3-1988

A Lexical Extension of Montague Semantics

William C. Purdy
Syracuse University, wcpurdy@ecs.syr.edu

CIS 88-1

A LEXICAL EXTENSION OF MONTAGUE SEMANTICS

William C. Purdy

March 1988

SCHOOL OF COMPUTER
AND INFORMATION SCIENCE
SYRACUSE UNIVERSITY

A Lexical Extension of Montague Semantics

William C. Purdy*

March 1988

Abstract

Montague's linguistic theory provides a completely formalized account of language in general and natural language in particular. It would appear to be especially applicable to the problem of natural language understanding by computer systems. However the theory does not deal with meaning at the lexical level. As a result, deduction in a system based on Montague semantics is severely restricted. This paper considers lexical extension of Montague semantics as a way to remove this restriction. Representation of lexical semantics by a logic program or semantic net is complex. An alternative representation, called a semantic space, is described. This alternative lacks the expressiveness of a logic program but it offers conceptual simplicity and intrinsically parallel structure.

*School of Computer and Information Science, Syracuse University

1 INTRODUCTION

The purpose of this paper is to examine an approach to lexical semantics in relation to Montague's linguistic theory [12]. The discussion is informal, making use of examples to illustrate the material. More precise treatment may be found in the references provided.

Although developed in the context of Montague semantics, the approach is compatible with recent work in the tradition of Montague such as that of Dowty [4], Keenan and Faltz [8] and Hausser [7]. In any practical application, lexical semantics would be integrated with morphological analysis such as that advanced by Dowty and by Hausser.

1.1 A Brief Overview of Montgue Semantics

Montague's linguistic theory, developed during the 1960's, culminated in two definitive papers: "Universal Grammar" [12], referred to as "UG", and "The Proper Treatment of Quantification in Ordinary English" [12], known as "PTQ." In UG, Montague characterizes his approach:

> There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed, I consider it possible to comprehend the syntax and semantics of both kinds of languages within a single natural and mathematically precise theory.

As such a radical premise would suggest, Montague's approach differed sharply from that of most natural language theorists. First, as with formal languages, the "surface" structure of a natural language, rather than a hypothesized "deep" structure, is the object of analysis. Second, adherence to the principle of compositionality dictates a strict parallel between syntactic and semantic analysis. Third, all aspects of the theory are completely formalized.

In PTQ Montague defined an English fragment to illustrate the theory. This fragment also became known as "PTQ." Although small, PTQ was designed to embody many of the hard problems that a theory of language must account for. Montague also defined an intermediate language, the Intensional Logic (IL), to which PTQ is translated. IL provides higher order logical operators, tense and modal operators and a possible-worlds interpretation. The interpretation of IL, along with the translation of PTQ to IL, induces an interpretation of PTQ.

This two-step interpretation of PTQ is not required by Montague's theory. Direct interpretation has equal status in the theory. However, a two-step interpretation does provide certain advantages. Montague considered it important enough to give (in UG) a general theory of translation and of interpretation mediated by translation. Therefore, it is not inappropriate to characterize Montague's linguistic theory as comprising (see Figure 1)

1. structural (syntactic) analysis of expressions of the surface language
2. translation of these expressions to an intermediate language in accordance with
   (a) the structural analysis
   (b) formal translation rules selected by the structural analysis
   (c) the principle of compositionality
3. interpretation of the intermediate language in a possible-worlds model

As Montague points out, use of an appropriate intermediate language such as IL provides a perspicuous semantic characterization of expressions in the object language. Even more important, it seems clear that some such intermediate representation is a necessary construct in a theory of a natural language faculty of either organism or machine.

The semantics of PTQ can be understood by examining its translation rules. This brief overview will consider only extensional constructs, thus ignoring a most important part of Montague's contribution. However, it is adequate to motivate the discussion of lexical semantics, which is the topic of this paper.[1]

Expressions of PTQ are classified according to syntactic category. Each category contains basic expressions and phrases. The basic expressions are lexical items. Phrases are combinations of lexical items. Among the simplest of basic expressions is the common noun. A basic common noun translates to a unary predicate of IL, denoted by the common noun with a prime appended. The unary predicate is interpreted as a subset of individuals, i.e., an attribute. For example, the basic common noun **man** translates to **man′**.[2] This is denoted **man** $\Rightarrow_T$ **man′**. **man′** is interpreted as a particular subset of the universe.

Basic common nouns are syntactically primitive expressions from which more complex syntactic structures (phrases) are formed. As a consequence of the treatment

[1] This discussion of Montague's theory follows Dowty [5], using the modification of Bennett. To simplify the presentation, intensional operators are eliminated.

[2] To reduce the use of quotation marks, words and strings in the surface language are written in boldface. The corresponding IL expressions are boldface and primed.

of semantic evaluation as parallel to structural analysis, basic common nouns are semantically primitive as well. They are not decomposed by translation, but translate to unanalyzed predicates of IL. Basic expressions treated in this way are called *nonlogical constants.*

By contrast, *logical constants* are decomposed by translation. Determiners provide examples. A determiner translates to an IL expression that is interpreted as a relation between sets of individuals. For example, the determiner **every** translates to the IL expression $\lambda P \lambda Q \forall x[P(x) \to Q(x)]$. Since unary predicates are interpreted as sets of individuals, it is easy to see that the translation of **every** is interpreted as the inclusion relation.

The other determiners of PTQ are **a** and **the**. **a** is translated:
a $\Rightarrow_T \lambda P \lambda Q \exists x[P(x) \wedge Q(x)]$.
the is translated:
the $\Rightarrow_T \lambda P \lambda Q \exists x[\forall y[P(y) \leftrightarrow y = x] \wedge Q(x)]$,
or,[3]
the $\Rightarrow_T \lambda P \lambda Q[\exists x![P(x)] \wedge \exists x[P(x) \wedge Q(x)]]$.

The expressions to which determiners translate are examples of functors, expressions that combine with other expressions to form more complex expressions. Determiners combine syntactically with common nouns to form noun phrases, called term phrases in PTQ. In compliance with the principle of compositionality, translations of determiners combine with translations of common nouns to form translations of term phrases. For example, the determiner **every** combines with the common noun **man** to form the term phrase **every man**. Parallel to this structural operation, $\lambda P \lambda Q \forall x[P(x) \to Q(x)]$ combines with **man′** to form $(\lambda P \lambda Q \forall x[P(x) \to Q(x)])(\mathbf{man}')$. The latter expression is interpreted as a set of sets of individuals, viz., the set of attributes possessed by every man.

Another important category of logical constants is the proper nouns and pronouns. Since they are term phrases, playing the same syntactic role as **every man**, they translate to IL expressions of the same type, viz., sets of sets of individuals. For example,
John $\Rightarrow_T \lambda P[P(j)]$
which is interpreted as the set of attributes possessed by the individual named by the IL constant j. Similarly,[4]
$\mathbf{he}_1 \Rightarrow_T \lambda P[P(x_1)]$.

A simple example to summarize the presentation thus far is given in Figure 2. The surface string, **Every man talks**, has the structural analysis shown in Figure 2a.

[3] $\exists x![R(x)]$ is defined $\exists x[R(x) \wedge \forall y[R(y) \to y = x]]$.

[4] Pronouns are subscripted in the structural analysis of a surface string to make the pronominal reference unambiguous.

Each node of the tree corresponds to a formation rule. Figure 2b shows the parallel formation of the translation.

1.2 Deduction and Montague Semantics

The previous example illustrates the important point that Montague's theory is a model theory, not a proof theory. IL is a term algebra: it has no axioms or identities[5] that permit expressions to be established as equivalent. Equivalence and truth are established in the model.

However, it is possible to provide IL with axioms that will permit inference in IL [6]. This is a further advantage of an intermediate language. Provided with the axioms of λ-reduction, the expression of Figure 2 can be shown to be equivalent to $\forall x[\mathbf{man}'(x) \rightarrow \mathbf{talk}'(x)]$.

A more impressive example is the following. **Mary loves every man who loves her**, which in PTQ takes the form **Mary loves every man such that he loves her**, has a translation that is equivalent to $\forall y[\mathbf{man}'(y) \wedge \mathbf{love}'(y, m) \rightarrow \mathbf{love}'(m, y)]$. (See Appendix A for details of this and the following translations.) The PTQ sentences **John is a man** and **John loves Mary** have IL images equivalent to $\exists x[\mathbf{man}'(x) \wedge j = x]$ and $\mathbf{love}'(j, m)$, respectively. With the help of standard axioms of logic, the conclusion $\mathbf{love}'(m, j)$ can be deduced.

1.3 Deduction and Lexical Semantics

Deductive capability seems a necessary part of a natural language faculty. Suppose $\mathcal{S}$ is a set of English sentences which translate to the set $\mathcal{S}'$ of IL formulas. The set $\mathcal{S}'$ determines a set $\mathcal{W}$ of contexts or world states in which each member of $\mathcal{S}'$ is true. Let $\mathcal{E}'$ be the set of all formulas that are true in $\mathcal{W}$, i.e., the set of formulas entailed by $\mathcal{S}'$. The capability to recognize or generate (as the occasion demands) members of $\mathcal{E}'$ is prerequisite to any useful response. Indeed the extent of those formulas of $\mathcal{E}'$ that can be recognized or generated might be taken as an operational definition of the extent to which the natural language faculty "understands $\mathcal{S}$."

One might conclude therefore that the mathematically precise theory of Montague offers a most promising approach to realization of a natural language faculty in a machine. But this optimistic conclusion is not wholly justified because deduction in

[5]IL does have a small number of "meaning postulates" (for an example, see Appendix A). However, these meaning postulates are simply terms that must be true in any interpretation and thus restrict the possible interpretations of IL.

