the axis θ. However, if r does intersect g_1 then it will do so at two points, p_1 and p_2, which may degenerate to the same point. The shorter of the distances from $p(f)$ to p_1 and from $p(f)$ to p_2 is a limit on the translation of F along the axis θ. The position of f after the translation is depicted as f_1 in figure 9.9.

The circle f is inside g.

This subcase is depicted in figure 9.10. In this subcase the circle f is translated until it is tangent to and inside the circle g. Construct a circle g_1, concentric with g, whose radius is the radius of g minus the radius of f. Project a translation ray r from the center $p(f)$ of f along the axis θ. Note that r must intersect g_1 at a single point p_1. The distance from $p(f)$ to p_1 is a limit on the translation of F along the axis θ. The position of f after the translation is depicted as f_1 in figure 9.10.

The circle g is inside f.

This subcase reduces to the second subcase by translating g in the opposite direction $\theta + \pi$ until blocked by f.

Rotating a line segment f until blocked by another line segment g.

This case contains four subcases, all of which must be considered. The tightest limit returned by

Figure 9.10: Translating a circle f until blocked by another circle g when f is inside g.

Figure 9.11: Rotating a line segment f until its endpoint $p(f)$ touches another line segment g.

any of the subcases is the limit returned by this case.

Rotating f until its endpoint $p(f)$ touches g.

This subcase is depicted in figure 9.11. Construct a circle c whose center is the pivot point p and whose radius is the distance from p to the endpoint $p(f)$ of line segment f. This circle will be called a *pivot circle*. If c does not intersect line segment g then this subcase does not limit the rotation of F about the pivot point p. However, if c does intersect g at a single point p_1 then $\theta(p, p(f)) - \theta(p, p_1)$ is a limit on the clockwise rotation of F about the pivot point p while $\theta(p, p_1) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. If c intersects g at two points p_1 and p_2 then the larger of $\theta(p, p(f)) - \theta(p, p_1)$ and $\theta(p, p(f)) - \theta(p, p_2)$ is a limit on clockwise rotation while the larger of $\theta(p, p_1) - \theta(p, p(f))$ and $\theta(p, p_2) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. The position of f after the maximal clockwise rotation is depicted as f_1 in figure 9.11. Ignoring limits introduced by other subcases, the position of f after the maximal counterclockwise rotation is depicted as f_2 in figure 9.11.

This case has a boundary case to consider when the pivot circle c intersects g at one of its endpoints. If either p_1 or p_2 in the above discussion is an endpoint of g then that point is considered as an intersection of c with g, for the purposes of limiting the rotation of f only if $|\theta(f) - \theta(p(f), p)| < \frac{\pi}{2}$ when normalized. This boundary case is illustrated in figure 9.12. In figure 9.12, the endpoint $p(g)$ of line segment g limits the rotation of f but not the rotation of f'.

Figure 9.12: A boundary case of the case depicted in figure 9.11 occurs when the pivot circle c intersects an endpoint of g. In this case, the endpoint $p(g)$ of g limits the rotation of f but not the rotation of f'.

Rotating f until its endpoint $q(f)$ touches g.
This subcase is analogous to the first subcase except that the pivot circle is constructed with a radius equal to the distance from p to $q(f)$ instead of the distance from p to $p(f)$.

Rotating f until it touches the endpoint $p(g)$.
This subcase reduces to the first subcase by rotating g in the opposite direction until $p(g)$ touches f. Clockwise limits become counterclockwise limits and vice versa.

Rotating f until it touches the endpoint $q(g)$.
This subcase reduces to the second subcase by rotating g in the opposite direction until $q(g)$ touches f. Clockwise limits become counterclockwise limits and vice versa.

Rotating a circle f until blocked by a line segment g.
This case contains three subcases, all of which must be considered. The tightest limit returned by any of the subcases is the limit returned by this case.

Rotating f until it is tangent to g.
This subcase is depicted in figure 9.13. Construct two lines segments, g_1 and g_2, parallel to and on either side of the line segment g, separated from g by a distance equal to the radius of the circle f. The endpoints of g_1 and g_2 are those that result from moving the endpoints of g a distance equal to the radius of f along axes which are perpendicular to g. The line segments g_1 and g_2 are the potential loci of the center of f if it were tangent to g. Construct a pivot circle c whose center is the pivot point p and whose radius is the distance from p to the center $p(f)$ of the circle. If c does not intersect either g_1 or g_2 then this subcase does not limit the rotation of F about the pivot point p. However, if c does intersect g_1 at a single point p_1 then $\theta(p, p(f)) - \theta(p, p_1)$ is a limit on the clockwise rotation of F about the pivot point p while $\theta(p, p_1) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. If c intersects g_1 at two points p_1 and p_2 then the larger of $\theta(p, p(f)) - \theta(p, p_1)$ and $\theta(p, p(f)) - \theta(p, p_2)$ is a limit on clockwise rotation while the larger of $\theta(p, p_1) - \theta(p, p(f))$ and $\theta(p, p_2) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. Likewise, if c intersects g_2 at a single point q_1 then $\theta(p, p(f)) - \theta(p, q_1)$ is a limit on clockwise rotation while $\theta(p, q_1) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. If c intersects g_2 at two points q_1 and q_2 then the larger of $\theta(p, p(f)) - \theta(p, q_1)$ and $\theta(p, p(f)) - \theta(p, q_2)$ is a limit on clockwise rotation while the larger of $\theta(p, q_1) - \theta(p, p(f))$ and $\theta(p, q_2) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. The position of f after the maximal clockwise rotation is depicted as f_1 in figure 9.13 while the position of f after the maximal counterclockwise rotation is depicted as f_2.

Rotating f until it touches the endpoint $p(g)$.
This subcase reduces to the second subcase of the next case by rotating the line segment g in the opposite direction until its endpoint $p(g)$ touches the circle f. Clockwise limits become counterclockwise limits and vice versa.

Rotating f until it touches the endpoint $q(g)$.
This subcase reduces to the third subcase of the next case by rotating the line segment g in the opposite direction until its endpoint $p(g)$ touches the circle f. Clockwise limits become counterclockwise limits and vice versa.

Rotating a line segment f until blocked by a circle g.
This case contains three subcases, all of which must be considered. The tightest limit returned by any of the subcases is the limit returned by this case.

Rotating f until it is tangent to g.
This subcase reduces to the first subcase of the previous case by rotating the circle f in

Figure 9.13: Rotating a circle f until it is tangent to a line segment g.

Figure 9.14: Rotating a line segment f until its endpoint $p(f)$ touches a circle g.

the opposite direction until it is tangent to the line segment g. Clockwise limits become counterclockwise limits and vice versa.

Rotating f until its endpoint $p(f)$ touches g.

This subcase is depicted in figure 9.14. Construct a pivot circle c whose center is the pivot point p and whose radius is the distance from p to the endpoint $p(f)$ of the line segment. If c does not intersect the circle g then this subcase does not limit the rotation of F about the pivot point p. However, if c does intersect g then it will do so at the two points, p_1 and p_2, which may degenerate to the same point. The larger of $\theta(p, p(f)) - \theta(p, p_1)$ and $\theta(p, p(f)) - \theta(p, p_2)$ is a limit on the clockwise rotation of F about the pivot point p while the larger of $\theta(p, p_1) - \theta(p, p(f))$ and $\theta(p, p_2) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. Ignoring limits introduced by other subcases, the position of f after the maximal clockwise rotation is depicted as f_1 in figure 9.14 while the position of f after the maximal counterclockwise rotation is depicted as f_2.

Figure 9.15: Rotating a circle f until blocked by another circle g when f and g are outside each other.

Rotating f until its endpoint $q(f)$ touches g.
This subcase is analogous to the second subcase except that the pivot circle is constructed with a radius equal to the distance from p to $q(f)$ instead of the distance from p to $p(f)$.

Rotating a circle f until blocked by another circle g.
This case contains three disjoint subcases. The applicable subcase can be determined analytically by examining the centers and radii of the circles f and g.

The circles are outside each other.
This subcase is depicted in figure 9.15. In this subcase the circle f is rotated until it is tangent to and outside the circle g. Construct a circle g_1, concentric with g, whose radius is the sum of the radii of f and g. Construct a pivot circle c whose center is the pivot point p and whose radius is the distance from p to the center $p(f)$ of f. If c does not intersect g_1 then this subcase does not limit the rotation of F about the pivot point p. However, if c does intersect g_1 then it will do so at two points, p_1 and p_2, which may degenerate to the same point. The larger of $\theta(p, p(f)) - \theta(p, p_1)$ and $\theta(p, p(f)) - \theta(p, p_2)$ is a limit on the clockwise rotation of F about the pivot point p while the larger of $\theta(p, p_1) - \theta(p, p(f))$ and $\theta(p, p_2) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. The position of f after the maximal clockwise rotation is depicted as f_1 in figure 9.15 while the position of f after the maximal counterclockwise rotation is depicted as f_2.

The circle f is inside g.
This subcase is depicted in figure 9.16. In this subcase the circle f is rotated until it is tangent to and inside the circle g. Construct a circle g_1, concentric with g, whose radius is the radius of g minus the radius of f. Construct a pivot circle c whose center is the pivot point p and whose radius is the distance from p to the center $p(f)$ of f. If c does not intersect g_1 then this subcase does not limit the rotation of F about the pivot point p. However, if c does intersect g_1

Figure 9.16: Rotating a circle f until blocked by another circle g when f is inside g.

then it will do so at two points, p_1 and p_2, which may degenerate to the same point. The larger of $\theta(p, p(f)) - \theta(p, p_1)$ and $\theta(p, p(f)) - \theta(p, p_2)$ is a limit on the clockwise rotation of F about the pivot point p while the larger of $\theta(p, p_1) - \theta(p, p(f))$ and $\theta(p, p_2) - \theta(p, p(f))$ is the corresponding limit in the counterclockwise direction. The position of f after the maximal clockwise rotation is depicted as f_1 in figure 9.16 while the position of f after the maximal counterclockwise rotation is depicted as f_2.

The circle g is inside f.
This subcase reduces to the second subcase by rotating g in the opposite direction until blocked by f. Clockwise limits become counterclockwise limits and vice versa.

9.3 Complications

The algorithm presented in the previous two sections is only a framework for kinematic simulation. It handles only the general cases, not the complications caused by the many anomalous special cases that arise during actual use of the simulator to support analysis of animated stick figure movies like the one described in section 6.1. This section discusses some of these complications and how to deal with them. During the development of ABIGAIL the process of discovering that these anomalous cases existed, and

then determining how to correctly deal with them, was substantially more difficult and took significantly more time and effort than implementing the general case. One may ask whether it is necessary to handle all of these special cases. Many of these special cases were discovered because the event perception mechanism built on top of the imagination capacity would produce the wrong results due to incorrect handling of these anomalous cases. For example, prior to dealing with roundoff errors, objects would mysteriously and unpredictably fall through the floor for reasons which will be discussed in section 9.3.4.

9.3.1 Clusters

As described in section 9.1, at each step during imagination, the kinematic simulator will attempt to translate or rotate a single set of figures, the foreground, leaving the remaining figures, the background, stationary. Foregrounds were chosen as connected components in the connection graph of the image, i.e. sets of figures connected by joints. Figure 9.17(a) depicts problems that arise with this simple choice of foregrounds. The figure shows two interlocking yet distinct objects, A and B. Since they are not joined together they constitute separate connected components and will be considered as separate foregrounds for translation and rotation. However, when attempting to translate A downward alone, B blocks any downward motion of A. Likewise, when attempting to translate B downward alone, A blocks any downward motion of B. Thus neither A nor B will fall when simulated. They will remain suspended in mid-air. This same situation happens not only for the case of falling; it can happen for all of the types of movement considered in section 9.1. This includes falling down, falling over, sliding along a linear or circular surface, and varying a joint's flexible rotation and translational or rotational displacement parameters. Figure 9.17(b) depicts two objects jointly sliding down an inclined plane. Figure 9.17(c) depicts two objects jointly falling over. Figure 9.17(d) shows how the problem can arise when varying the flexible rotation parameter of a joint which would jointly pivot two interlocking objects about that joint. It occurs even without interlocking objects. The heavy ball in figure 9.17(e) will not push the see-saw down since the ball and see-saw are distinct connected components and thus they will not rotate together around the pivot. The see-saw prevents downward movement of the ball. Yet the see-saw alone will not rotate since rotating it alone will increase its potential energy.

The solution to this problem is conceptually simple. Treat A and B together as a single foreground called a *cluster*. More generally, the solution can be stated as follows. Form all connected components $F_1, \ldots, F_n$ in the connection graph of the image. Two connected components are said to touch if some figure from one touches some figure from the other. Consider as a foreground, all clusters F that are union sets of a collection of connected components, i.e. $F_{i_1} \cup \cdots \cup F_{i_m}$, where the collection of connected components is itself connected by the component touching relation. When varying a flexible parameter of a joint j, only clusters which do not contain both $f(j)$ and $g(j)$ are considered.

The above solution has a drawback, however. It becomes intractable when there is a large set of connected components that are connected by the touching relation since every subset of that set which is still connected by the touching relation must be considered as a cluster. This situation does arise in practice in at least one case. ABIGAIL begins watching a movie with an empty joint model. Objects containing many figures which will later be treated as a single connected component due to joints not yet hypothesized will initially be treated as clusters. While two joined figures will always be considered as part of the same foreground, two touching but unjoined figures are only optionally considered as part of the same cluster. Such nondeterminism in the choice of clustered foregrounds with an empty joint model leads to intractability in the kinematic simulator. This intractability is eliminated in the current implementation by forming clusters only once an initial joint model has been formulated.

Figure 9.17: These situations require cluster movement. When attempting to move either object A or B alone, one will block any motion of the other yielding anomalous simulation results where objects A and B remain suspended but unsupported. The solution is to treat A and B as a single clustered foreground and attempt to translate or rotate them together.

9.3.2 Tangential Movement

Section 9.2 presented analytic methods for calculating the maximal amount that one figure can translate or rotate until blocked by another figure. The methods presented dealt only with the non-degenerate cases. Some of the computations required finding the intersection between a translation ray and a line segment. What happens if the ray is coincident with the line segment? In this case, they intersect at infinitely many points. This degenerate case can arise when one line segment slides along another. Other computations require finding the intersection between a pivot circle and another circle. What happens if the two circles are concentric and equiradial? In this case again, they intersect at infinitely many points. This degenerate case can arise when pivoting a line segment that lies inside a circle about the center of the circle, so that its endpoint slides along the interior wall of the circle.

In general, all such degenerate cases involve movement tangent to some surface. Though the above cases of tangential movement resulted in degenerate computation of intersection points, tangential movement need not produce such degeneracies. One example of such a situation would be the translation of a line segment until its endpoint was blocked by a circle. If the translation ray is tangent to the circle, it intersects the circle at one point instead of two. Sometimes, a surface that is tangent to the direction of motion does not block motion of the foreground. The first two examples are illustrations of such situations. In other situations, a surface that is tangent to the direction of motion can block motion of the foreground. The third example depicts such a situation. Each of the eight cases discussed in section 9.2, and all of their subcases, must be analyzed in detail to determine when the background blocks tangential movement of the foreground, and when it does not. Detailed analysis of each of these cases has demonstrated that in all cases where f does not touch g, if g would limit tangential movement f then that movement would be even further limited by some other non-tangential case. Thus the limits introduced by tangential movement can be ignored when f does not touch g. When f touches g, however, g may or may not totally limit any tangential movement of f depending on the situation. This analysis for each of the ten irreducible subcases is summarized below and depicted in figures 9.18 and 9.19.

Translating a line segment f until its endpoint $p(f)$ touches another line segment g.
Tangential movement arises in this subcase when the translation ray r is coincident with g. A line segment g never limits tangential movement of another line segment f. This case is depicted in figure 9.18(a).

Translating a circle f until it is tangent to a line segment g.
Tangential movement arises in this subcase when the translation ray r is coincident with either g_1 or g_2. This subcase never limits tangential movement. This subcase is depicted in figure 9.18(b).

Translating a line segment f until its endpoint $p(f)$ touches a circle g.
Tangential movement arises in this subcase when the translation ray r is tangent to circle g. This subcase limits tangential movement only when f is inside g. This is the case only when $|\theta(f) - \theta(p_1, p(g))| < \frac{\pi}{2}$ when normalized. The subcase where g blocks f is depicted in figure 9.18(e), while the subcase where g does not block f is depicted in figure 9.18(c).

Translating a circle f until blocked by a circle g when f and g are outside each other.
Tangential movement arises in this subcase when the translation ray r is tangent to g_1. This subcase never blocks tangential movement. This subcase is depicted in figure 9.18(d).

Translating a circle f until blocked by another circle g when f is inside g.
Tangential movement arises in this subcase when the translation ray r is tangent to g_1. This subcase always blocks tangential movement. This subcase is depicted in figure 9.18(f).

Figure 9.18: An analysis of all cases where the translation of the foreground figure f is tangential to the background figure g. In cases (e) and (f) g blocks movement of f while in the remaining cases it does not.

Figure 9.19: An analysis of all cases where the rotation of the foreground figure f is tangential to the background figure g. In cases (b), (d), (e), (i), (j), (k), (m), and (o) g blocks movement of f while in the remaining cases it does not.