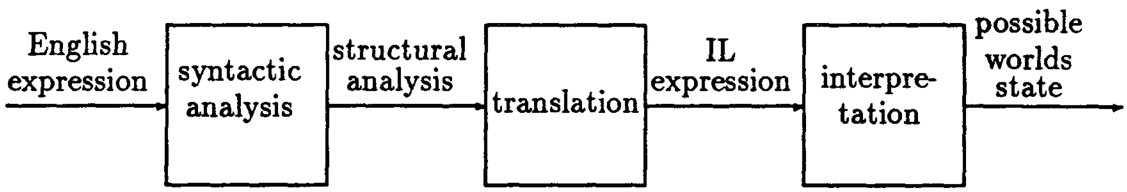
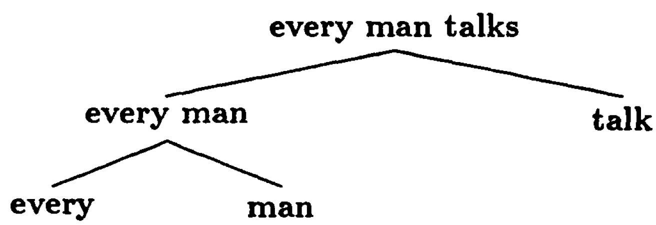
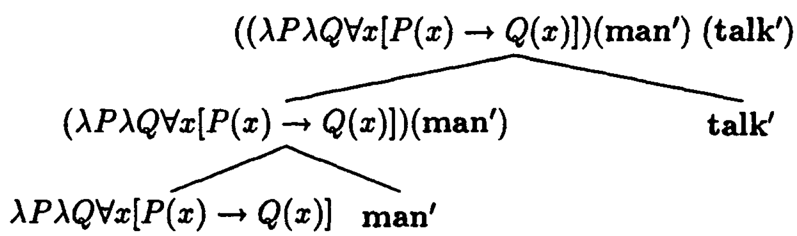


Figure 1: Characterization of Montague linguistic theory



(a) Structural Analysis



(b) Translation

Figure 2: Example of structural analysis and translation

IL is severely limited.

Continuing the previous example, consider the PTQ-like sentences **An actor loves Mary** and **Bill adores Mary** with translations in IL equivalent to $\exists x[\mathbf{actor}'(x) \wedge \mathbf{love}'(x, m)]$ and $\mathbf{adore}'(b, m)$, respectively. What can be deduced from these new sentences in conjunction with the previous ones? One would wish that $\exists x[\mathbf{actor}'(x) \wedge \mathbf{love}'(m, x)]$ (image of **Mary loves an actor**) and $\mathbf{love}'(m, b)$ (image of **Mary loves Bill**) could be deduced since this is surely within the capability of a competent English speaker.

But clearly, since **actor, man, adore** and **love** are nonlogical constants, translating to **actor**$'$, **man**$'$, **adore**$'$ and **love**$'$ respectively, the desired deductions cannot be made in IL with only the axioms of logic and λ-reduction. Expressions $\forall x[\mathbf{actor}'(x) \rightarrow \mathbf{man}'(x)]$ and $\forall x \forall y[\mathbf{adore}'(x, y) \rightarrow \mathbf{love}'(x, y)]$ are also needed.

This problem will be referred to as the *problem of lexical semantics.* The typical, if not the only, solution to this problem has been the creation of a relational database into which implicit definitions of lexical items, such as those of the previous paragraph, are placed. The same database usually contains also any other knowledge thought to be required by the system. The database may be called a semantic net, a relational hierarchy, or a production system. In any case, this solution is less than ideal because the structure is computationally unwieldy.

2 Approaches to Lexical Semantics

To illustrate the representation of lexical semantics by a relational database and to introduce an alternative approach to lexical semantics, a running example will be used. This example deals with English words for kinship.

The initial definition of the kinship vocabulary, shown in Figure 3, is taken from Eugene Nida's book, *Compositional Analysis of Meaning* [10].

The elements of the vocabulary are listed at the top of the table. "Diagnostic components," properties that distinguish elements of the vocabulary each from the other, appear along the left side of the table. The body of the table indicates which diagnostic components characterize each vocabulary element.

Consideration will at first be restricted to consanguineal kinship (c-kinship). In a later section, partial consanguineal relations will be added. Finally affinal kinship will be considered.

2.1 Logic Programs

Considering kinship as a binary relation, c-kinship can be viewed as a relational structure. Axioms of the structure define elements of the vocabulary. For example, **father′** is defined by the axiom[6] $\mathsf{father}(x,y) \leftrightarrow \mathsf{male}(x) \wedge \mathsf{prec}(x,y) \wedge \mathsf{LO}(x,y)$ where $\mathsf{prec}(x,y)$ asserts that x is of the generation preceeding that of y and $\mathsf{LO}(x,y)$ asserts a direct lineal relation between x and y. If $\mathsf{male}(x,y)$ is taken to assert that x is male, and application is defined to distribute over Boolean operations, the above can be written more compactly $\mathsf{father}(x,y) \leftrightarrow (\mathsf{male} \wedge \mathsf{prec} \wedge \mathsf{LO})(x,y)$. The axioms can also be viewed as a logic program, viz., a logic definition of the lexical items of the c-kinship vocabulary. Definition of c-kinship as a logic program is given in Figure 4.

A relation R_1 is said to be contained by or included in a relation R_2 if for all pairs (x,y), $\mathsf{R}_1(x,y) \rightarrow \mathsf{R}_2(x,y)$. To illustrate this, c-kinship has been extended in Figure 4 to include definitions of the lexical items **self, parent, child, sibling** and **immediate family**. From the definitions of these new lexical items it can be deduced for example that $\mathsf{sister}(x,y) \rightarrow \mathsf{sibling}(x,y)$ and $\mathsf{sibling}(x,y) \rightarrow \mathsf{immediate\ family}(x,y)$.

This suggests an approach to deduction involving lexical items. Nida [10] shows how linguistic analysis can identify diagnostic components for a given set of related lexical items (a "semantic domain"). The kinship relations are an example. The derived relational structure or logic program can then supplement the Intensional Logic to

[6] Predicates are written in sans serif type.

	father	mother	uncle	aunt	brother	sister	son	daughter	nephew	niece	cousin
male	X		X		X		X		X		X
female		X		X		X		X		X	X
prec.gen.	X	X	X	X							X
same gen.					X	X					X
succ.gen.							X	X	X	X	X
dir.lin.	X	X					X	X			
once rem.			X	X	X	X			X	X	
twice rem.											X
consang.	X	X	X	X	X	X	X	X	X	X	X
affinal			X	X					X	X	

Figure 3: Definition of Kinship Relations (from Nida)

male (x,y) $\vee$ female (x,y)
male (x,y) $\rightarrow$ $\neg$ female (x,y)
prec (x,y) $\vee$ same (x,y) $\vee$ succ (x,y)
prec (x,y) $\rightarrow$ $\neg$ same (x,y) $\vee \neg$ succ (x,y)
same (x,y) $\rightarrow$ $\neg$ succ (x,y)
L0 (x,y) $\vee$ L1 (x,y) $\vee$ L2 (x,y)
L0 (x,y) $\rightarrow$ $\neg$ L1 (x,y) $\wedge \neg$ L2 (x,y)
L1 (x,y) $\rightarrow$ $\neg$ L2 (x,y)
father (x,y) $\leftrightarrow$ male (x,y) $\wedge$ prec (x,y) $\wedge$ L0 (x,y)
mother (x,y) $\leftrightarrow$ female (x,y) $\wedge$ prec (x,y) $\wedge$ L0 (x,y)
uncle (x,y) $\leftrightarrow$ male (x,y) $\wedge$ prec (x,y) $\wedge$ L1 (x,y)
aunt (x,y) $\leftrightarrow$ female (x,y) $\wedge$ prec (x,y) $\wedge$ L1 (x,y)
brother (x,y) $\leftrightarrow$ male (x,y) $\wedge$ same (x,y) $\wedge$ L1 (x,y)
sister (x,y) $\leftrightarrow$ female (x,y) $\wedge$ same (x,y) $\wedge$ L1 (x,y)
son (x,y) $\leftrightarrow$ male (x,y) $\wedge$ succ (x,y) $\wedge$ L0 (x,y)
daughter (x,y) $\leftrightarrow$ female (x,y) $\wedge$ succ (x,y) $\wedge$ L0 (x,y)
nephew (x,y) $\leftrightarrow$ male (x,y) $\wedge$ succ (x,y) $\wedge$ L1 (x,y)
niece (x,y) $\leftrightarrow$ female (x,y) $\wedge$ succ (x,y) $\wedge$ L1 (x,y)
cousin (x,y) $\leftrightarrow$ L2 (x,y)
self (x,y) $\leftrightarrow$ same (x,y) $\wedge$ L0 (x,y)
parent (x,y) $\leftrightarrow$ prec (x,y) $\wedge$ L0 (x,y)
child (x,y) $\leftrightarrow$ succ (x,y) $\wedge$ L0 (x,y)
sibling (x,y) $\leftrightarrow$ same (x,y) $\wedge$ L1 (x,y)
immediate family (x,y) $\leftrightarrow$ L0 (x,y) $\vee$ (same (x,y) $\wedge$ L1 (x,y))

Figure 4: C-kinship as a Logic Program

extend the deductive capability. To illustrate, c-kinship is given in clausal form in Figures 5 and 6. Examples of deduction using resolution are shown in Figures 7 and 8.

The obvious advantage of a logic program as a representation of lexical semantics is its expressiveness. For example, it can be asserted that the parent relation is inverse to the child relation:

$$\mathsf{parent}(x, y) \leftrightarrow \mathsf{child}(y, x)$$

Or, it can be asserted that the uncle relation entails a brother relation:

$$\mathsf{uncle}(x, y) \rightarrow \exists z[\mathsf{brother}(x, z)]$$

There are also disadvantages. Some are illustrated by the examples of deduction. In a resolution based system, special procedures are required to avoid infinite loops when if and only if definitions are used. When the system is restricted to Horn clauses, assertion of disjunctions is awkward.

In any case, deduction is computationally quite complex, even for simple vocabularies, as shown by the examples.

2.2 Semantic Networks

Another representation of lexical semantics should be mentioned: the semantic network. There is not a consensus on exactly what constitutes a semantic network. A number of definitions are found in the literature [1]. One representation for c-kinship is the AKO (a-kind-of) network.

AKO networks are limited in what they can represent. However, Deliyanni and Kowalski [2] have shown that an appropriately extended form of semantic network has all the expressive capability of clausal form logic. Indeed, they show that their extended semantic network can be regarded as a syntactic variant of clausal form representation.

It is reasonable to assume that more restricted forms of semantic networks will be equivalent to similarly restricted clausal form logic. Therefore, semantic networks, while enjoying a certain graphic appeal, share the computational complexity of clausal form representations of lexical semantics.

male (x, y), female (x, y)
-male (x, y), -female (x, y)
prec (x, y), same (x, y), succ (x, y)
-prec (x, y), -same (x, y)
-prec (x, y), -succ (x, y)
-same (x, y), -succ (x, y)
L0 (x, y), L1 (x, y), L2 (x, y)
-L0 (x, y), -L1 (x, y)
-L0 (x, y), -L2 (x, y)
-L1 (x, y), -L2 (x, y)
-father (x, y), male (x, y)
-father (x, y), prec (x, y)
-father (x, y), L0 (x, y)
father (x, y), -male (x, y), -prec (x, y), -L0 (x, y)
-mother (x, y), female (x, y)
-mother (x, y), prec (x, y)
-mother (x, y), L0 (x, y)
mother (x, y), -female (x, y), -prec (x, y), -L0 (x, y)
-uncle (x, y), male (x, y)
-uncle (x, y), prec (x, y)
-uncle (x, y), L1 (x, y)
uncle (x, y), -male (x, y), -prec (x, y), -L1 (x, y)
-aunt (x, y), female (x, y)
-aunt (x, y), prec (x, y)
-aunt (x, y), L1 (x, y)
aunt (x, y), -female (x, y), -prec (x, y) -L1 (x, y)
-brother (x, y), male (x, y)
-brother (x, y), same (x, y)
-brother (x, y), L1 (x, y)
brother (x, y), -male (x, y), -same (x, y), -L1 (x, y)
-sister (x, y), female (x, y)
-sister (x, y), same (x, y)
-sister (x, y), L1 (x, y)
sister (x, y), -female (x, y), -same (x, y), -L1 (x, y)

Figure 5: C-Kinship in Clausal Form (First Half)

-son (x,y), male (x,y)
-son (x,y), succ (x,y)
-son (x,y), L0 (x,y)
son (x,y), -male (x,y), -succ (x,y) -L0 (x,y)
-daughter (x,y), female (x,y)
-daughter (x,y), succ (x,y)
-daughter (x,y), L0 (x,y)
daughter (x,y), -female (x,y), -succ (x,y), -L0 (x,y)
-nephew (x,y), male (x,y)
-nephew (x,y), succ (x,y)
-nephew (x,y), L1 (x,y)
nephew (x,y), -male (x,y), -succ (x,y), -L1 (x,y)
-niece (x,y), female (x,y)
-niece (x,y), succ (x,y)
-niece (x,y), L1 (x,y)
niece (x,y), -female (x,y), -succ (x,y), -L1 (x,y)
-cousin (x,y), L2 (x,y)
cousin (x,y), -L2 (x,y)
-self (x,y), same (x,y)
-self (x,y), L0 (x,y)
self (x,y), -same (x,y), -L0 (x,y)
-parent (x,y), prec (x,y)
-parent (x,y), L0 (x,y)
parent (x,y), -prec (x,y), -L0 (x,y)
-child (x,y), succ (x,y)
-child (x,y), L0 (x,y)
child (x,y), -succ (x,y), -L0 (x,y)
-sibling (x,y), same (x,y)
-sibling (x,y), L1 (x,y)
sibling (x,y), -same (x,y), -L1 (x,y)
-immediate family (x,y), same (x,y), L0 (x,y)
-immediate family (x,y), L0 (x,y), L1 (x,y)
immediate family (x,y), -L0 (x,y)
immediate family (x,y), -same (x,y), -L1 (x,y)

Figure 6: C-Kinship in Clausal Form (Second Half)

Example 1. **Father′ entails parent′**

(1)	father (a,b)	denial
(2)	-parent (a,b)	denial
(3)	-father (x,y), prec (x,y)	given
(4)	-father (x,y), L0 (x,y)	given
(5)	parent (x,y), -prec (x,y), -L0 (x,y)	given
(6)	-prec (a,b), -L0 (a,b)	from (2,5)
(7)	-parent (x,y), prec (x,y)	given
(8)	-parent (a,b), -L0 (a,b)	from (6,7)
(9)	-prec (a,b), -L0 (a,b)	from (5,8)
(10)	-parent (a,b), -L0 (a,b)	from (7,9)
$\vdots$	(loop)	
(7′)	prec (a,b)	from (1,3)
(8′)	L0 (a,b)	from (1,4)
(9′)	-L0 (a,b)	from (6,7′)
(10′)	$\square$	from (8′,9′)

Example 2. **Child′ entails immediate family′**

(1)	child (a,b)	denial
(2)	-immediate family (a,b)	denial
(3)	-child (x,y), L0 (x,y)	given
(4)	L0 (a,b)	from (1,3)
(5)	immediate family (x,y), -L0 (x,y)	given
(6)	immediate family (a,b)	from (4,5)
(7)	$\square$	from (2,6)

Figure 7: Deduction Using Resolution

2.3 Semantic Spaces

Consider the interpretation implied by Nida's definition of c-kinship. Let H be a set of individuals. The power set $2^{H \times H}$ represents the set of all binary relations on H. Indeed, a binary relation is typically identified with the set of pairs that satisfy it. For example, prec is identified with the set $\{(x, y) \in H \times H | \mathsf{prec}(x, y)\}$.

Let $S \subseteq H \times H$ be a subset of consanguineal pairs such that {prec,same,succ} partitions S. That is,

1. $\mathsf{prec} \cup \mathsf{same} \cup \mathsf{succ} = S$
2. $\mathsf{prec} \cap (\mathsf{same} \cup \mathsf{succ}) = \emptyset$
   $\mathsf{same} \cap \mathsf{succ} = \emptyset$
3. $\mathsf{prec} \neq \emptyset$
   $\mathsf{same} \neq \emptyset$
   $\mathsf{succ} \neq \emptyset$

Let {L0,L1,L2} and {male,female} also partition S.

S can be diagrammed as in Figure 9a, or to suggest a multidimensional space, as in Figure 9b. In this multidimensional space, subspaces or subsets are denotations of c-kinship relations. For example, the subspace $\mathsf{parent} = \mathsf{prec} \cap \mathsf{L0}$ is the denotation of **parent′**. Some examples of subspaces are given in Figure 10.

Thus a subspace can be viewed as the extension or meaning of the associated lexical item. Moreover, relations between subspaces can be viewed as relations between meanings. Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be any c-kinship lexical items, $\mathbf{R}_1{}'$ and $\mathbf{R}_2{}'$ their respective translations, and R_1 and R_2 their respective denotations (subspaces). (Cf. Figure 1.) Then $\mathbf{R}_1$ entails $\mathbf{R}_2$ if and only if $\forall (x, y) : \mathbf{R'}_1(x, y) \rightarrow \mathbf{R'}_2(x, y)$ if and only if $\mathsf{R}_1 \subseteq \mathsf{R}_2$. Thus subspace inclusion can be viewed as entailment or meaning inclusion. Similarly, subspace exclusion (disjointness) can be viewed as contradiction. The intersection of two subspaces can be viewed as the meaning common to the corresponding lexical items.

In the c-kinship space, inclusion, exclusion, intersection and the like can be determined quite directly. The examples of Figure 11 illustrate this. Since these examples were used as illustrations of deduction with the logic program for c-kinship, they also afford a comparision of the two representations.

The partitions that subdivide the multidimensional space in the preceeding example have an important property that was not made explicit. Residence in any given block of the partition {prec,same,succ} in no way restricts residence in any block of the partition {L0,L1,L2}. A similar assertion holds for any subset of the three partitions. This property is called independence.

More precisely, let $B = \{P_i | i \in I\}$ be a set of partitions of a set S, where $P_i = \{p_i^j | j \in J_i\}$. Then B will be said to be *independent* if and only if for any finite $I' \subseteq I$ and for any $j_i \in J_i$, $\bigcap_{i \in I'} p_i^j$ is nonempty.[7]

An independent set of partitions of a set S will be called a *basis of* S. The partitions of a basis of S define dimensions of S. Their blocks correspond to the coordinate values. Thus each partition can be viewed as a dimension of meaning. The blocks can be viewed as mutually antonymous "primitive" meanings.

Geometrically each block can be thought of as a hyperplane orthogonal to a coordinate axis. These hyperplanes are the simplest subspaces. Next in order of simplicity are those subspaces that can be expressed as the intersection of such hyperplanes, one or the union of several from each dimension.