Rotating a line segment f until its endpoint $p(f)$ touches another line segment g.
Tangential movement arises in this subcase when the pivot circle c is tangent to g. This subcase limits tangential movement only when $|\theta(f) - \theta(p(f), p)| > \frac{\pi}{2}$ when normalized. A boundary case arises when $|\theta(f) - \theta(p(f), p)| = \frac{\pi}{2}$. This boundary case will be discussed in section 9.3.3. The subcase where g blocks f is depicted in figure 9.19(b), while the subcase where g does not block f is depicted in figure 9.19(a).

Rotating a circle f until it is tangent to a line segment g.
Tangential movement arises in this subcase when the pivot circle c is tangent to either g_1 or g_2. This subcase limits tangential movement only when p and $p(f)$ are on opposite sides of g or when p is closer to g than $p(f)$. These two subcases where g blocks f are depicted in figure 9.19(d) and 9.19(e) respectively, while the subcase where g does not block f is depicted in figure 9.19(c).

Rotating a line segment f until its endpoint $p(f)$ touches a circle g.
Tangential movement arises in this subcase when the pivot circle c is tangent to g. There are three subcases to consider.

c is inside g.
This subcase limits tangential movement only when f is outside g. This is the case only when $|\theta(p(f), p(g)) - \theta(f)| > \frac{\pi}{2}$ when normalized. The subcase where g blocks f is depicted in figure 9.19(k), while the subcase where g does not block f is depicted in figure 9.19(h).

g is inside c.
This subcase limits tangential movement only when f is inside g. This is the case only when $|\theta(p(f), p(g)) - \theta(f)| < \frac{\pi}{2}$ when normalized. The subcase where g blocks f is depicted in figure 9.19(j), while the subcase where g does not block f is depicted in figure 9.19(g).

g and c are outside each other.
This subcase limits tangential movement only when f is inside g. This is the case only when $|\theta(p(f), p(g)) - \theta(f)| < \frac{\pi}{2}$ when normalized. The subcase where g blocks f is depicted in figure 9.19(i), while the subcase where g does not block f is depicted in figure 9.19(f).

Rotating a circle f until blocked by a circle g when f and g are outside each other.
Tangential movement arises in this subcase when the pivot circle c is tangent to g_1. This subcase limits tangential movement only when c is inside g_1. This is the case only when $\Delta(p(g), p) < \Delta(p(g), q(g)) + \Delta(p(f), q(f))$. The subcase where g blocks f is depicted in figure 9.19(m), while the subcase where g does not block f is depicted in figure 9.19(l).

Rotating a circle f until blocked by another circle g when f is inside g.
Tangential movement arises in this subcase when the pivot circle c is tangent to g_1. This subcase limits tangential movement only when c is outside g_1. This is the case only when $\Delta(p(g), p) > \Delta(p(g), q(g)) - \Delta(p(f), q(f))$. The subcase where g blocks f is depicted in figure 9.19(o), while the subcase where g does not block f is depicted in figure 9.19(n).

9.3.3 Touching Barriers

Section 9.2 presented analytic methods for calculating the maximal translation or rotation of one figure until blocked by another figure. The methods presented dealt only with the non-degenerate case of movement by some nonzero ϵ. When however, a figure f to be moved touches a figure g, g may prevent *any* movement of f along a given axis or in a given direction about a given pivot. In such cases, the analytic methods from section 9.2 will yield $\epsilon = 0$. If movement of f is indeed blocked by g then this is the correct solution. But there are cases where the analytic methods incorrectly yield $\epsilon = 0$ even

Figure 9.20: The analytic methods for determining maximum translation and rotation yield $\epsilon = 0$ when two figures touch. Sometimes movement is indeed blocked in this situation, as in (a), while other times, movement is not blocked, as in (b).

though f is *not* blocked by g. This happens when f touches g but is on the other side of g given its direction of movement. Figure 9.20 shows how this can arise when translating one line segment relative to another line segment. In figure 9.20(a), g blocks translation of f while in figure 9.20(b), g does not block translation of f. Analogous situations occur when translating or rotating any figure type relative to any other figure type.

To deal with this problem, the analytic methods must be augmented to determine whether g is or is not a barrier to the movement of f when they would otherwise yield $\epsilon = 0$. All of the cases and subcases can be handled by the same general technique which operates as follows. The maximal movement ϵ will be limited to zero only when f and g touch. Denote their point of contact by q. Form a line l through q as follows. If g is a line segment then it is extended to form l. If g is a circle then l is the line tangent to g at point q. This *barrier line* divides the plane into two half-planes. The figure f will lie in at most one of these half-planes. Let ϕ be the direction of the movement of f. A ray projected from q in the direction ϕ will also lie in at most one half-plane. The figure g blocks the movement of f only when f does not lie in the same half-plane as that ray. Applying this technique to each case and subcase requires determining both the half-plane in which f lies as well as the direction of movement ϕ. The former depends on the shape of f. If f is a line segment, one endpoint lies on l at the point of contact q. The other endpoint occupies the same half-plane as all of f. If f is a circle, its center $p(f)$ occupies the same half-plane as all of f. Thus determining the half-plane occupied by a figure f can be determined by examining a single point which I will denote as q'. When translating f along an axis θ, the direction of movement ϕ is the same as θ. When rotating f about a pivot point p, the direction of movement is given by the direction of a ray projected from the contact point q tangent to a circle c whose center is p and whose radius is the distance from p to q. For clockwise rotation this is $\theta(p,q) - \frac{\pi}{2}$ while for counterclockwise rotation this is $\theta(p,q) + \frac{\pi}{2}$.

Given a barrier line l, a direction of movement ϕ, and a point q', g blocks the movement of f only when a ray projected from q', along the axis ϕ, intersects l. When applying this check to each of the cases and subcases one must remember that some of the cases determine whether g blocks the movement of f by determining whether f blocks the movement of g is the opposite direction. Each case and subcase must take this into account when computing the parameters l, ϕ, and q' for this check procedure.

The above check whether g blocks the movement of f can be viewed as a boundary case of the more general case of movement discussed in section 9.2. This boundary case itself has two boundary cases.

One occurs when the direction of movement ϕ is parallel to the barrier line l. In this case, neither half-plane is in front of or behind the figure f. This case is covered by the tangential movement cases discussed is section 9.3.2. The other occurs when q' lies on the barrier line l. In this case, f does not lie in either half-plane. An ambiguity arises as to which side of l figure f lies on. This can only happen when f is a line segment. When g is a circle, f can only move in a direction that will keep it outside g. Analytic methods similar to those discussed above can determine the allowed direction of movement. When g is a line segment, however, a genuine ambiguity arises. This can only happen when f is coincident with g as is depicted in figure 9.21(a). In this case it is genuinely ambiguous as to which side of g the figure f lies on. This situation therefore admits only two consistent interpretations. Either g blocks or doesn't block f uniformly for any type of movement. Adopting the latter interpretation would lead to problems since objects then could fall through the floor. Adopting the former interpretation, however, leads to the anomalous situation depicted in figure 9.21(b) where John falls on his knee, but doesn't fall any further, since his calf, being coincident to the ground, cannot rotate or translate. ABIGAIL adopts the latter alternative, thus exhibiting this anomaly. A solution to this problem would require modifying the procedures described in section 9.3.2 to examine the context of two figures, i.e. other figures connected to either the foreground f or the background g, when determining whether g blocks movement of f.

9.3.4 Tolerance

All of the procedures described in sections 9.2, 9.3.2, and 9.3.3 must be modified to deal with roundoff error. Roundoff error can introduce gross substantiality violations in the resulting simulation as depicted in figure 9.22. Figure 9.22(a) depicts a line segment f falling toward a line segment g. If the limit calculation has roundoff error, it can produce a situation, depicted in figure 9.22(b), where f is translated slightly too far. In the next step of the simulation, however, the endpoint of f is now past g and thus a translation ray projected from that endpoint will not intersect g. Thus g limits the translation of f only to the position indicated in figure 9.22(c). At this point, f can fall away from g, as in figure 9.22(d), since in figure 9.22(c), g does not block f in its direction of movement. Thus due to slight roundoff error in the transition from figure 9.22(a) to figure 9.22(b), f is able to pass through g. As figure 9.22 shows, roundoff error can introduce gross deviations from the desired simulation, not just minor differences. Accordingly, ABIGAIL incorporates a notion of tolerance whenever determining whether two figures touch, so that figure 9.22(b) is interpreted as an instance of touching barriers to be handled via the methods described in section 9.3.3. Furthermore, the methods described in section 9.2 must be modified in this case to return $\epsilon = 0$ even though the translation ray does not intersect g.

9.4 Limitations

The kinematic simulator just presented suffers from a severe limitation. It can only collectively translate or rotate one group of figures at a time. Such collective movement can correctly simulate either rigid body motion, or the motion of a non-rigid mechanism where only a single joint parameter changes. It is not able to correctly simulate the behavior of mechanisms which require that different collections of figures simultaneously move along different paths. Several such mechanisms are shown in figures 9.23 and 9.24.

The mechanism in figure 9.23 contains two line segments f_1 and f_2, fastened at the endpoints $p(f_1)$ and $p(f_2)$ by a joint j with flexible rotation and rigid displacement parameters. The endpoints $q(f_1)$ and $q(f_2)$ are supported on the ground. Since the micro-world ontology lacks any notion of friction, the endpoints $q(f_1)$ and $q(f_2)$ should slide apart along the ground while the flexible joint rotation $\theta(j)$ increases until both f_1 and f_2 lie flat on the ground. ABIGAIL, however, is not able to predict this motion since it requires simultaneously rotating the line segments f_1 and f_2 in opposite directions, as well as

Figure 9.21: An ambiguous situation occurs when the foreground f and background g are two coincident line segments. In this situation it is not possible to determine on which side of the background the foreground lies. Because of this ambiguity, ABIGAIL will neither translate nor rotate f relative to g for fear of violating substantiality as depicted in (a). A case where this arises in practice is depicted in (b). Once John falls on his knee he will not fall any further, since his calf, being coincident with the ground, cannot rotate or translate.

Figure 9.22: Roundoff errors during simulation can cause substantiality violations and result in gross deviations from the desired simulation. Here an object f falls toward an object g. Ordinarily g should block the fall of f. Roundoff error during step (b), however, causes a substantiality violation. Since the endpoint of f is now past g, a translation ray projected from that endpoint will not intersect g and thus g will limit the movement of f only until the position indicated in (c). Since in (c), g does not block f in its direction of movement, f can fall from g as in (d). Thus due to the roundoff error in (b), f falls through g.

Figure 9.23: A mechanism whose behavior ABIGAIL cannot predict. This mechanism has two line segments f_1 and f_2, and a single joint j, where $f(f) = f_1$, $g(j) = f_2$, $\theta(j)$ is flexible, $\delta_f(j) = 0$ and $\delta_g(j) = 0$. The endpoints $q(f_1)$ and $q(f_2)$ should slide along the ground while $\theta(j)$ increases until f_1 and f_2 lie flat on the ground. ABIGAIL is not able to predict such motion since it requires the simultaneous rotation and translation of f_1 and f_2 along different paths.

Figure 9.24: A four bar linkage. Using the terminology of this thesis, this linkage can be described as four line segments $f_0, \ldots, f_3$ and four joints $j_0, \ldots, j_3$ where for $i = 0, \ldots, 3$, $f(j_i) = f_i$, $g(j_i) = f_{i+1 \bmod 4}$, $\delta_f(j_i) = 0$, $\delta_g(j_i) = 1$, and $\theta(j_i)$ is flexible. ABIGAIL cannot predict the behavior of such linkages since changing the rotation parameter of any joint would require the simultaneous motion of at least three line segments along different paths.

translating them collectively downward, in order to decrease the potential energy of the mechanism. Any one of these movements alone will increase the potential energy so no movement will be attempted.

The mechanism in figure 9.24 is a classic *four bar linkage*. It contains four line segments $f_0, \ldots, f_3$ joined at their endpoints by four joints $j_0, \ldots, j_3$ with flexible rotation and rigid displacement parameters. Assuming that one of the line segments has a fixed position and orientation, changing the rotation parameter of any one of the joints will cause all of the joint rotation parameters to change and the remaining line segments to translate and rotate along different paths.

Both of these mechanisms share a common property. They have a cycle in their connection graph.[9] The cycle in figure 9.24 is apparent. The cycle in figure 9.23 results from the fact that due to the ground plane constraint, the mechanism behaves as if the ground was a line segment g and figures f_1 and f_2 were joined to g by joints with flexible rotations, rigid displacements along f_1 and f_2, and flexible displacements along g.

ABIGAIL can only accurately predict the behavior of mechanisms whose connection graphs do not contain cycles.[10] This includes both explicit cycles due to joints as well as implicit cycles due to the ground plane and substantiality constraints. This means that the kinematic simulator used to implement ABIGAIL's imagination capacity is not cognitively plausible since people can understand the behavior of such mechanisms. While a person might not be able to accurately calculate the exact quantitative relationship between the motion of parts A and B in mechanism shown in figure 9.25, she nonetheless could at least predict that pushing A will cause B to move and perhaps even predict the direction of motion.

9.5 Experimental Evidence

Spelke (1988) reports a number of experiments that illuminate the nature of infant visual perception. Most of these experiments use the paradigms of habituation/dishabituation and preferential looking

[9] The connection graph of a mechanism is a graph where the figures constitute the vertices and there is an undirected edge between two vertices if their corresponding figures are joined.

[10] She can still watch movies that depict such mechanisms without breaking. She will just treat a cyclic mechanism as a rigid body.

Figure 9.25: ABIGAIL's imagination capacity is impoverished with respect to human imagination capacity. While humans can predict that pushing A will cause B to move, ABIGAIL cannot make such a prediction since the connection graph of this mechanism contains cycles and the kinematic simulator used to implement ABIGAIL's imagination capacity cannot handle cycles.

as windows on infant perception. A general property of the nervous system is that it *habituates* to repeated stimuli. The level of response elicited from repeated applications of similar stimuli decreases when compared with the initial application of the stimulus. After habituation however, application of a novel stimulus will again elicit a greater level of response. Since this *dishabituation* happens only for novel stimuli it can be used as a probe to determine whether two stimuli are characterized as similar or different. The experimental framework is as follows. Subjects are first habituated to stimulus A and then exposed to stimulus B. Alternatively, they are habituated to A and then exposed to C. A greater level of dishabituation for C than for B is taken as evidence that B is classified as more similar to A than C is. In the case of infants, the response level is often measured by *preferential looking*, measuring the amount of time they look at a presented stimulus, or at one stimulus versus another.

Spelke reports two experiments which give evidence that by age five months, children are aware of the substantiality constraint. The first experiment was originally reported by Baillargeon et al. (1985). This experiment is illustrated in figure 9.26. Infants were habituated to a scenario depicting a screen. Initially the screen lay flat on its front. Subsequently, it lifted upwards and rotated backwards until it lay flat on its back. Finally, its motion was reversed until it again lay flat on its front. To make this motion clear, both front and side views are depicted in figure 9.26(a), though the actual stimulus in the experiment contained only the front view. The two dishabituation stimuli are shown in figures 9.26(b) and 9.26(c). In both, a block is situated behind the screen such that it is occluded as the screen is raised. The first depicts a possible event: the screen only rotates as far back as it can without penetrating the occluded block. The second depicts an impossible event: the screen continues to rotate 180°. Unless the block disappears, this would constitute a substantiality violation. Five-month-old infants dishabituate more to the latter scenario than the former. This is interpreted as evidence that they interpret both scenarios (a) and (b) as normal but scenario (c) as abnormal. Baillargeon (1987) reports continued experiments along these lines which show that children are attentive to substantiality violations by age $4\frac{1}{2}$-months and perhaps even by age $3\frac{1}{2}$-months. Baillargeon (1986) reports additional experiments which show that children take the location of hidden objects into account in their desire to uphold the substantiality constraint.

Spelke reports a similar experiment performed jointly with Macomber and Keil on four-month-old infants. This experiment is depicted in figure 9.27. Here, the infants were habituated to the following scenario. An object was dropped behind a screen. The screen was then lifted to reveal the object lying on the ground as shown in figure 9.27(a). The two dishabituation stimuli are shown in figure 9.27(b) and 9.27(c). In both, a table appears in the path of the falling object when the screen is removed. The first depicts the object lying on the table—a different position than in the habituation scenario. The second depicts the object lying underneath the table—in the same position as in the habituation scenario—yet one which cannot be reached without a substantiality violation. Four-month-old infants dishabituate more to the latter scenario than the former, again giving evidence that they are cognizant of the substantiality constraint by age four months.

Spelke reports that Macomber performed a variation of the previous experiment in attempt to determine the age at which infants know about gravity. This variation is depicted in figure 9.28. Infants were habituated to an object falling behind a screen with the screen being removed to reveal the object lying on a table. In both dishabituation stimuli, the table top was removed. In the first dishabituation stimulus, removing the screen revealed the object at rest on the floor, beneath its original position on the table top, while in the second, removing the screen revealed the object at the same position as it was in the habituation scenario. This time however, the object was suspended unsupported in mid-air due to the disappearance of the table top. Spelke reports that four-month-old infants dishabituate more to the former scenario than the latter, implying that they do not yet form correct judgments based on gravity and support.

At some point however, children do come to possess knowledge of gravity and support. The only question is at what point they do so. I conjecture that such development happens early. If the analysis

Figure 9.26: Displays for an experiment demonstrating infant knowledge of substantiality. (Figure 7.7 from Spelke (1988).) Infants habituated to sequence (a) dishabituate more to sequence (c) than to sequence (b). Since sequence (c) depicts a substantiality violation, this is interpreted as evidence that five-month-old children have knowledge of substantiality.

Figure 9.27: Displays for an experiment demonstrating infant knowledge of substantiality. (Figure 7.8 from Spelke (1988).) Infants habituated to sequence (a) dishabituate more to sequence (c) than to sequence (b). Since sequence (c) depicts a substantiality violation this is interpreted as evidence that four-month-old children have knowledge of substantiality.

Figure 9.28: Displays for an experiment testing infant knowledge of gravity. (Figure 7.9 from Spelke (1988).) The conjecture was that infants habituated to sequence (a) would dishabituate more to sequence (c) than to sequence (b), since sequence (c) depicts an unsupported object. This expected result was not exhibited by four-month-old infants.

from chapter 7 is correct, and the meanings of so many everyday simple spatial motion verbs depend on the concepts of gravity and support, then the knowledge of gravity and support must precede the onset of language acquisition.

Spelke reports a fourth experiment, done jointly with Kestenbaum, that gives evidence that by age four months, children know that objects must obey continuity. This experiment is depicted in figure 9.29. Two groups of subjects participated in this experiment. The first group was habituated to the scenario depicted in figure 9.29(a). In this scenario, an object passed behind one screen, as it moved from left to right, emerged from behind that screen, and then passed behind and emerged from a second screen. The second group was habituated to a similar scenario except that no object appeared in the gap between the screens. An object passed behind one screen and then emerged from the second, as depicted in figure 9.29(b). Both groups received the same two dishabituation stimuli shown in figures 9.29(c) and 9.29(d). One simply showed a single object without the screens while the other showed two objects without the screens. The group habituated to (a) dishabituated more to (d) while the group habituated to (b) dishabituated more to (c). The subjects appear to attribute scenario (a) to a single object while attributing scenario (b) to two objects. This is interpreted as evidence that by age four months, children know that objects must move along continuous paths, and furthermore, a single object cannot follow a continuous path without being visible in between the screens.

These experiments reported by Spelke demonstrate that infants at a very early age possess knowledge of substantiality and continuity. Furthermore, they use this knowledge as part of object and event perception. She offers the following claim.

> The principles of cohesion, boundedness, substance and spatio-temporal continuity appear to stand at the centre of adults' intuitive conceptions of the physical world and its behaviour: our deepest conceptions of objects appear to be the notions that they are internally connected and distinct from one another, that they occupy space, and that they exist and move continuously (for further discussion, see Spelke 1983, 1987). These conceptions are so central to human thinking about the physical world that their uniformity sometimes goes unremarked. In studies of intuitive physical thought, for example, much attention is paid to the idiosyncratic and error-ridden predictions adults sometimes make about the motions of objects (e.g. McCloskey 1983). It is rarely noted, however, that adults predict with near uniformity that objects will move as cohesive wholes on connected paths through unoccupied space. This conception, at least, is clear and central to our thinking; it appears to have guided our thinking since early infancy.
>
> [p. 181]

She then goes on to suggest that the physical knowledge which underlies object and event perception precedes linguistic development.

> In this context, one may consider the possible role of language in the development of physical knowledge. Our research provides evidence, counter to the views of Quine (1960) and others, that the organization of the world into objects precedes the development of language and thus does not depend upon it. I suspect, moreover, that language plays no important role in the spontaneous elaboration of physical knowledge. To learn that objects tend to move at smooth speeds, for example, one need only observe objects and their motions; one need not articulate the principles of one's theory or communicate with others about it.
>
> [p. 181]

Spelke's work attempts to refute the claim that linguistic ability is needed to formulate physical knowledge. This thesis carries Spelke's argument one step further. It suggests that physical knowledge is needed to formulate linguistic concepts.

Figure 9.29: Displays for an experiment demonstrating infant knowledge of continuity. (Figure 7.10 from Spelke (1988).) Infants habituated to sequence (a) dishabituate more to sequence (d) than to sequence (c), while infants habituated to sequence (b) dishabituate more to sequence (c) than to sequence (d). This is interpreted as evidence that five-month-old children have knowledge that sequence (a) involves the continuous motion of one object, while sequence (b) must involve two objects.

9.6 Summary

In chapter 7, I argued that the notions of support, contact, and attachment play a central role in the definitions of simple spatial motion verbs such as *throw*, *pick up*, *put*, and *walk*. In chapter 8, I presented a theory of how these notions can be grounded in perception via counterfactual simulation. A simple formulation of this theory has been implemented as a computer program called ABIGAIL that watches movies constructed out of line segments and circles and uses counterfactual simulation to produce descriptions of the objects depicted in those movies, along with the changing status of support, contact, and attachment relations between those objects. In this chapter I have argued that counterfactual simulation is performed by a modular imagination capacity which directly encodes naive physical knowledge such as the substantiality, continuity, gravity, and ground plane constraints. I have argued that by being based on these principles, the human imagination capacity, operates in a very different fashion from conventional kinematic simulators. The incremental stepwise behavior of traditional kinematic simulators is both slow and cognitively implausible since it does not faithfully reflect the substantiality and continuity constraints. This chapter has presented an alternate simulation mechanism, which for a limited class of mechanisms, can directly predict in a single step that objects fall along continuous paths until they collide with obstacles in their path of motion. This mechanism appears better suited to the task of recovering support, contact, and attachment relations since the recovery of these relations appears to be based more on collision detection than on physical accuracy. Perhaps that is why human visual perception is more sensitive to the notions of substantiality and continuity than to velocity, momentum, and acceleration. While these mechanisms have to date been implemented only for the drastically simplified ontology of ABIGAIL's micro-world, it appears that similar, though probably much more complex variants of these mechanisms form the basis of the imagination capacity which drives human visual perception. Extending the mechanisms explored with ABIGAIL to deal with more complex world ontologies remains for future work.

Chapter 10

Conclusion

10.1 Related Work

Computer models of event perception are not new. A number of previous attempts at producing event descriptions from animated movies have been described in the literature. Thibadeau (1986) describes a system that processes the movie created by Heider and Simmel (1944) and determines when events occur. The Heider and Simmel movie depicts two-dimensional geometric objects moving in a plane. When viewing that movie, most people project an elaborate story onto the motion of abstract objects. Thibadeau's system does not classify event types. It just produces a single binary function over time delineating when an 'event' is said to have occurred. Badler (1975) describes an unimplemented strategy for processing computer-generated animated line drawings to recover event descriptions. Badler's proposed system hierarchically recognizes predicates which are true over successively longer segments of the movie. His proposed system does not incorporate counterfactual simulation. The lowest level predicates are computed geometrically on figures in a single frame of the movie. He thus does not have accurate methods for deriving support, contact, and attachment relations between objects. Adler (1977), Tsotsos (1977), Tsotsos and Mylopoulos (1979), Tsuji et al. (1977), Tsuji et al. (1979), Okada (1979), and Abe et al. (1981) describe systems similar to Badler's. Again these systems do not incorporate counterfactual simulation and do not derive support, contact, and attachment relations between objects. Novak and Bulko (1990) describe a system for interpreting drawings depicting physics problems. Their system uses the linguistic description of the problem as an aid to the process of understanding the image. It cannot correctly interpret the image without the help of the linguistic description and thus unlike ABIGAIL, cannot be used as a model of the event perception mechanism that provides the non-linguistic input to the language acquisition device.

Kinematic simulation is also widely discussed in the literature, though it has never been applied to the task of event perception. While most of the work falls within the classic approach of numerical integration, two notable exceptions are the work of Kramer and Funt.

10.1.1 Kramer

Kramer (1990a, 1990b) discusses a kinematic simulator called TLA. Like this thesis, Kramer eschews the classic approach based on numerical integration in favor of a more closed-form solution. He does so, however, for reasons of efficiency. Kramer is not concerned with cognitive modeling and plausibility. Like ABIGAIL, TLA ignores dynamics. This includes velocity, momentum, kinetic energy, and the magnitude of forces acting on components.

On one hand, TLA is substantially more powerful than ABIGAIL. Besides simulating three-dimensional

movement constrained by a wide variety of joint types, TLA is able to handle closed-loop kinematic chains. Kramer presents TLA simulating a number of complex mechanisms including a sofabed. It does so by constructing what Kramer calls an *assembly plan*, a procedure for incrementally satisfying the joint constraints of a mechanism, one by one, in a fashion which is analogous to assembling the mechanism in a given configuration. When a mechanism contains a closed-loop kinematic chain, there are constraints between the values of its joint parameters. Some independent set of joint parameters is taken as the *driving inputs* so that the values of the remaining dependent joint parameters is uniquely determined given particular values for those inputs. An assembly plan is thus a procedure for computing the values of dependent joint parameters from these driving inputs.[1] TLA operates by repeatedly assembling a mechanism for different values of the driving inputs. When a mechanism does not contain any closed-loop kinematic chains, its assembly plan is trivial. All of its flexible joint parameters are driving inputs and none are computed as dependent results. In essence ABIGAIL handles just this simple case. The novel contribution of TLA is an algorithm for deriving assembly plans for mechanisms with closed-loop kinematic chains.

On the other hand, ABIGAIL addresses issues that do not concern Kramer. Even ignoring dynamics, the motion of objects must obey a number of constraints in addition to those imposed by joints. These include substantiality, continuity, gravity, and ground plane, none of which are handled by TLA. In essence, TLA is an extremely sophisticated and competent analog of the inner loop of ABIGAIL's simulator which moves the foreground relative to the background. In ABIGAIL this inner loop is trivial since she does not deal with closed-loop kinematic chains. The focus in ABIGAIL however, is on what is built on top of this inner loop—the mechanism for repeatedly choosing a foreground, deciding whether to translate of rotate this foreground, determining an appropriate translation axis θ or pivot point p, and most importantly *analytically determining how far to translate or rotate the foreground along that translation axis or pivot point until potential energy would increase or substantiality would be violated.* This is one novel contribution of the kinematic simulator incorporated into ABIGAIL, apart from all of the higher-level mechanisms which use that simulator to support event perception and the grounding of language in perception.

One may consider merging the two ideas together in an attempt to allow ABIGAIL to understand images that contain closed-loop kinematic chains. This is actually *much* more complicated than it would seem at first glance. In ABIGAIL's ontology, all motion follows either linear or circular paths. Furthermore, all objects are constructed from line segments and circles. Thus all motion limits can be found by computing the intersection of lines and circles. This is conceptually straightforward despite the myriad of cases, subcases, and boundary cases which must be considered to make it work. As the driving inputs of a mechanism with closed-loop kinematic chains are varied, however, their components follow paths which are substantially more complex as they move. Merging TLA and ABIGAIL would first require that TLA compute a representation of the path a point on an object would follow as a result of varying a driving input. Currently, TLA does not compute such representations. It only computes individual positions along the path given particular values for the driving inputs. Even if an explicit representation of paths were produced, two further capabilities are needed to incorporate such a capacity into the simulation framework discussed in section 9.1. First, a method is needed to compute how far one can vary a driving input while still decreasing potential energy. Second, a method is needed for intersecting arbitrary paths. My guess is that this would be a substantial endeavor.

It is not clear that such an effort would be worthwhile. People might not have the ability to accurately simulate complex mechanisms as part of the hypothesized imagination capacity. While they clearly can predict, at least at a gross level, the behavior of mechanisms such as the ones in figures 9.23, 9.24, and 9.25, they might do so by some approximation method which removes the closed-loop kinematic

[1] A given set of joint parameters may be sufficient for uniquely determining a mechanism's configuration for some values of the parameters but not others. Thus an assembly plan must be flexible about which joint parameters it takes as driving inputs and which parameters it returns as computed results.

chains. How this may be done is a topic for future research.

10.1.2 Funt

Funt (1980) describes a system called WHISPER that shares many of the same goals as ABIGAIL's imagination capacity. Like ABIGAIL, WHISPER can determine the support relationships between objects in a static image. WHISPER can also predict the sequence of events that will occur during the collapse of a pile of objects depicted in a static image. WHISPER differs from ABIGAIL in one key detail however. While ABIGAIL represents images as collections of line segments and circles, WHISPER instead represents images as bitmaps. Thus, unlike ABIGAIL, WHISPER can represent and operate on images containing arbitrarily shaped objects.

WHISPER maintains two distinct bitmap representations of each image. One uses a conventional rectilinear layout of pixels. Funt calls this representation the *diagram*. The other uses a concentric layout of pixels which Funt calls the *retina*. Various transformation operations can be performed on an image in each representation. For example, objects in the diagram may be translated or rotated, a process which Funt calls redrawing the diagram. The concentric layout of the retina representation supports a number of efficient transformations, in particular rotation about the center of the retina. Funt allows the diagram representation to be converted to the retina representation but not vice versa. This process, called fixation, can specify a point in the diagram to be aligned with the center of the retina. Higher-level processes request sequences of fixation and transformation operations. These processes can also perform a number of query operations on the retina representation. Direct queries on the diagram are not supported. In addition to rotation about its center, the concentric layout of the retina representation supports several other efficient query operations. These include computing the center-of-area of an object, finding the points of contact between two objects, examining curves to find points of abrupt change in slope, determining whether an object is symmetric, and determining whether two objects have the same shape.

The higher-level supervisory processes determine support relationships and perform the simulation by issuing a sequence of transformations, fixations, and queries on the diagram and retina representations. In this respect WHISPER is very similar to ABIGAIL. Both ABIGAIL and WHISPER ignore dynamic effects of velocity, acceleration, momentum, moment of inertia, and kinetic energy during the simulation. Both assume that objects have a uniform density which allows equating center-of-mass with center-of-area. More importantly, both perform simulation by a sequence of single object translations and rotations, ignoring the possibility for simultaneous movement of multiple objects. Besides the inherent physical inaccuracy caused by this approach to simulation, WHISPER, like ABIGAIL, is unable to simulate scenarios with closed-loop kinematic chains.

Though WHISPER is similar in intent to ABIGAIL, and shares many of the same underlying assumptions and problems, WHISPER also differs from ABIGAIL in a number of key respects. First, as discussed previously, WHISPER uses a bitmap representation while ABIGAIL uses an edge-based representation. Second, WHISPER only performs simulations and determines support relationships. It does not perform the higher-level tasks of event perception which in ABIGAIL are built around the ability to perform such simulations by the methods described in chapter 8. Third, WHISPER's ontology is strictly two-dimensional. It lacks any notion of a third dimension, even a restricted one such as the concept of 'layer' incorporated into ABIGAIL. Furthermore, WHISPER's ontology does not include the capability for objects to be fastened together by joints. Since its ontology lacks joints and layers, it has no need to infer such information from the image and thus has no analog to the model updating process described in section 8.2.1. A fourth and more significant difference between ABIGAIL and WHISPER is that while ABIGAIL can determine analytically in a single step, the maximal rotation or translation an object can undergo subject to substantiality constraints, WHISPER operates more like a conventional simulator, repeatedly performing small transformations and checking for collisions after each transformation.

It is interesting to note that WHISPER incorporates a number of the same heuristics as ABIGAIL that limit the choice of pivot points and translation axes. Furthermore, WHISPER utilizes a notion of conglomeration—amalgamating several objects together for collective analysis of support relations—a concept which is analogous to that of clusters. Unlike ABIGAIL, WHISPER determines whether an object is supported without actually imagining it falling, by examining the relative positions of an object's center-of-mass and its support points. This method allows WHISPER to determine support relationships for some, but not all, situations where ABIGAIL would fail due to implied closed-loop kinematic chains.

10.2 Discussion

For pedagogical purposes, part II of this thesis has taken an extreme position on the representation of verb meanings. Chapter 7 has exaggerated the role played by the notions of support, contact, and attachment in order to motivate the event perception mechanisms presented in chapter 8 and 9. In doing so, it downplayed the notions of causality and force application which most prior approaches to lexical semantic representation (e.g. Miller 1972, Schank 1973, Jackendoff 1983, and Pinker 1989) have taken to be central to verb definitions. This thesis does not claim that the notions of support, contact, and attachment are sufficient to define verb meanings. Causality and force application, as well as numerous other notions, are needed to characterize word meanings in general, let alone the meanings of simple spatial motion verbs. Most of the words defined in chapter 7 (e.g *throw*, *fall*, *drop*, *bounce*, *jump*, *put*, *pick up*, *carry*, *raise*, *make*, *break*, *fix*, *step*, and *walk*) have clear causal components even though the definitions given there were able to circumvent the need for describing this causal component by sufficiently characterizing the non-causal aspects of the meanings of these verbs, namely the support, contact, and attachment relations they engender between objects participating in events that they describe. This ability for ignoring the causal component of verb meanings broke down for verbs like *roll* and *slide* in their transitive uses. Thus ultimately it will be necessary to incorporate causality into a comprehensive lexical semantic representation. Doing so will require an explanation of how to ground the notion of causality in perception.