In the c-kinship space defined previously, prec corresponds to a plane orthogonal to the "generation" axis. The intersection of prec, L0 (a plane orthogonal to the "lineality" axis) and male ∪ female (union of planes orthogonal to the "gender" axis) is the subspace previously identified as the extension of **parent′**. Such subspaces are analogous to convex subspaces. They will be called "elementary subsets."

More precisely, if $B = \{P_i | i \in I\}$ is a basis of S where $P_i = \{p_i^j | j \in J_i\}$, then an *elementary subset of* S *relative to the basis* B is a subspace x that can be represented $x = \bigcap_{i \in I} \bigcup_{j \in J_i^x} p_i^j$ where $J_i^x \subseteq J_i$.

The smallest nonempty elementary subsets are the intersections of hyperplanes where exactly one hyperplane is orthogonal to each coordinate axis. These elementary subsets are called *atoms*. For example, father = prec ∩ L0 ∩ male is an atom.

[7] In this paper it will be assumed that all partitions are finite and all sets of partitions are finite. Therefore B is independent if and only if for any selection of $j_i \in J_i$, for each $i \in I$, $\bigcap_{i \in I} p_i^j$ is nonempty.

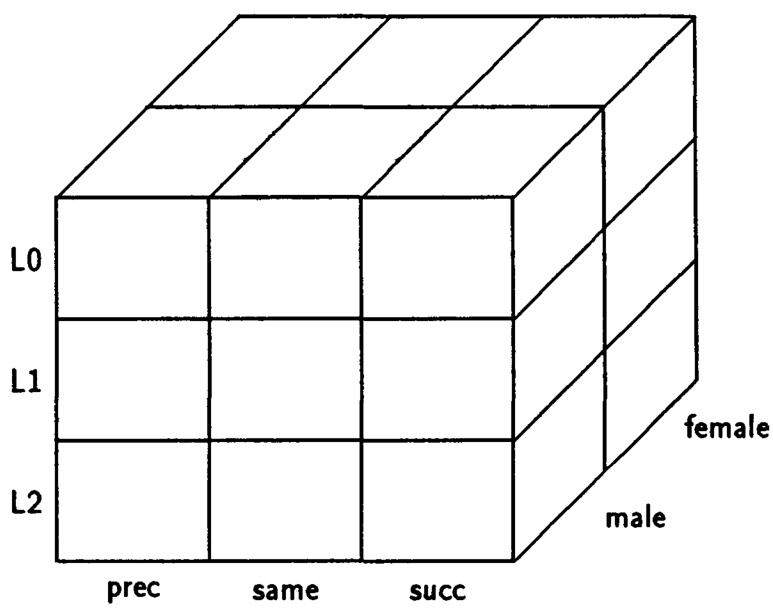
Example 3. **Uncle′** entails $\neg$ **immediate family′**

(1)	uncle (a, b)	denial
(2)	immediate family (a, b)	denial
(3)	-uncle (x, y), prec (x, y)	given
(4)	-uncle (x, y) L1 (x, y)	given
(5)	prec (a, b)	from (1,3)
(6)	L1 (a, b)	from (1,4)
(7)	-immediate family (x, y), same (x, y), L0 (x, y)	given
(8)	same (a, b), L0 (a, b)	from (2,7)
(9)	-prec (x, y), -same (x, y)	given
(10)	-same (a, b)	from (5,9)
(11)	L0 (a, b)	from (8,10)
(12)	-L0 (x, y), -L1 (x, y)	given
(13)	-L0 (a, b)	from (6,12)
(14)	$\square$	from (11,13)

Figure 8: Deduction Using Resolution (continued)

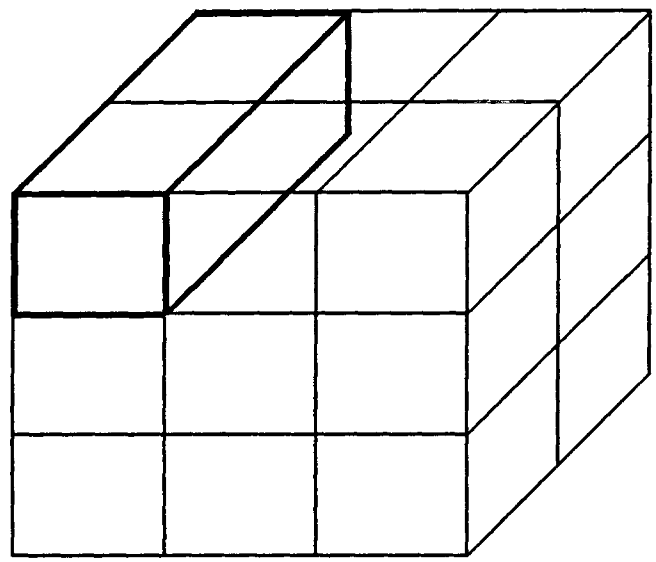
	male			female		
L0	father	self	son	mother	self	daughter
L1	uncle	brother	nephew	aunt	sister	niece
L2	cousin	cousin	cousin	cousin	cousin	cousin
	prec	same	succ	prec	same	succ

(a) Planar Representation

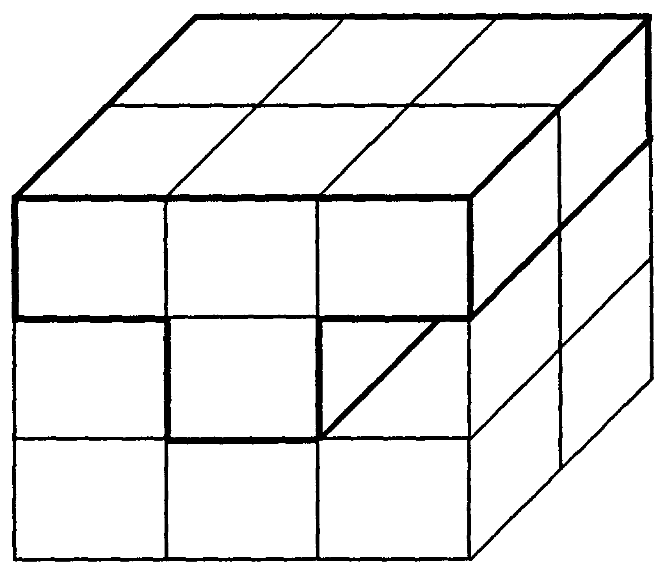


(b) Spatial Representation

Figure 9: C-Kinship as a Multidimensional Space



(a) parent



(b) immediate family

Figure 10: Subspaces of the C-Kinship Semantic Space

Example 1. **Father′** entails **parent′**

(1) **father′** $\mapsto$ prec $\cap$ L0 $\cap$ male

(2) **parent′** $\mapsto$ prec $\cap$ L0

(3) prec $\cap$ L0 $\cap$ male $\subseteq$ prec $\cap$ L0

Example 2. **Child′** entails **immediate family′**

(1) **child′** $\mapsto$ succ $\cap$ L0

(2) **immediate family′** $\mapsto$ L0 $\cup$ same $\cap$ L1

(3) succ $\cap$ L0 $\subseteq$ L0 $\subseteq$ L0 $\cup$ same $\cap$ L1

Example 3. **Uncle′** entails $\neg$ **immediate family′**

(1) **uncle′** $\mapsto$ prec $\cap$ L1 $\cap$ male

(2) **immediate family′** $\mapsto$ L0 $\cup$ same $\cap$ L1

(3) (prec $\cap$ L1 $\cap$ male) $\cap$ (L0 $\cup$ same $\cap$ L1)
$=$ (prec $\cap$ L0 $\cap$ L1 $\cap$ male) $\cup$ (prec $\cap$ same $\cap$ L1 $\cap$ male)
$= 0$

Figure 11: Entailment as inclusion

3 A Normal Form

Any arbitrary subspace is a union of elementary subsets. Trivially, any subspace is a union of atoms. In general, there are many distinct sets of elementary subsets each having as its union the same subspace. For example,
{prec ∩ L0, same ∩ L0, succ ∩ L0, same ∩ L1}
{(prec ∪ succ) ∩ L0, same ∩ (L0 ∪ L1)}
{L0, same ∩ L1}
{L0, same ∩ (L0 ∪ L1)}
are each a set of elementary subsets whose union is immediate family.

If x is an arbitrary subspace and y is an elementary subset contained in x, then y is *maximal in x* if no other elementary subset z in x properly contains y. That is, if for every elementary subset $z \subseteq x$, $y \subseteq z \subseteq x$ implies $z = y$, then y is maximal in x.

An elementary subset was defined as a subspace that can be represented as an intersection of unions, where each union consists of blocks from a given partition and the intersection ranges over all partitions in the basis. That is, elementary subset x can be represented $x = \bigcap_{i \in I} \bigcup_{j \in J_i^x} p_i^j$. It can be shown [11] that this representation, called the *standard form* for elementary subset x, is unique. An equivalent representation that would be convenient for machine implementation is the sequence $\langle J_i^x \rangle_{i \in I}$.

It can also be shown [11] that if x is an arbitrary subspace the set of elementary subsets that are maximal in x is unique to x. Thus any subspace is the union of a unique set of maximal elementary subsets, each of which has a unique standard form. The set of maximal elementary subsets of a subspace therefore constitutes a unique representation or *normal form* for that subspace. Consequently each extension or meaning has a normal form.