It may be possible to extend the techniques described in chapters 8 and 9, namely counterfactual simulation, to support the perception of causality and force application. In essence, an object A can be said to cause an event e if e does actually happen in the observed world but does not happen in an imagined world where A either does not exist or moves differently than in the observed world. Imagining an alternate world without A can be accomplished using existing mechanisms in ABIGAIL. The notion of 'moving differently', however, requires extending ABIGAIL's ontology to support animate objects. Animate (or at least motile) objects are those which appear to move on their own initiative. Such motion occurs because parts of animate objects exert forces relative to other parts. Within the limited ontology of ABIGAIL's micro-world, such relative motion of animate object parts could be modeled completely using joints which exert forces to change their parameters. Currently, gravity is the only force incorporated into ABIGAIL. ABIGAIL could be extended to model joint forces in addition to gravity. This would require several changes. First, the joint model maintained by ABIGAIL must be extended to contain a representation of the changing forces exerted by each joint as a function of time. The changing force profile of the joints comprising an object A can be said to be the *motor program* executed by A. To model grasping and releasing, the motor program must have the capacity for representing the creation and dissolution of joints in addition to changing force profiles. Second, the imagination capacity must be extended to take a motor program as input in addition to a set of figures, a joint model, and a layer model. Such an extended imagination capacity would model the short-term future of the world under the effects of gravity assuming that each animate object executed the motor program given as input. Modeling the execution of motor programs would require a kinematic simulator that was more faithful to the time course of simulation than the simulator currently used by

ABIGAIL. Third, since the motor programs executed by animate objects in the world are not directly observable, ABIGAIL must be provided with mechanisms for hypothesizing these motor programs. Such mechanisms would be analogous to those currently used by ABIGAIL for updating her joint and layer models. Motor programs could be recovered by counterfactual simulation. Informally, ABIGAIL would incorporate into the hypothesized motor programs only those force applications which were needed to have the imagined world match the observed world. Finally, a primitive `(cause` x e`)` could be added to the lexical semantic representation described in chapter 7. Actually, there seem to be at least three distinct notions of causality. The first expresses the fact that the existence of an object x caused an event e. Such a causal relation is true if e occurs in the observed world but does not occur in a world imagined without x. Given this notion of causality, the two argument primitive `(supports` x y`)` can be reformulated as `(cause` x `(supported` y`))`. The second expresses the fact that the motion of an animate object x, namely the motion caused by the execution of its motor program, caused an event e. Such a causal relation is true if e occurs in the observed world but does not occur in a world imagined where x does not execute its motor program. During such counterfactual simulation, x would keep rigid all of the joints which it would have moved according to the motor program recovered from the observed world. The third variant of causality expresses the fact that the involuntary motion of an object x caused an event e. Such involuntary motion occurs not because of a motor program executed by x but rather as a result of either gravity, a motor program executed by some other object, or a combination of the two.

Putting these speculative ideas aside, there are several important areas of continued work along the main themes advanced in part II of this thesis. First, to date ABIGAIL has only processed a portion of a single movie. Additional work is needed to improve the robustness and performance of the imagination capacity and event perception mechanisms to allow ABIGAIL to successfully process many movies. Second, ABIGAIL currently does not produce complete semantic descriptions of event such as those presented in chapter 7. While she does recover perceptual primitives, including the notions of support, contact, and attachment, she does not aggregate these primitives into event expressions. It would be fairly straightforward to incorporate a lexicon of event expressions into ABIGAIL and have her continually assess which of these known event types were currently happening in the movie. A number of prior approaches to event perception (e.g. Badler 1975) utilized such a lexicon of event types. A more satisfying approach would not rely on a predefined set of event types but instead would be able to learn the appropriate event lexicon. The event lexicon might be acquired by noticing recurring sequences of perceptual primitives in the movie. Alternatively, there may be universal and perhaps innate principles that govern the aggregation of perceptual primitives into discrete events. Discerning the nature of such principles and testing their validity by building computational models awaits further research. Finally, ABIGAIL is currently not integrated with any language processing facility. The original goal that motivated the work on event perception described in part II of this thesis was the desire to ground the language acquisition task advanced in part I in a realistic lexical semantic representation which could be shown to be recoverable from visual input. In order to attempt the integration of the two halves of this thesis it is first necessary to successfully accomplish the first two tasks outlined above. Additionally, one must formulate a suitable linking rule for the semantic representation produced by the aggregation process described above. This linking rule must then be inverted in a fashion similar to the way the Jackendovian liking rule was inverted in section 3.1. This inverted linking rule could then be combined with a hybrid language acquisition model based on the syntactic theory of KENUNIA but utilizing a more elaborate semantic representation with a fracturing rule along the lines of MAIMRA and DAVRA. The substantial effort of building such a comprehensive computational model of language acquisition remains for future work. Nonetheless, this thesis has taken a modest first in this direction by elaborating a framework for approaching this task and demonstrating detailed working implementations of a number of crucial components that will ultimately be needed to construct such language acquisition models.

Appendix A

Maimra in Operation

This appendix contains a trace of MAIMRA processing the corpus from figure 1.2 using the grammar from figure 4.1. The final lexicon produced for this run is illustrated in figure 4.3. This trace depicts MAIMRA processing the corpus, utterance by utterance, producing first a disjunctive parse tree for each utterance and then a disjunctive lexicon formula for that utterance.

```
lcs: (OR (BE PERSON1 (AT PERSON3))
         (GO PERSON1 (PATH (FROM PERSON3) (TO PERSON2)))
         (GO PERSON1 (FROM PERSON3))
         (GO PERSON1 (TO PERSON2))
         (GO PERSON1 (PATH))
         (BE PERSON1 (AT PERSON2)))
sentence: (JOHN ROLLED)
parse: (S (NP (N JOHN)) (VP (V ROLLED)))
fracture:
(OR (AND (DEFINITION JOHN N PERSON3)
         (DEFINITION ROLLED V (BE PERSON1 (AT ?0))))
    (AND (DEFINITION JOHN N (AT PERSON3))
         (DEFINITION ROLLED V (BE PERSON1 ?0)))
    (AND (DEFINITION JOHN N PERSON1)
         (DEFINITION ROLLED V (BE ?0 (AT PERSON3))))
    (AND (DEFINITION JOHN N PERSON2)
         (DEFINITION ROLLED V (GO PERSON1 (PATH (FROM PERSON3) (TO ?0)))))
    (AND (DEFINITION JOHN N (TO PERSON2))
         (DEFINITION ROLLED V (GO PERSON1 (PATH ?0 (FROM PERSON3)))))
    (AND (DEFINITION JOHN N PERSON3)
         (DEFINITION ROLLED V (GO PERSON1 (PATH (FROM ?0) (TO PERSON2)))))
    (AND (DEFINITION JOHN N (FROM PERSON3))
         (DEFINITION ROLLED V (GO PERSON1 (PATH ?0 (TO PERSON2)))))
    (AND (DEFINITION JOHN N (PATH (FROM PERSON3) (TO PERSON2)))
         (DEFINITION ROLLED V (GO PERSON1 ?0)))
    (AND (DEFINITION JOHN N PERSON1)
         (DEFINITION ROLLED V (GO ?0 (PATH (FROM PERSON3) (TO PERSON2)))))
    (AND (DEFINITION JOHN N PERSON3)
         (DEFINITION ROLLED V (GO PERSON1 (FROM ?0))))
    (AND (DEFINITION JOHN N (FROM PERSON3))
```

```
     (DEFINITION ROLLED V (GO PERSON1 ?0)))
(AND (DEFINITION JOHN N PERSON1)
     (DEFINITION ROLLED V (GO ?0 (FROM PERSON3))))
(AND (DEFINITION JOHN N PERSON2)
     (DEFINITION ROLLED V (GO PERSON1 (TO ?0))))
(AND (DEFINITION JOHN N (TO PERSON2))
     (DEFINITION ROLLED V (GO PERSON1 ?0)))
(AND (DEFINITION JOHN N PERSON1)
     (DEFINITION ROLLED V (GO ?0 (TO PERSON2))))
(AND (DEFINITION JOHN N (PATH))
     (DEFINITION ROLLED V (GO PERSON1 ?0)))
(AND (DEFINITION JOHN N PERSON1)
     (DEFINITION ROLLED V (GO ?0 (PATH))))
(AND (DEFINITION JOHN N PERSON2)
     (DEFINITION ROLLED V (BE PERSON1 (AT ?0))))
(AND (DEFINITION JOHN N (AT PERSON2))
     (DEFINITION ROLLED V (BE PERSON1 ?0)))
(AND (DEFINITION JOHN N PERSON1)
     (DEFINITION ROLLED V (BE ?0 (AT PERSON2)))))
```

```
lcs: (OR (BE PERSON2 (AT PERSON3))
         (GO PERSON2 (PATH (FROM PERSON3) (TO PERSON1)))
         (GO PERSON2 (FROM PERSON3))
         (GO PERSON2 (TO PERSON1))
         (GO PERSON2 (PATH))
         (BE PERSON2 (AT PERSON1)))
sentence: (MARY ROLLED)
parse: (S (NP (N MARY)) (VP (V ROLLED)))
fracture: (OR (AND (DEFINITION MARY N PERSON2)
                   (DEFINITION ROLLED V (BE ?0 (AT PERSON3))))
              (AND (DEFINITION MARY N PERSON2)
                   (DEFINITION ROLLED V (GO ?0 (FROM PERSON3))))
              (AND (DEFINITION MARY N PERSON2)
                   (DEFINITION ROLLED V (GO ?0 (PATH)))))
```

```
lcs: (OR (BE PERSON3 (AT PERSON1))
         (GO PERSON3 (PATH (FROM PERSON1) (TO PERSON2)))
         (GO PERSON3 (FROM PERSON1))
         (GO PERSON3 (TO PERSON2))
         (GO PERSON3 (PATH))
         (BE PERSON3 (AT PERSON2)))
sentence: (BILL ROLLED)
parse: (S (NP (N BILL)) (VP (V ROLLED)))
fracture: (AND (DEFINITION BILL N PERSON3)
               (DEFINITION ROLLED V (GO ?O (PATH))))
```

```
lcs: (OR (BE OBJECT1 (AT PERSON1))
         (GO OBJECT1 (PATH (FROM PERSON1) (TO PERSON2)))
         (GO OBJECT1 (FROM PERSON1))
         (GO OBJECT1 (TO PERSON2))
         (GO OBJECT1 (PATH))
         (BE OBJECT1 (AT PERSON2)))
sentence: (THE CUP ROLLED)
parse: (OR (S (OR (NP (DET THE) (N CUP))
                  (NP (N THE) (NP (N CUP)))
                  (NP (N THE) (VP (V CUP)))
                  (NP (N THE) (PP (P CUP))))
              (VP (V ROLLED)))
           (S (NP (N THE))
              (OR (VP (OR (AUX (DO CUP))
                          (AUX (BE CUP))
                          (AUX (MODAL CUP))
                          (AUX (TO CUP))
                          (AUX (HAVE CUP)))
                      (V ROLLED))
                  (VP (V CUP) (VP (V ROLLED))))))
fracture: (OR (AND (DEFINITION THE N OBJECT1)
                   (OR (DEFINITION CUP HAVE SEMANTICLESS)
                       (DEFINITION CUP TO SEMANTICLESS)
                       (DEFINITION CUP MODAL SEMANTICLESS)
                       (DEFINITION CUP BE SEMANTICLESS)
                       (DEFINITION CUP DO SEMANTICLESS))
                   (DEFINITION ROLLED V (GO ?O (PATH))))
              (AND (DEFINITION THE DET SEMANTICLESS)
                   (DEFINITION CUP N OBJECT1)
                   (DEFINITION ROLLED V (GO ?O (PATH)))))
```

```
lcs: (OR (BE PERSON3 (AT PERSON1))
         (GO PERSON3 (PATH (FROM PERSON1) (TO PERSON2)))
         (GO PERSON3 (FROM PERSON1))
         (GO PERSON3 (TO PERSON2))
         (GO PERSON3 (PATH))
         (BE PERSON3 (AT PERSON2)))
sentence: (BILL RAN TO MARY)
parse: (OR (S (OR (NP (N BILL) (NP (N RAN)))
                  (NP (N BILL) (VP (V RAN)))
                  (NP (N BILL) (PP (P RAN))))
              (VP (V TO) (NP (N MARY))))
           (S (NP (N BILL))
              (OR (VP (V RAN) (PP (P TO)) (NP (N MARY)))
                  (VP (V RAN) (VP (V TO)) (NP (N MARY)))
                  (VP (V RAN) (NP (N TO)) (NP (N MARY)))
                  (VP (OR (AUX (DO RAN))
                          (AUX (BE RAN))
                          (AUX (MODAL RAN))
                          (AUX (TO RAN))
                          (AUX (HAVE RAN)))
                      (V TO)
                      (NP (N MARY)))
                  (VP (V RAN)
                      (OR (NP (DET TO) (N MARY))
                          (NP (N TO) (NP (N MARY)))))
                  (VP (V RAN) (VP (V TO) (NP (N MARY))))
                  (VP (V RAN) (PP (P TO) (NP (N MARY)))))))
```

```
fracture:
(OR (AND (DEFINITION BILL N PERSON3)
         (OR (AND (DEFINITION MARY N PERSON2)
                  (DEFINITION TO P (TO ?0))
                  (DEFINITION RAN V (GO ?0 (PATH ?1 (FROM PERSON1)))))
             (AND (DEFINITION MARY N PERSON2)
                  (DEFINITION TO P (PATH (FROM PERSON1) (TO ?0)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (DEFINITION MARY N PERSON2)
                  (DEFINITION TO V (TO ?0))
                  (DEFINITION RAN V (GO ?0 (PATH ?1 (FROM PERSON1)))))
             (AND (DEFINITION MARY N PERSON2)
                  (DEFINITION TO V (PATH (FROM PERSON1) (TO ?0)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (DEFINITION TO DET SEMANTICLESS)
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM PERSON1) (TO ?1)))))
             (AND (DEFINITION MARY N PERSON2)
                  (DEFINITION TO N (TO ?0))
                  (DEFINITION RAN V (GO ?0 (PATH ?1 (FROM PERSON1)))))
             (AND (DEFINITION MARY N PERSON2)
                  (DEFINITION TO N (PATH (FROM PERSON1) (TO ?0)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                      (DEFINITION RAN TO SEMANTICLESS)
                      (DEFINITION RAN MODAL SEMANTICLESS)
                      (DEFINITION RAN BE SEMANTICLESS)
                      (DEFINITION RAN DO SEMANTICLESS))
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION TO V (GO ?0 (PATH (FROM PERSON1) (TO ?1)))))
             (AND (DEFINITION TO N PERSON1)
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION TO N (FROM PERSON1))
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH ?1 (TO ?2)))))
             (AND (DEFINITION TO V PERSON1)
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION TO V (FROM PERSON1))
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH ?1 (TO ?2)))))
             (AND (DEFINITION TO P PERSON1)
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION TO P (FROM PERSON1))
                  (DEFINITION MARY N PERSON2)
                  (DEFINITION RAN V (GO ?0 (PATH ?1 (TO ?2)))))))
```

```
(AND (DEFINITION BILL N PERSON3)
     (OR (AND (DEFINITION MARY N PERSON2)
              (DEFINITION TO P (TO ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (DEFINITION MARY N PERSON2)
              (DEFINITION TO V (TO ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (DEFINITION TO DET SEMANTICLESS)
              (DEFINITION MARY N PERSON2)
              (DEFINITION RAN V (GO ?0 (TO ?1))))
         (AND (DEFINITION MARY N PERSON2)
              (DEFINITION TO N (TO ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                  (DEFINITION RAN TO SEMANTICLESS)
                  (DEFINITION RAN MODAL SEMANTICLESS)
                  (DEFINITION RAN BE SEMANTICLESS)
                  (DEFINITION RAN DO SEMANTICLESS))
              (DEFINITION MARY N PERSON2)
              (DEFINITION TO V (GO ?0 (TO ?1))))))
(AND (DEFINITION BILL N PERSON3)
     (OR (AND (DEFINITION MARY N PERSON2)
              (DEFINITION TO P (AT ?0))
              (DEFINITION RAN V (BE ?0 ?1)))
         (AND (DEFINITION MARY N PERSON2)
              (DEFINITION TO V (AT ?0))
              (DEFINITION RAN V (BE ?0 ?1)))
         (AND (DEFINITION TO DET SEMANTICLESS)
              (DEFINITION MARY N PERSON2)
              (DEFINITION RAN V (BE ?0 (AT ?1))))
         (AND (DEFINITION MARY N PERSON2)
              (DEFINITION TO N (AT ?0))
              (DEFINITION RAN V (BE ?0 ?1)))
         (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                  (DEFINITION RAN TO SEMANTICLESS)
                  (DEFINITION RAN MODAL SEMANTICLESS)
                  (DEFINITION RAN BE SEMANTICLESS)
                  (DEFINITION RAN DO SEMANTICLESS))
              (DEFINITION MARY N PERSON2)
              (DEFINITION TO V (BE ?0 (AT ?1)))))))
```

```
lcs: (OR (BE PERSON3 (AT PERSON1))
         (GO PERSON3 (PATH (FROM PERSON1) (TO PERSON2)))
         (GO PERSON3 (FROM PERSON1))
         (GO PERSON3 (TO PERSON2))
         (GO PERSON3 (PATH))
         (BE PERSON3 (AT PERSON2)))
sentence: (BILL RAN FROM JOHN)
parse: (OR (S (NP (N BILL) (VP (V RAN))) (VP (V FROM) (NP (N JOHN))))
           (S (NP (N BILL))
              (OR (VP (V RAN) (PP (P FROM)) (NP (N JOHN)))
                  (VP (V RAN) (VP (V FROM)) (NP (N JOHN)))
                  (VP (V RAN) (NP (N FROM)) (NP (N JOHN)))
                  (VP (OR (AUX (DO RAN))
                          (AUX (BE RAN))
                          (AUX (MODAL RAN))
                          (AUX (TO RAN))
                          (AUX (HAVE RAN)))
                      (V FROM)
                      (NP (N JOHN)))
                  (VP (V RAN)
                      (OR (NP (DET FROM) (N JOHN))
                          (NP (N FROM) (NP (N JOHN)))))
                  (VP (V RAN) (VP (V FROM) (NP (N JOHN))))
                  (VP (V RAN) (PP (P FROM) (NP (N JOHN)))))))
```

```
fracture:
(OR (AND (DEFINITION BILL N PERSON3)
         (OR (AND (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM P (AT ?0))
                  (DEFINITION RAN V (BE ?0 ?1)))
             (AND (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM V (AT ?0))
                  (DEFINITION RAN V (BE ?0 ?1)))
             (AND (DEFINITION FROM DET SEMANTICLESS)
                  (DEFINITION JOHN N PERSON1)
                  (DEFINITION RAN V (BE ?0 (AT ?1))))
             (AND (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM N (AT ?0))
                  (DEFINITION RAN V (BE ?0 ?1)))
             (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                      (DEFINITION RAN TO SEMANTICLESS)
                      (DEFINITION RAN MODAL SEMANTICLESS)
                      (DEFINITION RAN BE SEMANTICLESS)
                      (DEFINITION RAN DO SEMANTICLESS))
                  (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM V (BE ?0 (AT ?1))))))
    (AND (DEFINITION BILL N PERSON3)
         (OR (AND (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM P (PATH (FROM ?0) (TO PERSON2)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM V (PATH (FROM ?0) (TO PERSON2)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM N (PATH (FROM ?0) (TO PERSON2)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                      (DEFINITION RAN TO SEMANTICLESS)
                      (DEFINITION RAN MODAL SEMANTICLESS)
                      (DEFINITION RAN BE SEMANTICLESS)
                      (DEFINITION RAN DO SEMANTICLESS))
                  (DEFINITION JOHN N PERSON1)
                  (DEFINITION FROM V (GO ?0 (PATH (FROM ?1) (TO PERSON2)))))
             (AND (DEFINITION FROM N PERSON2)
                  (DEFINITION JOHN N PERSON1)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION FROM V PERSON2)
                  (DEFINITION JOHN N PERSON1)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION FROM P PERSON2)
                  (DEFINITION JOHN N PERSON1)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))))
```

```
(AND (DEFINITION BILL N PERSON3)
     (OR (AND (DEFINITION JOHN N PERSON1)
              (DEFINITION FROM P (FROM ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (DEFINITION JOHN N PERSON1)
              (DEFINITION FROM V (FROM ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (DEFINITION JOHN N PERSON1)
              (DEFINITION FROM N (FROM ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                  (DEFINITION RAN TO SEMANTICLESS)
                  (DEFINITION RAN MODAL SEMANTICLESS)
                  (DEFINITION RAN BE SEMANTICLESS)
                  (DEFINITION RAN DO SEMANTICLESS))
              (DEFINITION JOHN N PERSON1)
              (DEFINITION FROM V (GO ?0 (FROM ?1)))))))
```

```
lcs: (OR (BE PERSON3 (AT PERSON1))
         (GO PERSON3 (PATH (FROM PERSON1) (TO OBJECT1)))
         (GO PERSON3 (FROM PERSON1))
         (GO PERSON3 (TO OBJECT1))
         (GO PERSON3 (PATH))
         (BE PERSON3 (AT OBJECT1)))
sentence: (BILL RAN TO THE CUP)
```

```
parse: (OR (S (NP (N BILL) (VP (V RAN)))
              (OR (VP (V TO) (NP (N THE)) (NP (N CUP)))
                  (VP (V TO)
                      (OR (NP (DET THE) (N CUP))
                          (NP (N THE) (NP (N CUP)))))))
           (S (NP (N BILL))
              (OR (VP (V RAN) (PP (P TO) (NP (N THE))) (NP (N CUP)))
                  (VP (V RAN) (VP (V TO) (NP (N THE))) (NP (N CUP)))
                  (VP (V RAN)
                      (OR (NP (DET TO) (N THE))
                          (NP (N TO) (NP (N THE))))
                      (NP (N CUP)))
                  (VP (OR (AUX (DO RAN))
                          (AUX (BE RAN))
                          (AUX (MODAL RAN))
                          (AUX (TO RAN))
                          (AUX (HAVE RAN)))
                      (V TO)
                      (NP (N THE))
                      (NP (N CUP)))
                  (VP (V RAN) (NP (N TO)) (NP (N THE)) (NP (N CUP)))
                  (VP (V RAN) (VP (V TO)) (NP (N THE)) (NP (N CUP)))
                  (VP (V RAN) (PP (P TO)) (NP (N THE)) (NP (N CUP)))
                  (VP (V RAN) (PP (P TO))
                      (OR (NP (DET THE) (N CUP))
                          (NP (N THE) (NP (N CUP)))))
                  (VP (V RAN) (VP (V TO))
                      (OR (NP (DET THE) (N CUP))
                          (NP (N THE) (NP (N CUP)))))
                  (VP (V RAN) (NP (N TO))
                      (OR (NP (DET THE) (N CUP))
                          (NP (N THE) (NP (N CUP)))))
                  (VP (OR (AUX (DO RAN))
                          (AUX (BE RAN))
                          (AUX (MODAL RAN))
                          (AUX (TO RAN))
                          (AUX (HAVE RAN)))
                      (V TO)
                      (OR (NP (DET THE) (N CUP))
                          (NP (N THE) (NP (N CUP)))))
```

```
(VP (V RAN)
    (OR (NP (N TO) (NP (N THE)) (NP (N CUP)))
        (NP (DET TO) (N THE) (NP (N CUP)))
        (NP (N TO)
            (OR (NP (DET THE) (N CUP))
                (NP (N THE) (NP (N CUP)))))))
(VP (V RAN)
    (OR (VP (V TO) (NP (N THE)) (NP (N CUP)))
        (VP (V TO)
            (OR (NP (DET THE) (N CUP))
                (NP (N THE) (NP (N CUP)))))))
(VP (V RAN)
    (OR (PP (P TO) (NP (N THE)) (NP (N CUP)))
        (PP (P TO)
            (OR (NP (DET THE) (N CUP))
                (NP (N THE) (NP (N CUP)))))))))))
```

```
fracture:
(OR (AND (DEFINITION BILL N PERSON3)
         (OR (AND (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION TO P (PATH (FROM PERSON1) (TO ?0)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION TO V (PATH (FROM PERSON1) (TO ?0)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION TO N (PATH (FROM PERSON1) (TO ?0)))
                  (DEFINITION RAN V (GO ?0 ?1)))
             (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                      (DEFINITION RAN TO SEMANTICLESS)
                      (DEFINITION RAN MODAL SEMANTICLESS)
                      (DEFINITION RAN BE SEMANTICLESS)
                      (DEFINITION RAN DO SEMANTICLESS))
                  (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION TO V (GO ?0 (PATH (FROM PERSON1) (TO ?1)))))
             (AND (DEFINITION TO N PERSON1)
                  (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION TO V PERSON1)
                  (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))
             (AND (DEFINITION TO P PERSON1)
                  (DEFINITION THE DET SEMANTICLESS)
                  (DEFINITION CUP N OBJECT1)
                  (DEFINITION RAN V (GO ?0 (PATH (FROM ?1) (TO ?2)))))))
```

```
(AND (DEFINITION BILL N PERSON3)
     (OR (AND (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO P (TO ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO V (TO ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO N (TO ?0))
              (DEFINITION RAN V (GO ?0 ?1)))
         (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                  (DEFINITION RAN TO SEMANTICLESS)
                  (DEFINITION RAN MODAL SEMANTICLESS)
                  (DEFINITION RAN BE SEMANTICLESS)
                  (DEFINITION RAN DO SEMANTICLESS))
              (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO V (GO ?0 (TO ?1))))))
(AND (DEFINITION BILL N PERSON3)
     (OR (AND (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO P (AT ?0))
              (DEFINITION RAN V (BE ?0 ?1)))
         (AND (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO V (AT ?0))
              (DEFINITION RAN V (BE ?0 ?1)))
         (AND (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO N (AT ?0))
              (DEFINITION RAN V (BE ?0 ?1)))
         (AND (OR (DEFINITION RAN HAVE SEMANTICLESS)
                  (DEFINITION RAN TO SEMANTICLESS)
                  (DEFINITION RAN MODAL SEMANTICLESS)
                  (DEFINITION RAN BE SEMANTICLESS)
                  (DEFINITION RAN DO SEMANTICLESS))
              (DEFINITION THE DET SEMANTICLESS)
              (DEFINITION CUP N OBJECT1)
              (DEFINITION TO V (BE ?0 (AT ?1)))))))
```

```
lcs: (OR (BE OBJECT1 (AT PERSON1))
         (GO OBJECT1 (PATH (FROM PERSON1) (TO PERSON2)))
         (GO OBJECT1 (FROM PERSON1))
         (GO OBJECT1 (TO PERSON2))
         (GO OBJECT1 (PATH))
         (BE OBJECT1 (AT PERSON2)))
sentence: (THE CUP SLID FROM JOHN TO MARY)
```

```
parse:
(OR (S (OR (NP (DET THE)
               (N CUP)
               (SBAR (S (NP (N SLID)) (VP (V FROM) (NP (N JOHN))))))
           (NP (DET THE)
               (N CUP)
               (OR (PP (P SLID) (NP (N FROM)))
                   (PP (P SLID) (VP (V FROM)))
                   (PP (P SLID) (PP (P FROM))))
               (NP (N JOHN)))
           (NP (DET THE) (N CUP) (NP (N SLID)) (PP (P FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP) (VP (V SLID)) (PP (P FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP) (PP (P SLID)) (PP (P FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP)
               (OR (VP (OR (AUX (DO SLID))
                           (AUX (BE SLID))
                           (AUX (MODAL SLID))
                           (AUX (TO SLID))
                           (AUX (HAVE SLID)))
                       (V FROM))
                   (VP (V SLID) (NP (N FROM)))
                   (VP (V SLID) (VP (V FROM)))
                   (VP (V SLID) (PP (P FROM))))
               (NP (N JOHN)))
           (NP (DET THE) (N CUP) (NP (N SLID)) (VP (V FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP) (VP (V SLID)) (VP (V FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP) (PP (P SLID)) (VP (V FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP)
               (OR (NP (DET SLID) (N FROM))
                   (NP (N SLID) (NP (N FROM)))
                   (NP (N SLID) (VP (V FROM)))
                   (NP (N SLID) (PP (P FROM))))
               (NP (N JOHN)))
           (NP (DET THE) (N CUP) (NP (N SLID)) (NP (N FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP) (VP (V SLID)) (NP (N FROM)) (NP (N JOHN)))
           (NP (DET THE) (N CUP) (PP (P SLID)) (NP (N FROM)) (NP (N JOHN)))
           (NP (DET THE)
               (N CUP)
               (SBAR (S (NP (N SLID)) (VP (V FROM))))
               (NP (N JOHN)))
           (NP (DET THE) (N CUP) (PP (P SLID)) (NP (N FROM) (NP (N JOHN))))
           (NP (DET THE) (N CUP) (VP (V SLID)) (NP (N FROM) (NP (N JOHN))))
           (NP (DET THE) (N CUP) (NP (N SLID)) (NP (N FROM) (NP (N JOHN))))
```

```
    (NP (DET THE)
        (N CUP)
        (OR (NP (N SLID) (PP (P FROM)) (NP (N JOHN)))
            (NP (N SLID) (VP (V FROM)) (NP (N JOHN)))
            (NP (N SLID) (NP (N FROM)) (NP (N JOHN)))
            (NP (DET SLID) (N FROM) (NP (N JOHN)))
            (NP (N SLID) (NP (N FROM) (NP (N JOHN))))
            (NP (N SLID) (VP (V FROM) (NP (N JOHN))))
            (NP (N SLID) (PP (P FROM) (NP (N JOHN))))))
    (NP (DET THE) (N CUP) (PP (P SLID)) (VP (V FROM) (NP (N JOHN))))
    (NP (DET THE) (N CUP) (VP (V SLID)) (VP (V FROM) (NP (N JOHN))))
    (NP (DET THE) (N CUP) (NP (N SLID)) (VP (V FROM) (NP (N JOHN))))
    (NP (DET THE)
        (N CUP)
        (OR (VP (V SLID) (PP (P FROM)) (NP (N JOHN)))
            (VP (V SLID) (VP (V FROM)) (NP (N JOHN)))
            (VP (V SLID) (NP (N FROM)) (NP (N JOHN)))
            (VP (OR (AUX (DO SLID))
                    (AUX (BE SLID))
                    (AUX (MODAL SLID))
                    (AUX (TO SLID))
                    (AUX (HAVE SLID)))
                (V FROM)
                (NP (N JOHN)))
            (VP (V SLID) (NP (N FROM) (NP (N JOHN))))
            (VP (V SLID) (VP (V FROM) (NP (N JOHN))))
            (VP (V SLID) (PP (P FROM) (NP (N JOHN))))))
    (NP (DET THE) (N CUP) (PP (P SLID)) (PP (P FROM) (NP (N JOHN))))
    (NP (DET THE) (N CUP) (VP (V SLID)) (PP (P FROM) (NP (N JOHN))))
    (NP (DET THE) (N CUP) (NP (N SLID)) (PP (P FROM) (NP (N JOHN))))
    (NP (DET THE)
        (N CUP)
        (OR (PP (P SLID) (PP (P FROM)) (NP (N JOHN)))
            (PP (P SLID) (VP (V FROM)) (NP (N JOHN)))
            (PP (P SLID) (NP (N FROM)) (NP (N JOHN)))
            (PP (P SLID) (NP (N FROM) (NP (N JOHN))))
            (PP (P SLID) (VP (V FROM) (NP (N JOHN))))
            (PP (P SLID) (PP (P FROM) (NP (N JOHN)))))))
(VP (V TO) (NP (N MARY))))
```

```
(S (OR (NP (DET THE) (N CUP) (NP (N SLID)))
       (NP (DET THE) (N CUP) (VP (V SLID)))
       (NP (DET THE) (N CUP) (PP (P SLID))))
   (OR (VP (V FROM) (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
       (VP (V FROM) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
       (VP (V FROM) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
       (VP (V FROM)
           (OR (NP (N JOHN) (NP (N TO)))
               (NP (N JOHN) (VP (V TO)))
               (NP (N JOHN) (PP (P TO))))
           (NP (N MARY)))
       (VP (V FROM) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
       (VP (V FROM) (SBAR (S (NP (N JOHN)) (VP (V TO)))) (NP (N MARY)))
       (VP (V FROM) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
       (VP (V FROM)
           (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
               (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
               (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
               (NP (N JOHN) (NP (N TO) (NP (N MARY))))
               (NP (N JOHN) (VP (V TO) (NP (N MARY))))
               (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
       (VP (V FROM) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
       (VP (V FROM) (NP (N JOHN)) (PP (P TO) (NP (N MARY))))))
(S (NP (DET THE) (N CUP))
   (OR (VP (V SLID)
           (PP (P FROM))
           (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
       (VP (V SLID)
           (VP (V FROM))
           (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
       (VP (V SLID)
           (NP (N FROM))
           (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
       (VP (OR (AUX (DO SLID))
               (AUX (BE SLID))
               (AUX (MODAL SLID))
               (AUX (TO SLID))
               (AUX (HAVE SLID)))
           (V FROM)
           (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
       (VP (V SLID)
           (SBAR (S (NP (N FROM) (NP (N JOHN)))
                    (VP (V TO) (NP (N MARY))))))
```

```
(VP (V SLID)
    (OR (PP (P FROM) (SBAR (S (NP (N JOHN)) (VP (V TO)))))
        (PP (P FROM) (NP (N JOHN)) (NP (N TO)))
        (PP (P FROM)
            (OR (NP (N JOHN) (NP (N TO)))
                (NP (N JOHN) (VP (V TO)))
                (NP (N JOHN) (PP (P TO)))))
        (PP (P FROM) (NP (N JOHN)) (VP (V TO)))
        (PP (P FROM) (NP (N JOHN)) (PP (P TO))))
    (NP (N MARY)))
(VP (V SLID) (PP (P FROM)) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
(VP (V SLID) (VP (V FROM)) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
(VP (V SLID) (NP (N FROM)) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (NP (N JOHN))
    (PP (P TO))
    (NP (N MARY)))
(VP (V SLID) (NP (N FROM) (NP (N JOHN))) (PP (P TO)) (NP (N MARY)))
(VP (V SLID) (VP (V FROM) (NP (N JOHN))) (PP (P TO)) (NP (N MARY)))
(VP (V SLID) (PP (P FROM) (NP (N JOHN))) (PP (P TO)) (NP (N MARY)))
(VP (V SLID)
    (OR (VP (V FROM) (SBAR (S (NP (N JOHN)) (VP (V TO)))))
        (VP (V FROM) (NP (N JOHN)) (NP (N TO)))
        (VP (V FROM)
            (OR (NP (N JOHN) (NP (N TO)))
                (NP (N JOHN) (VP (V TO)))
                (NP (N JOHN) (PP (P TO)))))
        (VP (V FROM) (NP (N JOHN)) (VP (V TO)))
        (VP (V FROM) (NP (N JOHN)) (PP (P TO))))
    (NP (N MARY)))
(VP (V SLID) (PP (P FROM)) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
(VP (V SLID) (VP (V FROM)) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
(VP (V SLID) (NP (N FROM)) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
```

```
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (NP (N JOHN))
    (VP (V TO))
    (NP (N MARY)))
(VP (V SLID) (NP (N FROM) (NP (N JOHN))) (VP (V TO)) (NP (N MARY)))
(VP (V SLID) (VP (V FROM) (NP (N JOHN))) (VP (V TO)) (NP (N MARY)))
(VP (V SLID) (PP (P FROM) (NP (N JOHN))) (VP (V TO)) (NP (N MARY)))
(VP (V SLID)
    (OR (NP (N FROM) (SBAR (S (NP (N JOHN)) (VP (V TO)))))
        (NP (N FROM) (NP (N JOHN)) (NP (N TO)))
        (NP (N FROM)
            (OR (NP (N JOHN) (NP (N TO)))
                (NP (N JOHN) (VP (V TO)))
                (NP (N JOHN) (PP (P TO)))))
        (NP (N FROM) (NP (N JOHN)) (VP (V TO)))
        (NP (N FROM) (NP (N JOHN)) (PP (P TO))))
    (NP (N MARY)))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (OR (NP (N JOHN) (NP (N TO)))
        (NP (N JOHN) (VP (V TO)))
        (NP (N JOHN) (PP (P TO))))
    (NP (N MARY)))
(VP (V SLID)
    (NP (N FROM))
    (OR (NP (N JOHN) (NP (N TO)))
        (NP (N JOHN) (VP (V TO)))
        (NP (N JOHN) (PP (P TO))))
    (NP (N MARY)))
(VP (V SLID)
    (VP (V FROM))
    (OR (NP (N JOHN) (NP (N TO)))
        (NP (N JOHN) (VP (V TO)))
        (NP (N JOHN) (PP (P TO))))
    (NP (N MARY)))
```

```
(VP (V SLID)
    (PP (P FROM))
    (OR (NP (N JOHN) (NP (N TO)))
        (NP (N JOHN) (VP (V TO)))
        (NP (N JOHN) (PP (P TO))))
    (NP (N MARY)))
(VP (V SLID) (PP (P FROM)) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
(VP (V SLID) (VP (V FROM)) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
(VP (V SLID) (NP (N FROM)) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (NP (N JOHN))
    (NP (N TO))
    (NP (N MARY)))
(VP (V SLID) (NP (N FROM) (NP (N JOHN))) (NP (N TO)) (NP (N MARY)))
(VP (V SLID) (VP (V FROM) (NP (N JOHN))) (NP (N TO)) (NP (N MARY)))
(VP (V SLID) (PP (P FROM) (NP (N JOHN))) (NP (N TO)) (NP (N MARY)))
(VP (V SLID)
    (SBAR (S (NP (N FROM) (NP (N JOHN))) (VP (V TO))))
    (NP (N MARY)))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (SBAR (S (NP (N JOHN)) (VP (V TO))))
    (NP (N MARY)))
(VP (V SLID)
    (NP (N FROM))
    (SBAR (S (NP (N JOHN)) (VP (V TO))))
    (NP (N MARY)))
(VP (V SLID)
    (VP (V FROM))
    (SBAR (S (NP (N JOHN)) (VP (V TO))))
    (NP (N MARY)))
(VP (V SLID)
    (PP (P FROM))
    (SBAR (S (NP (N JOHN)) (VP (V TO))))
    (NP (N MARY)))
```

```
(VP (V SLID) (PP (P FROM) (NP (N JOHN))) (NP (N TO) (NP (N MARY))))
(VP (V SLID) (VP (V FROM) (NP (N JOHN))) (NP (N TO) (NP (N MARY))))
(VP (V SLID) (NP (N FROM) (NP (N JOHN))) (NP (N TO) (NP (N MARY))))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
(VP (V SLID) (NP (N FROM)) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
(VP (V SLID) (VP (V FROM)) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
(VP (V SLID) (PP (P FROM)) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
(VP (V SLID)
    (PP (P FROM))
    (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
        (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO) (NP (N MARY))))
        (NP (N JOHN) (VP (V TO) (NP (N MARY))))
        (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
(VP (V SLID)
    (VP (V FROM))
    (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
        (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO) (NP (N MARY))))
        (NP (N JOHN) (VP (V TO) (NP (N MARY))))
        (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
(VP (V SLID)
    (NP (N FROM))
    (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
        (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO) (NP (N MARY))))
        (NP (N JOHN) (VP (V TO) (NP (N MARY))))
        (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
        (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
        (NP (N JOHN) (NP (N TO) (NP (N MARY))))
        (NP (N JOHN) (VP (V TO) (NP (N MARY))))
        (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
```

```
(VP (V SLID)
    (OR (NP (N FROM)
            (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
        (NP (N FROM) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
        (NP (N FROM) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
        (NP (N FROM)
            (OR (NP (N JOHN) (NP (N TO)))
                (NP (N JOHN) (VP (V TO)))
                (NP (N JOHN) (PP (P TO))))
            (NP (N MARY)))
        (NP (N FROM) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
        (NP (N FROM)
            (SBAR (S (NP (N JOHN)) (VP (V TO))))
            (NP (N MARY)))
        (NP (N FROM) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
        (NP (N FROM)
            (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
                (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
                (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
                (NP (N JOHN) (NP (N TO) (NP (N MARY))))
                (NP (N JOHN) (VP (V TO) (NP (N MARY))))
                (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
        (NP (N FROM) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
        (NP (N FROM) (NP (N JOHN)) (PP (P TO) (NP (N MARY))))))
(VP (V SLID) (PP (P FROM) (NP (N JOHN))) (VP (V TO) (NP (N MARY))))
(VP (V SLID) (VP (V FROM) (NP (N JOHN))) (VP (V TO) (NP (N MARY))))
(VP (V SLID) (NP (N FROM) (NP (N JOHN))) (VP (V TO) (NP (N MARY))))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (NP (N JOHN))
    (VP (V TO) (NP (N MARY))))
(VP (V SLID) (NP (N FROM)) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
(VP (V SLID) (VP (V FROM)) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
(VP (V SLID) (PP (P FROM)) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
```

```
(VP (V SLID)
    (OR (VP (V FROM)
            (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
        (VP (V FROM) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
        (VP (V FROM) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
        (VP (V FROM)
            (OR (NP (N JOHN) (NP (N TO)))
                (NP (N JOHN) (VP (V TO)))
                (NP (N JOHN) (PP (P TO))))
            (NP (N MARY)))
        (VP (V FROM) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
        (VP (V FROM)
            (SBAR (S (NP (N JOHN)) (VP (V TO))))
            (NP (N MARY)))
        (VP (V FROM) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
        (VP (V FROM)
            (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
                (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
                (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
                (NP (N JOHN) (NP (N TO) (NP (N MARY))))
                (NP (N JOHN) (VP (V TO) (NP (N MARY))))
                (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
        (VP (V FROM) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
        (VP (V FROM) (NP (N JOHN)) (PP (P TO) (NP (N MARY))))))
(VP (V SLID) (PP (P FROM) (NP (N JOHN))) (PP (P TO) (NP (N MARY))))
(VP (V SLID) (VP (V FROM) (NP (N JOHN))) (PP (P TO) (NP (N MARY))))
(VP (V SLID) (NP (N FROM) (NP (N JOHN))) (PP (P TO) (NP (N MARY))))
(VP (OR (AUX (DO SLID))
        (AUX (BE SLID))
        (AUX (MODAL SLID))
        (AUX (TO SLID))
        (AUX (HAVE SLID)))
    (V FROM)
    (NP (N JOHN))
    (PP (P TO) (NP (N MARY))))
(VP (V SLID) (NP (N FROM)) (NP (N JOHN)) (PP (P TO) (NP (N MARY))))
(VP (V SLID) (VP (V FROM)) (NP (N JOHN)) (PP (P TO) (NP (N MARY))))
(VP (V SLID) (PP (P FROM)) (NP (N JOHN)) (PP (P TO) (NP (N MARY))))
```

```
(VP (V SLID)
    (OR (PP (P FROM)
            (SBAR (S (NP (N JOHN)) (VP (V TO) (NP (N MARY))))))
        (PP (P FROM) (NP (N JOHN)) (PP (P TO)) (NP (N MARY)))
        (PP (P FROM) (NP (N JOHN)) (VP (V TO)) (NP (N MARY)))
        (PP (P FROM)
            (OR (NP (N JOHN) (NP (N TO)))
                (NP (N JOHN) (VP (V TO)))
                (NP (N JOHN) (PP (P TO))))
            (NP (N MARY)))
        (PP (P FROM) (NP (N JOHN)) (NP (N TO)) (NP (N MARY)))
        (PP (P FROM)
            (SBAR (S (NP (N JOHN)) (VP (V TO))))
            (NP (N MARY)))
        (PP (P FROM) (NP (N JOHN)) (NP (N TO) (NP (N MARY))))
        (PP (P FROM)
            (OR (NP (N JOHN) (PP (P TO)) (NP (N MARY)))
                (NP (N JOHN) (VP (V TO)) (NP (N MARY)))
                (NP (N JOHN) (NP (N TO)) (NP (N MARY)))
                (NP (N JOHN) (NP (N TO) (NP (N MARY))))
                (NP (N JOHN) (VP (V TO) (NP (N MARY))))
                (NP (N JOHN) (PP (P TO) (NP (N MARY))))))
        (PP (P FROM) (NP (N JOHN)) (VP (V TO) (NP (N MARY))))
        (PP (P FROM) (NP (N JOHN)) (PP (P TO) (NP (N MARY)))))))))
```

```
fracture: (AND (DEFINITION THE DET SEMANTICLESS)
               (DEFINITION CUP N OBJECT1)
               (OR (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM N (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO P (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM V (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO P (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM P (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO P (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM N (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO V (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM V (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO V (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM P (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO V (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM N (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO N (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM V (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO N (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))
                   (AND (DEFINITION JOHN N PERSON1)
                        (DEFINITION FROM P (FROM ?0))
                        (DEFINITION MARY N PERSON2)
                        (DEFINITION TO N (TO ?0))
                        (DEFINITION SLID V (GO ?0 (PATH ?1 ?2))))))
```

```
lcs: (OR (ORIENT PERSON1 (TO PERSON2))
         (ORIENT PERSON2 (TO PERSON3))
         (ORIENT PERSON3 (TO PERSON1)))
sentence: (JOHN FACED MARY)
parse: (S (NP (N JOHN)) (VP (V FACED) (NP (N MARY))))
fracture: (AND (DEFINITION JOHN N PERSON1)
               (DEFINITION MARY N PERSON2)
               (DEFINITION FACED V (ORIENT ?0 (TO ?1))))
```

```
FACED: [V] (ORIENT ?0 (TO ?1))
SLID: [V] (GO ?0 (PATH ?1 ?2))
FROM: *[P] (FROM ?0)
TO: *[N] (TO ?0)
RAN: [V] (GO ?0 ?1)
THE: [DET] SEMANTICLESS
CUP: [N] OBJECT1
BILL: [N] PERSON3
MARY: [N] PERSON2
JOHN: [N] PERSON1
ROLLED: [V] (GO ?0 (PATH))
```