Continuing the running example, immediate family has the normal form
{L0, same ∩ (L0 ∪ L1)}
or, putting each elementary subset in standard form,[8]
{(prec ∪ same ∪ succ) ∩ L0 ∩ (male ∪ female), same ∩ (L0 ∪ L1) ∩ (male ∪ female)}.
Notice that no elementary subset in immediate family properly contains either of the elementary subsets in the normal form. Moreover, every elementary subset in immediate family is contained in one of the elementary subsets in the normal form.

The normal form of a subspace x will be denoted $\mathcal{N}(x)$.

[8] The expression {L0, same ∩ (L0 ∪ L1)} provides all the information that the expression {(prec ∪ same ∪ succ) ∩ L0 ∩ (male ∪ female), same ∩ (L0 ∪ L1) ∩ (male ∪ female)} does. The first form will be called the ***abbreviated standard form***. The definition is as follows. Let $i \in I^x$ if and only if $J_i^x \neq J_i$. Then $x = \bigcap_{i \in I^x} \bigcup_{j \in J_i^x} p_i^j$ is the abbreviated standard form for x.

Having defined a normal form for subspaces of the multidimensional space of lexical meaning, the next task is to identify useful operations under which the set of normal forms is closed.

In the simple case of elementary subsets, geometric intuition may be invoked. Let x and y be elementary subsets with standard forms $\bigcap_{i\in I}\bigcup_{j\in J_i^x} p_i^j$ and $\bigcap_{i\in I}\bigcup_{j\in J_i^y} p_i^j$ respectively. One is easily convinced by geometrical considerations that $x \cap y$ is also an elementary subset and moreover that its standard form is $\bigcap_{i\in I}\bigcup_{j\in J_i^x\cap J_i^y} p_i^j$. (See Figure 12 for an example.) That is, intersection of elementary subsets is computed componentwise. This result will not be proved here. However, this and all subsequent results leading to a Boolean algebra of normal forms are proved in [11].

Now consider the elementary subset $z_i = \bigcup_{j\in(J_i-J_i^x)} p_i^j$. From the previous result, it follows that $x \cap z_i = 0$ (the null subspace) for each $i \in I$, since $J_i^x \cap (J_i - J_i^x) = \emptyset$. Since the distributive law holds for the multidimensional space, $x \cap (\bigcup_{i\in I} z_i) = 0$ as well. Further, $x \cup (\bigcup_{i\in I} z_i) = 1$ (the unit subspace). Thus, $\bigcup_{i\in I} z_i$ is the complement of subspace x. (See Figure 13 for an example.) The complement will be denoted $-x$. Of course, $-x$ is not in general an elementary subset. But notice that the z_i are maximal in $-x$ and are irredundant. Therefore, $\{z_i | i \in I\} = \mathcal{N}(-x)$. For the special case where x is an elementary subset, define $\sim \mathcal{N}(x) = \{\bigcup_{j\in(J_i-J_i^x)} p_i^j | i \in I\}$. Then if x is an elementary subset, $\mathcal{N}(-x) = \sim \mathcal{N}(x)$.

Next consider arbitrary subspaces x and y with $\mathcal{N}(x) = \{x_1, x_2, \ldots, x_m\}$ and $\mathcal{N}(y) = \{y_1, y_2, \ldots, y_l\}$. Since by definition $x = x_1 \cup x_2 \cup \cdots \cup x_m$ and $y = y_1 \cup y_2 \cup \cdots \cup y_l$, it follows by distributivity that $x \cap y = \bigcap_{1\le r\le m, 1\le q\le l} x_r \cap y_q$. Each of the $x_r \cap y_q$ is an elementary subset. Moreover, the set $\{x_r \cap y_q | 1 \le r \le m, 1 \le q \le l\}$ contains all the maximal elementary subsets in $x \cap y$. It does not, however, contain *only* the maximal elementary subsets. Therefore, letting **irr** be the operation that removes subsumed elements, $\mathcal{N}(x \cap y) = \mathbf{irr}\{x_r \cap y_q | 1 \le r \le m, 1 \le q \le l\}$. Define $\mathcal{N}(x)\triangle\mathcal{N}(y) = \mathbf{irr}\{x_r \cap y_q | 1 \le r \le m, 1 \le q \le l\}$. Then the set of normal forms is closed under $\triangle$ and $\mathcal{N}(x \cap y) = \mathcal{N}(x)\triangle\mathcal{N}(y)$.

By De Morgan's law, $-x = -x_1 \cap -x_2 \cap \cdots \cap -x_m$, where each $-x_r$ is the complement of an elementary subset. Applying the result for intersection of normal forms, $\mathcal{N}(-x) = \mathcal{N}(-x_1)\triangle\cdots\triangle\mathcal{N}(-x_m)$ or $\sim \mathcal{N}(x) = \sim \mathcal{N}(x_1)\triangle\cdots\triangle \sim \mathcal{N}(x_m)$. Thus $\sim$ is defined for arbitrary subspaces as well as elementary subsets.

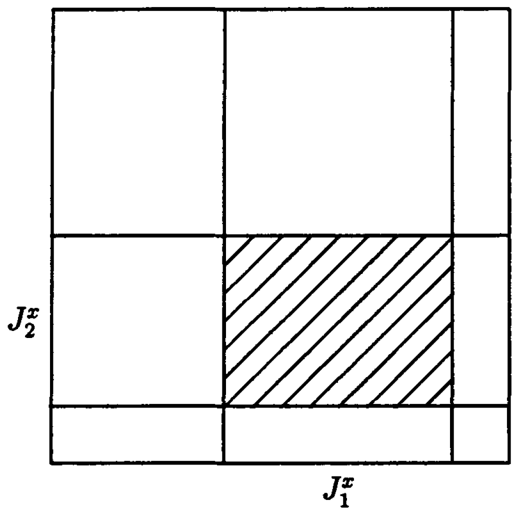
The set of normal forms is closed under a complement operation $\sim$ and an intersection operation $\triangle$. A direct union operation for normal forms cannot be obtained. However, it can be shown that $\mathcal{N}(x \cup y) = \sim (\sim \mathcal{N}(x)\triangle \sim \mathcal{N}(y))$. Therefore a union operation for normal forms is defined $\mathcal{N}(x)\underline{\vee}\mathcal{N}(y) = \sim (\sim \mathcal{N}(x)\triangle \sim \mathcal{N}(y))$.

These results may be summarized as follows. Given a multidimensional space of lexical meaning defined by some basis, the set of normal forms along with operations

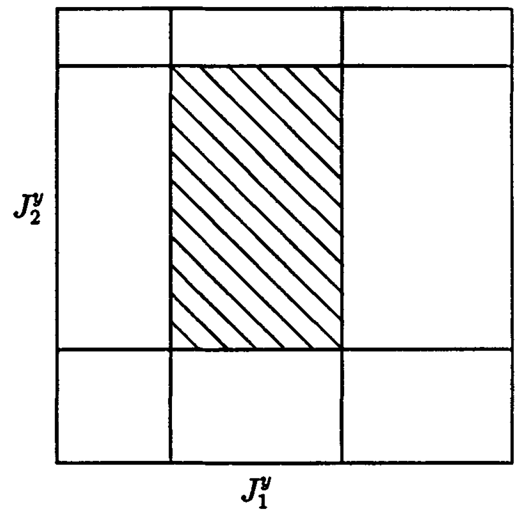
Δ, $\veebar$ and $\sim$ form a Boolean algebra.

Inclusion between normal forms can be defined: $\mathcal{N}(x) \leq \mathcal{N}(y)$ if and only if $\mathcal{N}(x) \Delta \mathcal{N}(y) = \mathcal{N}(x)$. Thus $\mathcal{N}(x) \leq \mathcal{N}(y)$ is equivalent to $x \subseteq y$.

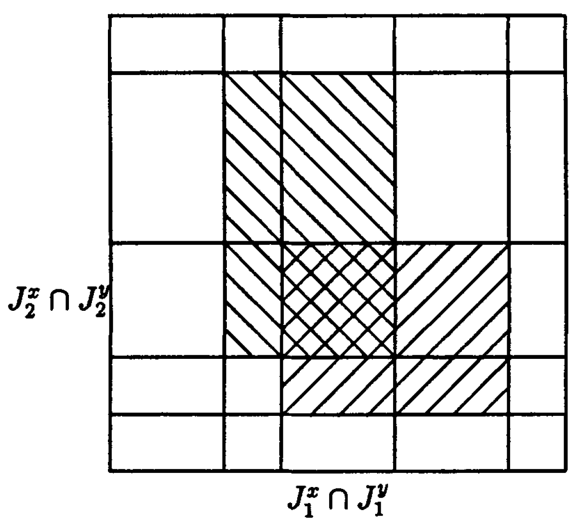
Two examples based on c-kinship will illustrate these results. (See Figure 14.) Each demonstrates computation of a union of subspaces. In both cases the resulting subspace is **immediate family**.