Appendix B

Kenunia in Operation

This appendix contains a trace of KENUNIA processing the corpus from figure 4.8 using the prior semantic knowledge from figure 4.10. Given this information, KENUNIA can derive the syntactic parameter settings and word-to-category mappings illustrated in figure 4.11. This trace depicts KENUNIA processing the corpus, utterance by utterance, showing the interim language model after each utterance, as well as the hypothesized analysis for each utterance. When no analysis is possible, the propositions to be retracted from the language model are highlighted as *culprits*.

John roll -ed.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Theme} : \mathbf{person}_1\}$

Syntactic Parameters:

[I^0 initial]
[I^1 final]
[C^0 final]

Lexicon:

cup:	[X^n]	$\mathbf{object}_1\{\}$
-ed:	[V^2]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[X^n]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[X^n]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[X^n]	$\perp\{\textsc{Source} : 0\}$
Bill:	[X^n]	$\mathbf{person}_3\{\}$
the:	[X^n]	$\perp\{\}$
Mary:	[X^n]	$\mathbf{person}_2\{\}$
to:	[X^n]	$\perp\{\textsc{Goal} : 0\}$
run:	[X^n]	$\perp\{\textsc{Theme} : 1\}$
roll:	[I^0]	$\perp\{\textsc{Theme} : 1\}$

Mary roll -ed.

$\{\textsc{Agent} : \mathbf{person}_2, \textsc{Theme} : \mathbf{person}_2\}$

Syntactic Parameters:

[I^0 initial]
[I^1 final]
[C^0 final]

Lexicon:

cup:	$[X^n]$	$\mathbf{object}_1\{\}$
-ed:	$[V^2]$	$\perp\{\}$
John:	$[D^2]$	$\mathbf{person}_1\{\}$
slide:	$[X^n]$	$\perp\{\textsc{Theme} : 1\}$
that:	$[X^n]$	$\perp\{\}$
$\emptyset$:	$[C^0]$	$\perp\{\}$
face:	$[X^n]$	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	$[X^n]$	$\perp\{\textsc{Source} : 0\}$
Bill:	$[X^n]$	$\mathbf{person}_3\{\}$
the:	$[X^n]$	$\perp\{\}$
Mary:	$[D^2]$	$\mathbf{person}_2\{\}$
to:	$[X^n]$	$\perp\{\textsc{Goal} : 0\}$
run:	$[X^n]$	$\perp\{\textsc{Theme} : 1\}$
roll:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$

Bill roll -ed.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3\}$

Syntactic Parameters:

[I^0 initial]
[I^1 final]
[C^0 final]

Lexicon:

cup:	$[X^n]$	$\mathbf{object}_1\{\}$
-ed:	$[V^2]$	$\perp\{\}$
John:	$[D^2]$	$\mathbf{person}_1\{\}$
slide:	$[X^n]$	$\perp\{\textsc{Theme} : 1\}$
that:	$[X^n]$	$\perp\{\}$
$\emptyset$:	$[C^0]$	$\perp\{\}$
face:	$[X^n]$	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	$[X^n]$	$\perp\{\textsc{Source} : 0\}$
Bill:	$[D^2]$	$\mathbf{person}_3\{\}$
the:	$[X^n]$	$\perp\{\}$
Mary:	$[D^2]$	$\mathbf{person}_2\{\}$
to:	$[X^n]$	$\perp\{\textsc{Goal} : 0\}$
run:	$[X^n]$	$\perp\{\textsc{Theme} : 1\}$
roll:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$

The cup roll -ed.

$\{\textsc{Theme} : \mathbf{object}_1\}$

Syntactic Parameters:

[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[V^2]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[X^n]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[X^n]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[X^n]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[X^n]	$\perp\{\textsc{Goal} : 0\}$
run:	[X^n]	$\perp\{\textsc{Theme} : 1\}$
roll:	[I^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to Mary.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[V^2]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[X^n]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[X^n]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[X^n]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[I^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed from John.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Source} : \mathbf{person}_1\}$

Syntactic Parameters:

[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[V^2]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[X^n]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[X^n]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[I^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to the cup.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{object}_1\}$

Syntactic Parameters:

[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]

Lexicon:

cup:	$[N^2]$	$\mathbf{object}_1\{\}$
-ed:	$[V^2]$	$\perp\{\}$
John:	$[D^2]$	$\mathbf{person}_1\{\}$
slide:	$[X^n]$	$\perp\{\textsc{Theme} : 1\}$
that:	$[X^n]$	$\perp\{\}$
$\emptyset$:	$[C^0]$	$\perp\{\}$
face:	$[X^n]$	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	$[P^0]$	$\perp\{\textsc{Source} : 0\}$
Bill:	$[D^2]$	$\mathbf{person}_3\{\}$
the:	$[D^0]$	$\perp\{\}$
Mary:	$[D^2]$	$\mathbf{person}_2\{\}$
to:	$[P^0]$	$\perp\{\textsc{Goal} : 0\}$
run:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$
roll:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$

The cup slide -ed from John to Mary.

$\{\textsc{Theme} : \mathbf{object}_1, \textsc{Source} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[V^2]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[X^n]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[I^0]	$\perp\{\textsc{Theme} : 1\}$

John face -ed Mary.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Patient} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Culprits:

category(-*ed*) = V
bar-level(-*ed*) = 2

Syntactic Parameters:

[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-*ed*:	[X^n]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[X^n]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[I^0]	$\perp\{\textsc{Theme} : 1\}$

John face -ed Mary.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Patient} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	$[N^2]$	$\mathbf{object}_1\{\}$
-ed:	$[I^0]$	$\perp\{\}$
John:	$[D^2]$	$\mathbf{person}_1\{\}$
slide:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$
that:	$[X^n]$	$\perp\{\}$
$\emptyset$:	$[C^0]$	$\perp\{\}$
face:	$[V^0]$	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	$[P^0]$	$\perp\{\textsc{Source} : 0\}$
Bill:	$[D^2]$	$\mathbf{person}_3\{\}$
the:	$[D^0]$	$\perp\{\}$
Mary:	$[D^2]$	$\mathbf{person}_2\{\}$
to:	$[P^0]$	$\perp\{\textsc{Goal} : 0\}$
run:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$
roll:	$[I^0]$	$\perp\{\textsc{Theme} : 1\}$

John roll -ed.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Theme} : \mathbf{person}_1\}$

Culprits:

category(*roll*) = I

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[X^0]	$\perp\{\textsc{Theme} : 1\}$

John roll -ed.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Theme} : \mathbf{person}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Mary roll -ed.