(a) Elementary Subset x

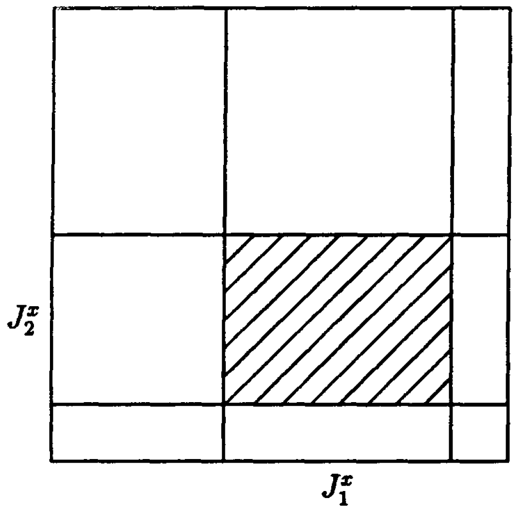


(b) Elementary Subset y

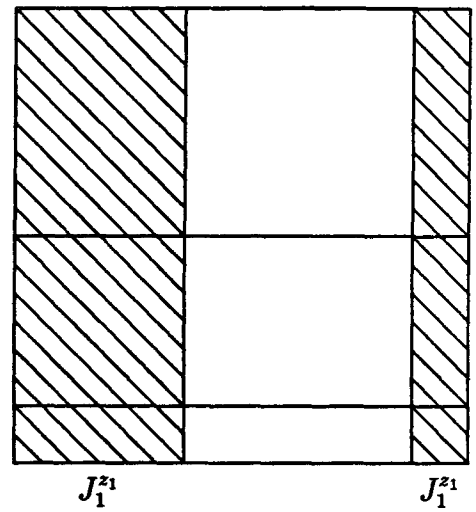


(c) Intersection of x and y

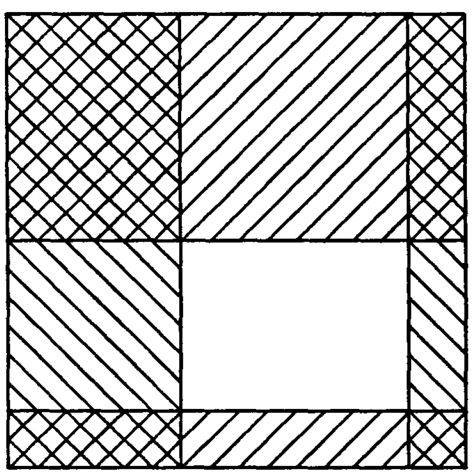
Figure 12: Example of Intersection of Elementary Subsets



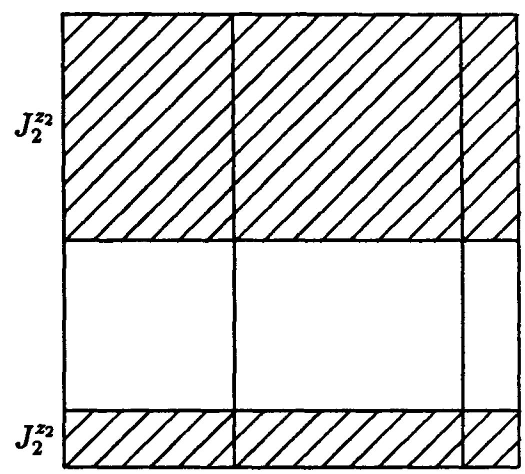
(a) Elementary Subset x



(b) $z_1 = \bigcup_{j \in (J_1 - J_1^x)} p_i^j$



(d) $-x = z_1 \cup z_2$



(c) $z_1 = \bigcup_{j \in (J_1 - J_1^x)} p_i^j$

Figure 13: Example of Complement of Elementary Subset

Example 1.

$$
\begin{aligned}
&\text{Let } \mathcal{N}(x) = \{\mathsf{L0}\} \text{ and } \mathcal{N}(y) = \{\mathsf{same} \cap \mathsf{L1}\}\\
&\text{Then } \sim \mathcal{N}(x \cup y) = \{\mathsf{L1} \cup \mathsf{L2}\}\Delta\{\mathsf{prec} \cup \mathsf{succ}, \mathsf{L0} \cup \mathsf{L2}\}\\
&\qquad\qquad\qquad\quad = \mathbf{irr}\{\mathsf{L2}, (\mathsf{prec} \cup \mathsf{succ}) \cap (\mathsf{L1} \cup \mathsf{L2})\}\\
&\text{Hence } \mathcal{N}(x \cup y) \quad = \{\mathsf{L0} \cup \mathsf{L1}\}\Delta\{\mathsf{L0}, \mathsf{same}\}\\
&\qquad\qquad\qquad\quad = \mathbf{irr}\{\mathsf{L0}, \mathsf{same} \cap (\mathsf{L0} \cup \mathsf{L1})\}\\
&\qquad\qquad\qquad\quad = \{\mathsf{L0}, (\mathsf{L0} \cup \mathsf{L1}) \cap \mathsf{same}\}
\end{aligned}
$$

The result is the set of maximal elementary subsets of the subspace immediate family.

Example 2.

$$
\begin{aligned}
&\text{Let } \mathcal{N}(x) = \{(\mathsf{prec} \cup \mathsf{same}) \cap \mathsf{L0}, \mathsf{same} \cap (\mathsf{L0} \cup \mathsf{L1})\} \text{ and } \mathcal{N}(y) = \{\mathsf{succ} \cap \mathsf{L0}\}\\
&\text{Then } \sim \mathcal{N}(x \cup y) = \{\mathsf{succ}, \mathsf{L1} \cup \mathsf{L2}\}\Delta\ \{\mathsf{prec} \cup \mathsf{succ}, \mathsf{L2}\}\Delta\ \{\mathsf{prec} \cup \mathsf{same}, \mathsf{L1} \cup \mathsf{L2}\}\\
&\qquad\qquad\qquad\quad = \mathbf{irr}\{\mathsf{L2}, (\mathsf{prec} \cup \mathsf{same}) \cap \mathsf{L2}, (\mathsf{prec} \cup \mathsf{succ}) \cap (\mathsf{L1} \cup \mathsf{L2}),\\
&\qquad\qquad\qquad\qquad\quad \mathsf{prec} \cap (\mathsf{L1} \cup \mathsf{L2}), \mathsf{succ} \cap \mathsf{L2}, \mathsf{succ} \cap (\mathsf{L1} \cup \mathsf{L2})\}\\
&\qquad\qquad\qquad\quad = \{\mathsf{L2}, (\mathsf{prec} \cup \mathsf{succ}) \cap (\mathsf{L1} \cup \mathsf{L2})\}\\
&\text{Hence } \mathcal{N}(x \cup y) \quad = \{\mathsf{L0} \cup \mathsf{L1}\}\Delta\ \{\mathsf{same}, \mathsf{L0}\}\\
&\qquad\qquad\qquad\quad = \mathbf{irr}\{\mathsf{L0}, \mathsf{same} \cap (\mathsf{L0} \cup \mathsf{L1})\}\\
&\qquad\qquad\qquad\quad = \{\mathsf{L0}, \mathsf{same} \cap (\mathsf{L0} \cup \mathsf{L1})\}
\end{aligned}
$$

Again the result is the normal form of subspace immediate family.

Figure 14: Boolean Operations on Normal Forms

4 The Lexicon

Given a set of lexical items, such as the words denoting c-kinship, distinguishing properties (i.e., diagnostic components) can be determined by linguistic analysis. These distinguishing properties can then be organized into sets that partition the entities modeling the lexical items. It is possible to select a subset of these partitions that has the property of independence. Such a set is called a basis. It structures the universe to yield a multidimensional space. Subspaces of the multidimensional space are uniquely represented by normal forms, for which a Boolean algebra can be defined. The multidimensional space so formed will be called a *semantic space*.

Linguistic analysis provides definitions of the lexical items in terms of (specifically, as Boolean functions of) the distinguishing properties. These definitions can be used to define a mapping from basic expressions of IL to the Boolean algebra of normal forms. This mapping will be called a *lexicon* for the vocabulary of lexical items.

Let the mapping be denoted v. Then the following definitions can be made. Relative to the basis that defines the semantic space, basic expressions x' and y' are synonymous if and only if $v(x') = v(y')$; x' and y' are contradictory if and only if $v(x')\Delta v(y') = 0$; x' entails y' if and only if $v(x') \leq v(y')$, that is, if and only if $v(x')\Delta v(y') = v(x')$ or equivalently, $v(x')\Delta \sim v(y') = 0$.

The mapping v can be extended to nonlogical constants of PTQ by defining $v(x) = v(x')$ if x is a nonlogical constant of PTQ and $x \Rightarrow_T x'$. The definitions of synonymy, contradiction and entailment are similarly extended.

Definition of a lexicon for c-kinship is given in Figure 15.

It is to be noted that the basis selected for construction of the semantic space of lexical meaning will determine the precision of the meanings associated with the lexical items. Therefore, meaning equivalence and meaning inclusion are understood relative to the basis. Equivalence or inclusion relative to a given basis may not hold relative to a refinement of that basis. Thus a notion of learning or development is inherent in this theory.

This approach to lexical semantics seems to have some advantages over the relational structure described in Section 2. Because unique normal forms can represent meanings, determination of synonymy, contradiction and entailment is computationally more straightforward. Even the inverse process of selecting a lexical item or an expression in lexical items, given a meaning, might be facilitated. It would also appear that this advantage will be more pronounced the larger the set of lexical items (i.e., semantic domain) becomes.