$\{\textsc{Agent} : \mathbf{person}_2, \textsc{Theme} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill roll -ed.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

The cup roll -ed.

$\{\textsc{Theme} : \mathbf{object}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to Mary.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{person}_2\}$

Culprits:

category(*run*) = I

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[X^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to Mary.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed from John.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Source} : \mathbf{person}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to the cup.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{object}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[I^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

The cup slide -ed from John to Mary.

$$\{\textsc{Theme} : \mathbf{object}_1, \textsc{Source} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$$

Culprits:

category(*slide*) = I

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[X^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

The cup slide -ed from John to Mary.

$\{\textsc{Theme} : \mathbf{object}_1, \textsc{Source} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

John face -ed Mary.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Patient} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

John roll -ed.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Theme} : \mathbf{person}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Mary roll -ed.

$\{\textsc{Agent} : \mathbf{person}_2, \textsc{Theme} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill roll -ed.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

The cup roll -ed.

$\{\textsc{Theme} : \mathbf{object}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to Mary.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed from John.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Source} : \mathbf{person}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Bill run -ed to the cup.

$\{\textsc{Agent} : \mathbf{person}_3, \textsc{Theme} : \mathbf{person}_3, \textsc{Goal} : \mathbf{object}_1\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

The cup slide -ed from John to Mary.

$\{\textsc{Theme} : \mathbf{object}_1, \textsc{Source} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	$[N^2]$	$\mathbf{object}_1\{\}$
-ed:	$[I^0]$	$\perp\{\}$
John:	$[D^2]$	$\mathbf{person}_1\{\}$
slide:	$[V^0]$	$\perp\{\textsc{Theme} : 1\}$
that:	$[X^n]$	$\perp\{\}$
$\emptyset$:	$[C^0]$	$\perp\{\}$
face:	$[V^0]$	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	$[P^0]$	$\perp\{\textsc{Source} : 0\}$
Bill:	$[D^2]$	$\mathbf{person}_3\{\}$
the:	$[D^0]$	$\perp\{\}$
Mary:	$[D^2]$	$\mathbf{person}_2\{\}$
to:	$[P^0]$	$\perp\{\textsc{Goal} : 0\}$
run:	$[V^0]$	$\perp\{\textsc{Theme} : 1\}$
roll:	$[V^0]$	$\perp\{\textsc{Theme} : 1\}$

John face -ed Mary.

$\{\textsc{Agent} : \mathbf{person}_1, \textsc{Patient} : \mathbf{person}_1, \textsc{Goal} : \mathbf{person}_2\}$

Syntactic Parameters:

[V^0 final]
[V^1 final]
[P^0 initial]
[D^0 initial]
[I^0 initial]
[I^1 final]
[C^0 final]
[adjoin V^2 right]
[adjoin I^0 left]

Lexicon:

cup:	[N^2]	$\mathbf{object}_1\{\}$
-ed:	[I^0]	$\perp\{\}$
John:	[D^2]	$\mathbf{person}_1\{\}$
slide:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
that:	[X^n]	$\perp\{\}$
$\emptyset$:	[C^0]	$\perp\{\}$
face:	[V^0]	$\perp\{\textsc{Patient} : 1, \textsc{Goal} : 0\}$
from:	[P^0]	$\perp\{\textsc{Source} : 0\}$
Bill:	[D^2]	$\mathbf{person}_3\{\}$
the:	[D^0]	$\perp\{\}$
Mary:	[D^2]	$\mathbf{person}_2\{\}$
to:	[P^0]	$\perp\{\textsc{Goal} : 0\}$
run:	[V^0]	$\perp\{\textsc{Theme} : 1\}$
roll:	[V^0]	$\perp\{\textsc{Theme} : 1\}$

Appendix C

Abigail in Operation

This appendix enumerates the perceptual primitives recovered by ABIGAIL after processing the first 172 frames of the movie discussed in section 6.1. Figure 8.13 contains an event graph depicting the temporal structure of these primitives.

```
[0,0](PLACE [JOHN-part] PLACE-0)
[0,0](SUPPORTED [JOHN-part])

[0,1](PLACE [(EYE JOHN)] PLACE-1)

[0,65](PLACE [BALL-part] PLACE-13)
[0,65](CONTACTS [TABLE BOX-part] [BALL-part])
[0,65](SUPPORTS [TABLE BOX-part] [BALL-part])
[0,65](PLACE [(LINE-SEGMENT3 BALL)] PLACE-11)

[0,71](SUPPORTED [BALL-part])
[0,71](SUPPORTED [(LINE-SEGMENT3 BALL)])
[0,71](SUPPORTS [BALL-part] [(LINE-SEGMENT3 BALL)])

[0,171](SUPPORTED [TABLE BOX-part])
[0,171](SUPPORTED [(BOTTOM BOX)])
[0,171](SUPPORTS [TABLE BOX-part] [(BOTTOM BOX)])

[1,64](MOVING [JOHN-part])

[2,2](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[2,2](ROTATING [JOHN-part])

[2,15](MOVING-ROOT [JOHN-part])
[2,15](TRANSLATING [(EYE JOHN)] PLACE-2)
[2,15](MOVING-ROOT [(EYE JOHN)])
[2,15](MOVING [(EYE JOHN)])

[2,60](TRANSLATING [JOHN-part] PLACE-9)

[16,16](SUPPORTED [JOHN-part])
```

```
[16,17](PLACE [(EYE JOHN)] PLACE-3)

[18,18](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[18,18](ROTATING [JOHN-part])

[18,32](MOVING-ROOT [JOHN-part])
[18,32](TRANSLATING [(EYE JOHN)] PLACE-4)
[18,32](MOVING-ROOT [(EYE JOHN)])
[18,32](MOVING [(EYE JOHN)])

[33,33](SUPPORTED [JOHN-part])

[33,34](PLACE [(EYE JOHN)] PLACE-5)

[35,35](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[35,35](ROTATING [JOHN-part])
```

```
[35,48](MOVING-ROOT [JOHN-part])
[35,48](TRANSLATING [(EYE JOHN)] PLACE-6)
[35,48](MOVING-ROOT [(EYE JOHN)])
[35,48](MOVING [(EYE JOHN)])

[49,49](SUPPORTED [JOHN-part])

[49,50](PLACE [(EYE JOHN)] PLACE-7)

[51,51](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[51,51](ROTATING [JOHN-part])

[51,58](MOVING-ROOT [JOHN-part])
[51,58](TRANSLATING [(EYE JOHN)] PLACE-8)
[51,58](MOVING-ROOT [(EYE JOHN)])
[51,58](MOVING [(EYE JOHN)])

[59,64](SUPPORTED [JOHN-part])

[59,70](PLACE [(EYE JOHN)] PLACE-16)

[64,64](TRANSLATING [JOHN-part] PLACE-10)

[65,65](PLACE [JOHN-part] PLACE-12)

[66,71](TRANSLATING [BALL-part] PLACE-19)
[66,71](MOVING-ROOT [BALL-part])
[66,71](MOVING [BALL-part])
[66,71](SUPPORTED [JOHN-part])
[66,71](SUPPORTS [JOHN-part] [BALL-part])
[66,71](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-17)
[66,71](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[66,71](MOVING [(LINE-SEGMENT3 BALL)])
[66,71](SUPPORTS [(LINE-SEGMENT3 BALL)] [BALL-part])
[66,71](SUPPORTED [BALL-part JOHN-part])
[66,71](SUPPORTS [BALL-part JOHN-part] [(LINE-SEGMENT3 BALL)])

[67,67](TRANSLATING [JOHN-part] PLACE-15)
[67,67](TRANSLATING [BALL-part JOHN-part] PLACE-14)
```

```
[71,71](FLIPPING [BALL-part])
[71,71](ROTATING-COUNTER-CLOCKWISE [BALL-part])
[71,71](ROTATING [BALL-part])
[71,71](FLIPPING [JOHN-part])
[71,71](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[71,71](ROTATING-CLOCKWISE [JOHN-part])
[71,71](ROTATING [JOHN-part])
[71,71](MOVING-ROOT [JOHN-part])
[71,71](SUPPORTS [BALL-part] [JOHN-part])
[71,71](TRANSLATING [(EYE JOHN)] PLACE-18)
[71,71](ROTATING-COUNTER-CLOCKWISE [(EYE JOHN)])
[71,71](ROTATING [(EYE JOHN)])
[71,71](MOVING-ROOT [(EYE JOHN)])
[71,71](MOVING [(EYE JOHN)])
[71,71](ROTATING-CLOCKWISE [(LINE-SEGMENT3 BALL)])
[71,71](ROTATING [(LINE-SEGMENT3 BALL)])
[71,71](FLIPPING [BALL-part JOHN-part])
[71,71](ROTATING-COUNTER-CLOCKWISE [BALL-part JOHN-part])
[71,71](ROTATING-CLOCKWISE [BALL-part JOHN-part])
[71,71](ROTATING [BALL-part JOHN-part])
[71,71](MOVING-ROOT [BALL-part JOHN-part])
[71,71](SUPPORTS [(LINE-SEGMENT3 BALL)] [BALL-part JOHN-part])

[72,72](PLACE [BALL-part] PLACE-22)
[72,72](PLACE [(EYE JOHN)] PLACE-21)
[72,72](PLACE [(LINE-SEGMENT3 BALL)] PLACE-20)

[73,80](TRANSLATING [BALL-part] PLACE-25)
[73,80](MOVING-ROOT [BALL-part])
[73,80](MOVING [BALL-part])
[73,80](MOVING-ROOT [JOHN-part])
[73,80](TRANSLATING [(EYE JOHN)] PLACE-24)
[73,80](MOVING-ROOT [(EYE JOHN)])
[73,80](MOVING [(EYE JOHN)])
[73,80](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-23)
[73,80](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[73,80](MOVING [(LINE-SEGMENT3 BALL)])
[73,80](MOVING-ROOT [BALL-part JOHN-part])
[73,80](MOVING-ROOT [BALL JOHN-part])

[81,82](PLACE [BALL-part] PLACE-28)
[81,82](PLACE [(EYE JOHN)] PLACE-27)
[81,82](PLACE [(LINE-SEGMENT3 BALL)] PLACE-26)
```

```
[83,83](ROTATING-CLOCKWISE [JOHN-part])
[83,83](ROTATING [JOHN-part])
[83,83](ROTATING-CLOCKWISE [BALL-part JOHN-part])
[83,83](ROTATING [BALL-part JOHN-part])
[83,83](ROTATING-CLOCKWISE [BALL JOHN-part])
[83,83](ROTATING [BALL JOHN-part])

[83,97](TRANSLATING [BALL-part] PLACE-31)
[83,97](MOVING-ROOT [BALL-part])
[83,97](MOVING [BALL-part])
[83,97](MOVING-ROOT [JOHN-part])
[83,97](TRANSLATING [(EYE JOHN)] PLACE-30)
[83,97](MOVING-ROOT [(EYE JOHN)])
[83,97](MOVING [(EYE JOHN)])
[83,97](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-29)
[83,97](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[83,97](MOVING [(LINE-SEGMENT3 BALL)])
[83,97](MOVING-ROOT [BALL-part JOHN-part])
[83,97](MOVING-ROOT [BALL JOHN-part])

[98,99](PLACE [BALL-part] PLACE-34)
[98,99](PLACE [(EYE JOHN)] PLACE-33)
[98,99](PLACE [(LINE-SEGMENT3 BALL)] PLACE-32)

[100,100](ROTATING-CLOCKWISE [JOHN-part])
[100,100](ROTATING [JOHN-part])
[100,100](ROTATING-CLOCKWISE [BALL-part JOHN-part])
[100,100](ROTATING [BALL-part JOHN-part])
[100,100](ROTATING-CLOCKWISE [BALL JOHN-part])
[100,100](ROTATING [BALL JOHN-part])

[100,113](TRANSLATING [BALL-part] PLACE-37)
[100,113](MOVING-ROOT [BALL-part])
[100,113](MOVING [BALL-part])
[100,113](MOVING-ROOT [JOHN-part])
[100,113](TRANSLATING [(EYE JOHN)] PLACE-36)
[100,113](MOVING-ROOT [(EYE JOHN)])
[100,113](MOVING [(EYE JOHN)])
[100,113](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-35)
[100,113](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[100,113](MOVING [(LINE-SEGMENT3 BALL)])
[100,113](MOVING-ROOT [BALL-part JOHN-part])
[100,113](MOVING-ROOT [BALL JOHN-part])

[114,115](PLACE [BALL-part] PLACE-40)
[114,115](PLACE [(EYE JOHN)] PLACE-39)
[114,115](PLACE [(LINE-SEGMENT3 BALL)] PLACE-38)
```

```
[116,116](ROTATING-CLOCKWISE [JOHN-part])
[116,116](ROTATING [JOHN-part])
[116,116](ROTATING-CLOCKWISE [BALL-part JOHN-part])
[116,116](ROTATING [BALL-part JOHN-part])
[116,116](ROTATING-CLOCKWISE [BALL JOHN-part])
[116,116](ROTATING [BALL JOHN-part])

[116,130](TRANSLATING [BALL-part] PLACE-43)
[116,130](MOVING-ROOT [BALL-part])
[116,130](MOVING [BALL-part])
[116,130](MOVING-ROOT [JOHN-part])
[116,130](TRANSLATING [(EYE JOHN)] PLACE-42)
[116,130](MOVING-ROOT [(EYE JOHN)])
[116,130](MOVING [(EYE JOHN)])
[116,130](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-41)
[116,130](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[116,130](MOVING [(LINE-SEGMENT3 BALL)])
[116,130](MOVING-ROOT [BALL-part JOHN-part])
[116,130](MOVING-ROOT [BALL JOHN-part])

[131,131](PLACE [BALL-part] PLACE-46)
[131,131](PLACE [(EYE JOHN)] PLACE-45)
[131,131](PLACE [(LINE-SEGMENT3 BALL)] PLACE-44)
```

```
[132,132](FLIPPING [BALL-part])
[132,132](TRANSLATING [BALL-part] PLACE-49)
[132,132](ROTATING-COUNTER-CLOCKWISE [BALL-part])
[132,132](ROTATING-CLOCKWISE [BALL-part])
[132,132](ROTATING [BALL-part])
[132,132](MOVING-ROOT [BALL-part])
[132,132](MOVING [BALL-part])
[132,132](FLIPPING [JOHN-part])
[132,132](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[132,132](ROTATING-CLOCKWISE [JOHN-part])
[132,132](ROTATING [JOHN-part])
[132,132](MOVING-ROOT [JOHN-part])
[132,132](TRANSLATING [(EYE JOHN)] PLACE-48)
[132,132](ROTATING-CLOCKWISE [(EYE JOHN)])
[132,132](ROTATING [(EYE JOHN)])
[132,132](MOVING-ROOT [(EYE JOHN)])
[132,132](MOVING [(EYE JOHN)])
[132,132](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-47)
[132,132](ROTATING-COUNTER-CLOCKWISE [(LINE-SEGMENT3 BALL)])
[132,132](ROTATING [(LINE-SEGMENT3 BALL)])
[132,132](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[132,132](MOVING [(LINE-SEGMENT3 BALL)])
[132,132](FLIPPING [BALL-part JOHN-part])
[132,132](ROTATING-COUNTER-CLOCKWISE [BALL-part JOHN-part])
[132,132](ROTATING-CLOCKWISE [BALL-part JOHN-part])
[132,132](ROTATING [BALL-part JOHN-part])
[132,132](MOVING-ROOT [BALL-part JOHN-part])
[132,132](FLIPPING [BALL JOHN-part])
[132,132](ROTATING-COUNTER-CLOCKWISE [BALL JOHN-part])
[132,132](ROTATING-CLOCKWISE [BALL JOHN-part])
[132,132](ROTATING [BALL JOHN-part])
[132,132](MOVING-ROOT [BALL JOHN-part])

[133,133](PLACE [BALL-part] PLACE-52)
[133,133](PLACE [(EYE JOHN)] PLACE-51)
[133,133](PLACE [(LINE-SEGMENT3 BALL)] PLACE-50)

[134,134](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[134,134](ROTATING [JOHN-part])
[134,134](ROTATING-COUNTER-CLOCKWISE [BALL-part JOHN-part])
[134,134](ROTATING [BALL-part JOHN-part])
[134,134](ROTATING-COUNTER-CLOCKWISE [BALL JOHN-part])
[134,134](ROTATING [BALL JOHN-part])
```

```
[134,147](TRANSLATING [BALL-part] PLACE-55)
[134,147](MOVING-ROOT [BALL-part])
[134,147](MOVING [BALL-part])
[134,147](MOVING-ROOT [JOHN-part])
[134,147](TRANSLATING [(EYE JOHN)] PLACE-54)
[134,147](MOVING-ROOT [(EYE JOHN)])
[134,147](MOVING [(EYE JOHN)])
[134,147](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-53)
[134,147](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[134,147](MOVING [(LINE-SEGMENT3 BALL)])
[134,147](MOVING-ROOT [BALL-part JOHN-part])
[134,147](MOVING-ROOT [BALL JOHN-part])

[148,149](PLACE [BALL-part] PLACE-58)
[148,149](PLACE [(EYE JOHN)] PLACE-57)
[148,149](PLACE [(LINE-SEGMENT3 BALL)] PLACE-56)

[150,150](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[150,150](ROTATING [JOHN-part])
[150,150](ROTATING-COUNTER-CLOCKWISE [BALL-part JOHN-part])
[150,150](ROTATING [BALL-part JOHN-part])
[150,150](ROTATING-COUNTER-CLOCKWISE [BALL JOHN-part])
[150,150](ROTATING [BALL JOHN-part])

[150,163](TRANSLATING [BALL-part] PLACE-61)
[150,163](MOVING-ROOT [BALL-part])
[150,163](MOVING [BALL-part])
[150,163](MOVING-ROOT [JOHN-part])
[150,163](TRANSLATING [(EYE JOHN)] PLACE-60)
[150,163](MOVING-ROOT [(EYE JOHN)])
[150,163](MOVING [(EYE JOHN)])
[150,163](TRANSLATING [(LINE-SEGMENT3 BALL)] PLACE-59)
[150,163](MOVING-ROOT [(LINE-SEGMENT3 BALL)])
[150,163](MOVING [(LINE-SEGMENT3 BALL)])
[150,163](MOVING-ROOT [BALL-part JOHN-part])
[150,163](MOVING-ROOT [BALL JOHN-part])

[164,165](PLACE [BALL-part] PLACE-64)
[164,165](PLACE [(EYE JOHN)] PLACE-63)
[164,165](PLACE [(LINE-SEGMENT3 BALL)] PLACE-62)

[166,166](ROTATING-COUNTER-CLOCKWISE [JOHN-part])
[166,166](ROTATING [JOHN-part])
[166,166](ROTATING-COUNTER-CLOCKWISE [BALL-part JOHN-part])
[166,166](ROTATING [BALL-part JOHN-part])
[166,166](ROTATING-COUNTER-CLOCKWISE [BALL JOHN-part])
[166,166](ROTATING [BALL JOHN-part])
```

Bibliography

[1] Norihiro Abe, Itsuya Soga, and Saburo Tsuji. A plot understanding system on reference to both image and language. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 77–84, 1981.