An inherent disadvantage of this approach relative to that employing logic pro-

$B = \{P_1, P_2, P_3\}$
$P_1 = \{\textsf{prec, same, succ}\}$
$P_2 = \{\textsf{L0, L1, L2}\}$
$P_3 = \{\textsf{male, female}\}$

v: **father′** $\mapsto$ prec $\cap$ L0 $\cap$ male
mother′ $\mapsto$ prec $\cap$ L0 $\cap$ female
uncle′ $\mapsto$ prec $\cap$ L1 $\cap$ male
aunt′ $\mapsto$ prec $\cap$ L1 $\cap$ female
brother′ $\mapsto$ same $\cap$ L1 $\cap$ male
sister′ $\mapsto$ same $\cap$ L1 $\cap$ female
son′ $\mapsto$ succ $\cap$ L0 $\cap$ male
daughter′ $\mapsto$ succ $\cap$ L0 $\cap$ female
nephew′ $\mapsto$ succ $\cap$ L1 $\cap$ male
niece′ $\mapsto$ succ $\cap$ L1 $\cap$ female
cousin′ $\mapsto$ L2
self′ $\mapsto$ same $\cap$ L0
parent′ $\mapsto$ prec $\cap$ L0
child′ $\mapsto$ succ $\cap$ L0
sibling′ $\mapsto$ same $\cap$ L1
immediate family′ $\mapsto$ L0 $\cup$ same $\cap$ L1

Figure 15: Lexicon for C-Kinship

grams is its limited expressiveness. While a logic program permits assertions such as $\mathsf{parent}(x,y) \leftrightarrow \mathsf{child}(y,x)$ and $\mathsf{uncle}(x,y) \rightarrow \exists z[\mathsf{brother}(x,z)]$, a semantic space cannot explicitly represent such knowledge. However, as the next definition of c-kinship demonstrates, it is sometimes possible to implicitly represent such knowledge.

Consider a set $S \subseteq H \times H$ comprising three generations of blood kin. For $i = 1, 2, 3$, define:
$\mathsf{L}i = \{(x,y) \in S | \text{the join of } x \text{ and } y \text{ in the family tree is a distance } i \text{ from } x\}$
$\mathsf{R}i = \{(x,y) \in S | \text{the join of } x \text{ and } y \text{ in the family tree is a distance } i \text{ from } y\}$

It will be assumed that S is partitioned by $P_1 = \{\mathsf{L0}, \mathsf{L1}, \mathsf{L2}\}$, $P_2 = \{\mathsf{R0}, \mathsf{R1}, \mathsf{R3}\}$ and $P_3 = \{\mathsf{male}, \mathsf{female}\}$. As a consequence, $B = \{P_1, P_2, P_3\}$ is a basis of S. The semantic space is shown in Figure 16.

This basis defines a structure that is better than the first one in several ways. First, the meanings are grouped more simply: **cousin** occupies just two atoms; **immediate family** is now an elementary subset, viz., $(\mathsf{L0} \cup \mathsf{L1}) \cap (\mathsf{R0} \cup \mathsf{R1})$. Second, $\mathsf{L}i \cap \mathsf{R}j$ is inverse to $\mathsf{L}j \cap \mathsf{R}i$. For example, $\mathsf{L1} \cap \mathsf{R2}$ is the extension of **uncle or aunt**. The inverse c-kinship relation is **nephew or niece** which has the extension $\mathsf{L2} \cap \mathsf{R1}$. Thus knowledge about inverse c-kinship relations is implicit in this semantic space. Third, $\mathsf{L}i \cap \mathsf{R}j$ where $i \neq 0 \neq j$ implies the existence of a sibling relation.

The basis defining this space and the underlying linguistic analysis seem to more fully represent the meanings of c-kinship relations. It is likely that a similar circumstance will obtain in most semantic domains. Therefore, the empirical linguistic analysis underlying construction of a lexicon seems to be a procedure requiring experience and good judgment.

		male			female	
L0	self	father	gfather	self	mother	gmother
L1	son	brother	uncle	daughter	sister	aunt
L0	grandson	nephew	cousin	gdaughter	niece	cousin
	R0	R1	R2	R0	R1	R2

Figure 16: A Second Basis for C-Kinship

5 Further Development of the Theory of Semantic Spaces

The theory of semantic spaces will be developed further with the help of two extensions of the running example. The first involves partial consanguineal relations. The second deals with affinal relations.

5.1 Extension to Partial Consanguineal Kinship

Relative to both of the bases considered thus far, half blood relationships would be synonymous with full blood relationships. For example, **half-brother** and **full-brother** would both have the extension $\mathsf{same} \cap \mathsf{L1} \cap \mathsf{male}$. The extension of c-kinship to include new lexical items denoting half blood relationships will be referred to as extended consanguineal kinship or ec-kinship. A new basis will be defined with sufficient detail to differentiate between half and full blood relationships. This will be accomplished by specifying not only the length of the path from individual x to a nearest common ancestor of individuals x and y, but also the kinds of ancestors on that path. For example,
$\mathsf{LMP} = \{(x, y) \in S |$the path from x to a nearest common ancestor of x and y contains x, the mother of x and the maternal grandfather of x; and there is no other path of length 2$\}$
$\mathsf{LMB} = \{(x, y) \in S |$the paths from x to nearest common ancestors of x and y contains x, the mother of x and both maternal grandparents of $x\}$
$\mathsf{LP} = \{(x, y) \in S |$the path from x to a nearest common ancestor of x and y contains x and the father of x; and there is no other path of length 1$\}$
$\mathsf{L} = \{(x, y) \in S |$the join of x and y is $x\}$

The complete partition is $P_1 = \{\mathsf{L,LM,LP,LB,LMM,LMP,LMB,LPM,LPP,LPB}\}$. The partition $P_2 = \{\mathsf{R,RM,RP,RB,RMM,RMP,RMB,RPM,RPP,RPB}\}$ is defined analogously for the right member y. The third partition is $P_3 = \{\mathsf{male,female}\}$. The subdivision of S produced by these partitions is shown in Figure 17.

It is apparent from the figure that these partitions are not independent and therefore do not form a basis of S. While P_1 and P_3 are independent, neither P_1 and P_2 nor P_2 and P_3 are. As a result, distinct standard forms do not represent distinct elementary subsets. For example,
$\mathsf{father} = \mathsf{L} \cap \mathsf{RP} \cap \mathsf{male}$
$= (\mathsf{L} \cup \mathsf{LM}) \cap \mathsf{RP} \cap \mathsf{male}$
$= (\mathsf{L} \cup \mathsf{LM}) \cap (\mathsf{RP} \cup \mathsf{RB}) \cap \mathsf{male}$
$= \cdots.$

male female

L	self	∅	father	∅	∅	matern gfather	∅	∅	patern gfather	∅
LM	son	half brother	∅	∅	half uncle	∅	∅	half uncle	∅	∅
LP	son	∅	half brother	∅	∅	half uncle	∅	∅	half uncle	∅
LB	∅	∅	∅	brother	∅	∅	uncle	∅	∅	uncle
LMM	gson	half nephew	∅	∅	half cousin	∅	∅	half cousin	∅	∅
LMP	gson	∅	half nephew	∅	∅	half cousin	∅	∅	half cousin	∅
LMB	∅	∅	∅	nephew	∅	∅	cousin	∅	∅	cousin
LPM	gson	half nephew	∅	∅	half cousin	∅	∅	half cousin	∅	∅
LPP	gson	∅	half nephew	∅	∅	half cousin	∅	∅	half cousin	∅
LPB	∅	∅	∅	nephew	∅	∅	cousin	∅	∅	cousin
	R	RM	RP	RB	RMM	RMP	RMB	RPM	RPP	RPB

Figure 17: Partitions of the Ec-Kinship Universe

Consequently, equality is not the same as identity.

To remedy this, a basis is formed from P_1 and P_3. $B = \{P_1, P_3\}$ will be called the *first level basis.* Next each subdivision defined by B is examined. These subdivisions are called the *atoms defined by* B. Consider the atom $a_7 = \mathsf{LB} \cap \textsf{female}$. Blocks of P_2 that have nonempty intersection with this atom are RB, RMB and RPB. Moreover, {RB,RMB,RPB} partitions a_7. $B_7 = \{\{\mathsf{RB,RMB,RPB}\}\}$ will be called a *second level basis.* Each of the atoms defined by B may have a basis. In the present example the second level bases are denoted $B_1, B_2, \ldots, B_{20}$. There are only two levels. The subdivision produced by this system of bases is shown in Figure 18.

The collection $\{B, B_1, B_2, \ldots, B_{20}\}$ will be referred to as an *extended basis* of S. An extended basis can be indexed by a tree domain. That is, the bases may be viewed as labels on the nodes of a tree whose root has the first level basis as its label.

Such an embedding of semantic spaces is typical. A simple example is the following. The domain of physical entities might be partitioned by $P_1 = \{\textsf{animal,vegetable,mineral}\}$ and again by $P_2 = \{\textsf{count,mass}\}$. Assuming that every combination is possible, $\{P_1, P_2\}$ is a first level basis of the domain, defining nine atoms: animal ∩ count, animal ∩ mass, ..., mineral ∩ mass. Each atom is itself a domain and can be partitioned by attributes appropriate to it. Hence each atom has a (in general distinct) basis. This subdivision can continue through a number of levels.

While the extended basis shown in Figure 18 does indeed yield a multidimensional space, it does not structure the subspaces neatly. For example,
half-cousin = [(LMM ∩ (male ∪ female)) ∩ (RMM ∪ RPM)] ∪ [(LMP ∩ (male ∪ female)) ∩ (RMP ∪ RPP)] ∪ [(LPM ∩ (male ∪ female)) ∩ (RMM ∪ RPM)] ∪ [(LPP ∩ (male ∪ female)) ∩ (RMP ∪ RPP)].