[2] Mark R. Adler. Computer interpretation of peanuts cartoons. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, page 608, August 1977.

[3] Jonathan Amsterdam. The iterate manual. A. I. Memo 1236, M. I. T. Artificial Intelligence Laboratory, October 1990.

[4] John R. Anderson. A theory of language acquisition based on general learning principles. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 97–103, 1981.

[5] Norman I. Badler. Temporal scene analysis: Conceptual descriptions of object movements. Technical Report 80, University of Toronto Department of Computer Science, February 1975.

[6] Renée Baillargeon. Representing the existence and the location of hidden objects: Object permanence in 6- and 8-month-old infants. *Cognition*, 23:21–41, 1986.

[7] Renée Baillargeon. Object permanence in $3\frac{1}{2}$- and $4\frac{1}{2}$-month-old infants. *Developmental Psychology*, 23(5):655–664, 1987.

[8] Renée Baillargeon, Elizabeth S. Spelke, and Stanley Wasserman. Object permanence in five-month-old infants. *Cognition*, 20:191–208, 1985.

[9] Robert C. Berwick. Learning structural descriptions of grammar rules from examples. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 56–58, 1979.

[10] Robert C. Berwick. *Locality Principles and the Acquisition of Syntactic Knowledge*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1982.

[11] Robert C. Berwick. Learning word meanings from examples. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 459–461, 1983.

[12] M. Blum and L. Blum. Towards a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[13] Gary Borchardt. A computer model for the representation and identification of physical events. Technical Report T-142, Coordinated Sciences Laboratory, University of Illinois at Urbana-Champaign, May 1984.

[14] Michael Brent. Earning dividends on lexical knowledge: How the rich can get richer. In *Proceedings of the First Annual Workshop on Lexical Acquisition*, 1989.

[15] Michael Brent. Semantic classification of verbs from their syntactic contexts: Automated lexicography with implications for child language acquisition. In *Proceedings of the* 12th *Annual Conference of the Cognitive Science Society*, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[16] Michael Brent. *Automatic Acquisition of Subcategorization Frames from Unrestricted English.* PhD thesis, Massachusetts Institute of Technology, 1991.

[17] Michael Brent. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the* 29th *Annual Meeting of the Association for Computational Linguistics*, 1991.

[18] Michael Brent. Semantic classification of verbs from their syntactic contexts: An implemented classifier for stativity. In *Proceedings of the 5th European ACL Conference.* Association for Computational Linguistics, 1991.

[19] Noam Chomsky. *Aspects of The Theory of syntax.* The M. I. T. Press, Cambridge, MA and London, England, 1965.

[20] Noam Chomsky. *Barriers*, volume 13 of *Linguistic Inquiry Monographs.* The M. I. T. Press, Cambridge, MA and London, England, 1986.

[21] James F. Cremer. *An Architecture for General Purpose Physical System Simulation—Integrating Geometry, Dynamics, and Control.* PhD thesis, Cornell University, April 1989. available as TR 89–987.

[22] A. A. DiSessa. Phenomenology and evolution of intuition. In D. Gentner and A. L. Stevens, editors, *Mental Models*, pages 15–33. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

[23] Bonnie Jean Dorr. *Lexical Conceptual Structure and Machine Translation.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1990.

[24] Bonnie Jean Dorr. Solving thematic divergences in machine translation. In *Proceedings of the* 28th *Annual Meeting of the Association for Computational Linguistics*, pages 127–134, University of Pittsburgh, Pittsburgh, PA, June 1990.

[25] Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63, November 1989.

[26] Jerome A. Feldman, George Lakoff, Andreas Stolcke, and Susan Hollbach Weber. Miniature language acquisition: A touchstone for cognitive science. In *Proceedings of the* 12th *Annual Conference of the Cognitive Science Society*, pages 686–693, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[27] Ronald A. Finke and Jennifer J. Freyd. Transformations of visual memory induced by implied motions of pattern elements. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 11(2):780–794, 1985.

[28] Ronald A. Finke, Jennifer J. Freyd, and Gary C.-W. Shyi. Implied velocity and acceleration induce transformations of visual memory. *Journal of Experimental Psychology: General*, 115(2):175–188, 1986.

[29] Cynthia Fisher, Geoffry Hall, Susan Rakowitz, and Lila Gleitman. When it is better to receive than to give: syntactic and conceptual constraints on vocabulary growth. Unpublished manuscript received directly from author, 1991.

[30] Sandiway Fong. *Computational Properties of Principle-Based Grammatical Theories.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1991.

[31] Jennifer J. Freyd. The mental representation of movement when static stimuli are viewed. *Perception and Psychophysics*, 33:575–581, 1983.

[32] Jennifer J. Freyd. Dynamic mental representations. *Psychological Review*, 94:427–438, 1987.

[33] Jennifer J. Freyd and Ronald A. Finke. Representational momentum. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 10:126–132, 1984.

[34] Jennifer J. Freyd and Ronald A. Finke. A velocity effect for representational momentum. *Bulletin of the Psychonomic Society*, 23:443–446, 1985.

[35] Jennifer J. Freyd and J. Q. Johnson. Probing the time course of representational momentum. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 13(2):259–268, 1987.

[36] Jennifer J. Freyd, Teresa M. Pantzer, and Jeannette L. Cheng. Representing statics as forces in equilibrium. *Journal of Experimental Psychology, General*, 117(4):395–407, December 1988.

[37] Brian V. Funt. Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13:201–230, 1980.

[38] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, April–June 1983.

[39] E. J. Gibson, C. J. Owsley, A. Walker, and J. Megaw-Nyce. Development of hte perception of invariants: Substance and shape. *Perception*, 8:609–619, 1979.

[40] J. J. Gibson. On the relation between hallucination and perception. *Leonardo*, 3:425–427, 1970.

[41] J. J. Gibson. *The ecological approach to visual perception.* Houghton Mifflin, Boston, 1979.

[42] Lila Gleitman. The structural sources of verb meanings. *Language Acquisition*, 1(1):3–55, 1990.

[43] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[44] Richard H. Granger, Jr. FOUL-UP a program that figures out meanings of words from context. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 172–178, August 1977.

[45] Jane Grimshaw. Complement selection and the lexicon. *Linguistic Inquiry*, 10:279–326, 1979.

[46] Jane Grimshaw. Form, function, and the language acquisition device. In C. L. Baker and J. J. McCarthy, editors, *The logical problem of language acquisition.* The M. I. T. Press, Cambridge, MA and London, England, 1981.

[47] Henry Hamburger and Kenneth N. Wexler. A mathematical theory of learning transformational grammar. *Journal of Mathematical Psychology*, 12:137–177, 1975.

[48] Fritz Heider and Marianne Simmel. An experimental study of apparent behavior. *Journal of Psychology*, 57:243–259, 1944.

[49] Ray Jackendoff. *Semantics and Cognition.* The M. I. T. Press, Cambridge, MA and London, England, 1983.

[50] Ray Jackendoff. *Semantic Structures.* The M. I. T. Press, Cambridge, MA and London, England, 1990.

[51] T. Kasami. An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford MA, 1965.

[52] M. H. Kelly and Jennifer J. Freyd. Explorations of representational momentum. *Cognitive Psychology*, 19:369–401, 1987.

[53] Glenn Andrew Kramer. Geometric reasoning in the kinematic analysis of mechanisms. Technical Report TR–91–02, Schlumberger Laboratory for Computer Science, October 1990.

[54] Glenn Andrew Kramer. Solving geometric constraint systems. In *Proceedings of the Eighth National Conference on Artifical Intelligence*, pages 708–714. Morgan Kaufmann Publishers, Inc., July 1990.

[55] George Lakoff. *Women, Fire, and Dangerous Things.* The University of Chicago Press, 1987.

[56] D. Lebeaux. *Language Acquisition and the Form of Grammar.* PhD thesis, University of Massachusetts, Amherst, 1988.

[57] Beth Levin. Lexical semantics in review. Lexicon Project Working Papers #1, M. I. T. Center for Cognitive Science, February 1985.

[58] Beth Levin. Approaches to lexical semantic representation. m. s., February 1987.

[59] David Lightfoot. *How to Set Parameters: Arguments from Language Change.* The M. I. T. Press, Cambridge, MA and London, England, 1991.

[60] David Allen McAllester. Solving SAT problems via dependency directed backtracking. Unpublished manuscript received directly from author.

[61] David Allen McAllester. A three valued truth maintenance system. A. I. Memo 473, M. I. T. Artificial Intelligence Laboratory, May 1978.

[62] David Allen McAllester. An outlook on truth maintenance. A. I. Memo 551, M. I. T. Artificial Intelligence Laboratory, August 1980.

[63] David Allen McAllester. Reasoning utility package user's manual version one. A. I. Memo 667, M. I. T. Artificial Intelligence Laboratory, April 1982.

[64] M. McCloskey. Naive theories of motion. In D. Gentner and A. L. Stevens, editors, *Mental Models.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

[65] George A. Miller. English verbs of motion: A case study in semantics and lexical memory. In Arthur W. Melton and Edwin Martin, editors, *Coding Processes in Human Memory*, chapter 14, pages 335–372. V. H. Winston and Sons, Inc., Washington, DC, 1972.

[66] Thomas M. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 305–310, August 1977.

[67] Gordon S. Novak Jr. and William C. Bulko. Understanding natural language with diagrams. In *Proceedings of the Eighth National Conference on Artifical Intelligence*. Morgan Kaufmann Publishers, Inc., July 1990.

[68] Naoyuki Okada. SUPP: Understanding moving picture patterns based on linguistic knowledge. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 690–692, 1979.

[69] Fernando C. N. Pereira and David H. D. Warren. Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3):231–278, 1980.

[70] Steven Pinker. Formal models of language learning. *Cognition*, 7:217–283, 1979.

[71] Steven Pinker. *Language Learnability and Language Development*. Harvard University Press, Cambridge MA, 1984.

[72] Steven Pinker. Resolving a learnability paradox in the acquisition of the verb lexicon. Lexicon Project Working Papers #17, M. I. T. Center for Cognitive Science, July 1987.

[73] Steven Pinker. *Learnability and Cognition*. The M. I. T. Press, Cambridge, MA and London, England, 1989.

[74] James Pustejovsky. On the acquisition of lexical entries: The perceptual origin of thematic relations. In *Proceedings of the* 25th *Annual Meeting of the Association for Computational Linguistics*, pages 172–178, July 1987.

[75] James Pustejovsky. Constraints on the acquisition of semantic knowledge. *International Journal of Intelligent Systems*, 3(3):247–268, 1988.

[76] W. V. O. Quine. *Word and object*. The M. I. T. Press, Cambridge, MA and London, England, 1960.

[77] Manny Rayner, Åsa Hugosson, and Göran Hagert. Using a logic grammar to learn a lexicon. Technical Report R88001, Swedish Institute of Computer Science, 1988.

[78] Sharon C. Salveter. Inferring conceptual graphs. *Cognitive Science*, 3(2):141–166, 1979.

[79] Sharon C. Salveter. Inferring building blocks for knowledge representation. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*, chapter 12, pages 327–344. Lawrence Erlbaum Associates, Hillsdale, NJ, 1982.

[80] Roger C. Schank. The fourteen primitive actions and their inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory, March 1973.

[81] Mallory Selfridge. A computer model of child language acquisition. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 92–96, 1981.

[82] R. N. Shepard. Psychophysical complementarity. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, pages 279–341. Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

[83] R. N. Shepard. Ecological constraints on internal representations: Resonant kinematics of perceiving, imagining, thinking, and dreaming. *Psychological Review*, 91:417–447, 1984.

[84] Jeffrey Mark Siskind. Acquiring core meanings of words, represented as Jackendoff-style conceptual structures, from correlated streams of linguistic and non-linguistic input. In *Proceedings of the* 28th *Annual Meeting of the Association for Computational Linguistics*, pages 143–156, University of Pittsburgh, Pittsburgh, PA, June 1990.

[85] Jeffrey Mark Siskind. Dispelling myths about language bootstrapping. In *The AAAI Spring Symposium Workshop on Machine Learning of Natural Language and Ontology*, pages 157–164, March 1991.

[86] Jeffrey Mark Siskind and David Allen McAllester. Screamer: A portable efficient implementation of nondeterministic common lisp. Submitted to LFP92.

[87] Elizabeth S. Spelke. Cognition in infancy. Occasional Papers in Cognitive Science 28, Massachusetts Institute of Technology, 1983.

[88] Elizabeth S. Spelke. Where perceiving ends and thinking begins: the apprehension of objects in infancy. In A. Yonas, editor, *Perceptual Development in infancy. Minnesota symposia in child psychology*, pages 197–234. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[89] Elizabeth S. Spelke. The origins of physical knowledge. In L. Weiskrantz, editor, *Thought without Language*, chapter 7, pages 168–184. Clarendon Press, 1988.

[90] Jess Stein, Leonore C. Hauck, and P. Y. Su, editors. *The Random House Colledge Dictionary.* Random House, revised edition edition, 1975.

[91] Andreas Stolcke. Learning feature-based semantics with simple recurrent networks. Technical Report TR–90–015, International Computer Science Institute, April 1990.

[92] Andreas Stolcke. Vector space grammars and the acquisition of syntactic categories: Getting connectionist and traditional models to learn from each other. In *The AAAI Spring Symposium Workshop on Machine Learning of Natural Language and Ontology*, pages 174–179, March 1991.

[93] Patrick Suppes, Lin Liang, and Michael Böttner. Complexity issues in robotic machine learning of natural language. In L. Lam and V. Naroditsky, editors, *Modeling Complex Phenomena.* Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1991.

[94] Robert Thibadeau. Artificial perception of actions. *Cognitive Science*, 10(2):117–149, 1986.

[95] John K. Tsotsos. Some notes on motion understanding. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, page 611, August 1977.

[96] John K. Tsotsos and John Mylopoulos. ALVEN: A study on motion understanding by computer. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 890–892, 1979.

[97] Saburo Tsuji, Akira Morizono, and Shinichi Kuroda. Understanding a simple cartoon film by a computer vision system. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 609–610, August 1977.

[98] Saburo Tsuji, Michiharu Osada, and Masahiko Yachida. Three dimensional movement analysis of dynamic line images. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 896–901, 1979.

[99] Susan Hollbach Weber. Connectionist semantics for miniature language acquisition. In *The AAAI Spring Symposium Workshop on Machine Learning of Natural Language and Ontology*, pages 185–190, March 1991.

[100] Susan Hollbach Weber and Andreas Stolcke. L_0: A testbed for miniature language acquisition. Technical Report TR–90–010, International Computer Science Institute, July 1990.

[101] David A. Wolfram. Intractable unifiability problems and backtracking. In Ehud Shapiro, editor, *Proceedings of the Third International Conference on Logic Programming*, pages 107–121, Berlin, Heidelberg, New York, London, Paris, Tokyo, July 1986. Springer-Verlag. Also available as Lecture Notes in Computer Science #225.

[102] D. H. Younger. Recognition and parsing of context-free languages in time $O(n^3)$. *Information and Control*, 10(2):189–208, 1967.

[103] Ramin D. Zabih and David Allen McAllester. A rearrangement search strategy for determining propositional satisfiability. In *Proceedings of the Seventh National Conference on Artifical Intelligence*, pages 155–160, August 1988.