A similar deficiency was found in the first basis for c-kinship:
{{prec,same,succ}, {L0,L1,L2}, {male,female}}.
The alternative basis
{{L0,L1,L2}, {R0,R1,R2}, {male,female}}
yielded a neater structure. This latter basis can be taken as a first level basis and refined by second level bases to distinguish between half and full blood relationships. The resulting extended basis for ec-kinship is:
B = {{L0,L1,L2}, {R0,R1,R2}, {male,female}}
B_3 = {{RMP,RPP}}
B_5 = {{LM,LP,LB}}
B_6 = {{LM,LP,LB}, {RMX,RPX}}
B_8 = {{LMM,LMP,LMB,LPM,LPP,LPB}}
B_9 = {{LMM,LMP,LMB,LPM,LPP,LPB}, {RMX,RPX}}
where RMX = RMM ∪ RMP ∪ RMB and similarly for RPX. The modified multidimen-

	male				female			
L	R self	RP father	RMP matern gfather	RPP patern gfather	R self	RM mother	RMM matern gmother	RPM patern gmother
LM	R son	RM half brother	RMM half uncle	RPM half uncle	R daughter	RM half sister	RMM half aunt	RPM half aunt
LP	R son	RP half brother	RMP half uncle	RPP half uncle	R daughter	RP half sister	RMP half aunt	RPP half aunt
LB	RB brother	RMB uncle	RPB uncle		RB sister	RMB aunt	RPB aunt	
LMM	R gson	RM half nephew	RMM half cousin	RPM half cousin	R gdau'ter	RM half niece	RMM half cousin	RPM half cousin
LMP	R gson	RP half nephew	RMP half cousin	RPP half cousin	R gdau'ter	RP half niece	RMP half cousin	RPP half cousin
LMB	RB nephew	RMB cousin	RPB cousin		RB niece	RMB cousin	RPB cousin	
LPM	R gson	RM half nephew	RMM half cousin	RPM half cousin	R gdau'ter	RM half niece	RMM half cousin	RPM half cousin
LPP	R gson	RP half nephew	RMP half cousin	RPP half cousin	R gdau'ter	RP half niece	RMP half cousin	RPP half cousin
LPB	RB nephew	RMB cousin	RPB cousin		RB niece	RMB cousin	RPB cousin	

Figure 18: Ec-Kinship as a Multidimensional Space

sional space is shown in Figure 19.

Relative to this basis,
half-cousin $=$ (L2 $\cap$ R2 $\cap$ (male $\cup$ female)) $\cap$ (LMM $\cup$ LMP $\cup$ LPM $\cup$ LPP).

It should be pointed out that all the results stated earlier for a simple basis hold as well for an extended basis. Each subspace has a normal form. The Boolean operations (suitably extended to observe the embedded structure of the multidimensional space) and the set of normal forms yield a Boolean algebra. (See [11] for definitions and proofs.)

5.2 Extension to Affinal Kinship

Consideration has been restricted to consanguineal or blood kinship, extended to recognize half blood kinship. Affinal relationships, established by marriage of persons unrelated by blood, have not been considered. Affinal kinship will be referred to as a-kinship.

It is possible to define any affinal relationship between individuals x and y by specifying two consanguineal relationships:

1. that between x and the relative w involved in the affinal bond, and
2. that between w's marriage partner z and y.

Some examples will clarify this.
father-in-law(x, y) if parent$(x, w) \wedge$ spouse$(w, z) \wedge$ self$(z, y) \wedge$ male(x, y)
step-brother(x, y) if child$(x, w) \wedge$ spouse$(w, z) \wedge$ parent$(z, y) \wedge$ male(x, y)
wife(x, y) if self$(x, w) \wedge$ spouse$(w, z) \wedge$ self$(z, y) \wedge$ female(x, y)

Either of the definitions for c-kinship could be used to construct a basis for a-kinship. For example, the basis $B = \{P_1, P_2, P_3, P_4, P_5\}$ where $P_1 = \{$LL0,LL1,LL2$\}$, $P_2 = \{$LR0,LR1,LR2$\}$, $P_3 = \{$RL0,RL1,RL2$\}$, $P_4 = \{$RR0,RR1,RR2$\}$ and $P_5 = \{$male,female$\}$ yields the 5-dimensional space shown in Figure 20. Notice that many of the relationships represented have no English words denoting them. They are all possible relationships, however, and could be expressed by paraphrase. For example, the subspace LL1$\cap$LR1$\cap$RL1$\cap$RR1$\cap$male is the denotation of the phrase **sibling's brother in law**.

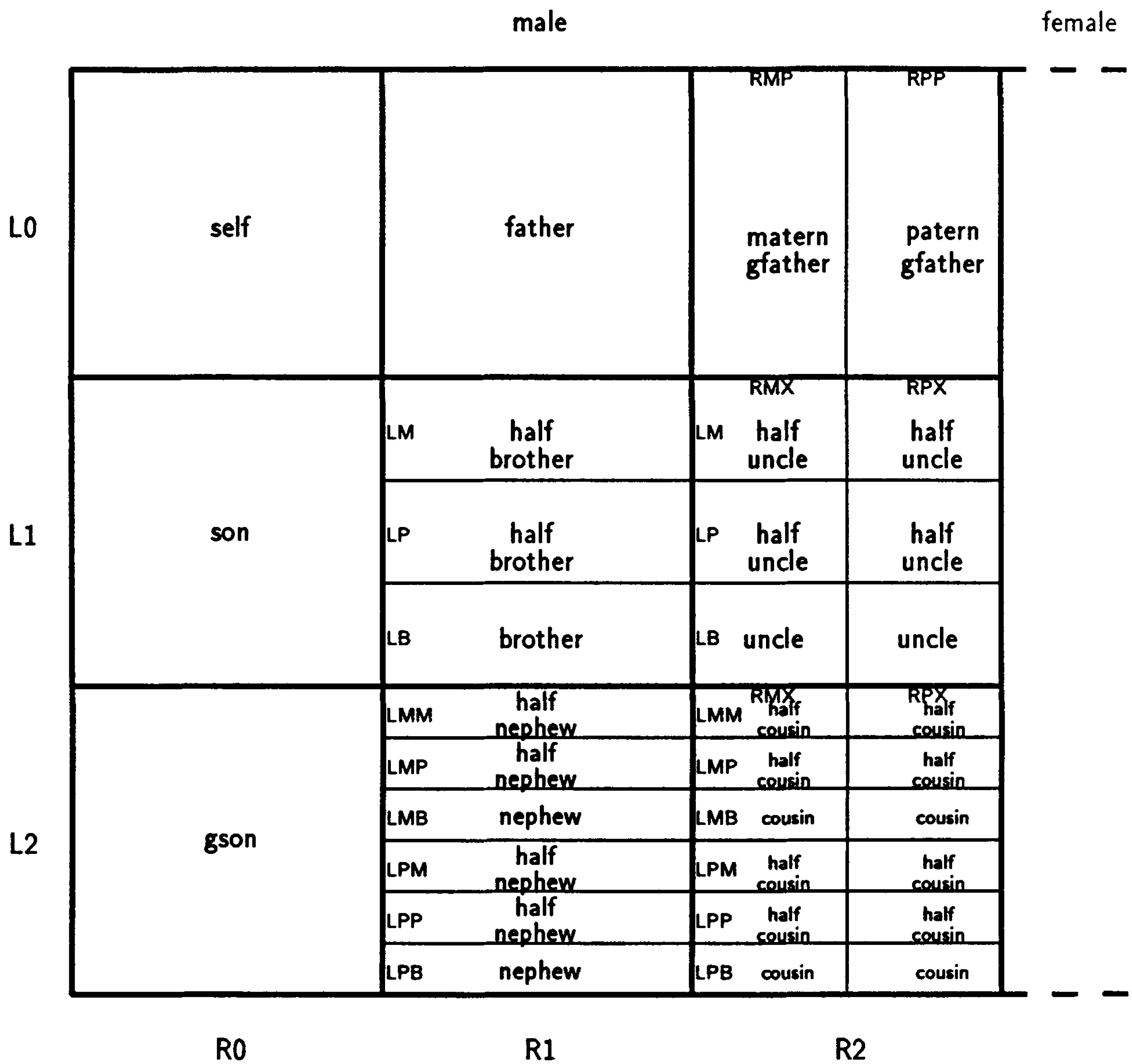


Figure 19: Another Basis for Ec-Kinship

male — female

		RL0 RR0	RL0 RR1	RL0 RR2	RL1 RR0	RL1 RR1	RL1 RR2	RL2 RR0	RL2 RR1	RL2 RR2
LL0	LR0	husband	step father	step gfather	son in law	brother in law	uncle in law	gson in law	nephew in law	cousin in law
LL0	LR1	father in law	gfather in law							
LL0	LR2	gfather in law								
LL1	LR0	step son	step brother							
LL1	LR1	brother in law	uncle in law			sib's brother in law				
LL1	LR2	uncle in law								
LL2	LR0	gson in law								
LL2	LR1	nephew in law								
LL2	LR2	cousin in law								

Figure 20: A-Kinship as a Multidimensional Space

6 The Role of the Lexicon

A semantic space will in general contain many levels of embedding. For example, a general kinship space comprising extended consanguineal and affinal relationships might combine the previously defined spaces as follows.

$B = \{P_1, P_2\}$ where $P_1 = \{\text{cons,affin}\}$ and $P_2 = \{\text{male,female}\}$
$B_1 = \{P_{11}, P_{12}\}$ where $P_{11} = \{\text{L0,L1,L2}\}$ and $P_{12} = \{\text{R0,R1,R2}\}$
$B_{13} = \{P_{131}\}$ where $P_{131} = \{\text{RMP,RPP}\}$
$B_{15} = \{P_{151}\}$ where $P_{151} = \{\text{LM,LP,LB}\}$
$B_{16} = \{P_{161}, P_{162}\}$ where $P_{161} = \{\text{LM,LP,LB}\}$ and $P_{162} = \{\text{RMX,RPX}\}$
$B_{18} = \{P_{181}\}$ where $P_{181} = \{\text{LMM,LMP,LMB,LPM,LPP,LPB}\}$
$B_{19} = \{P_{191}, P_{192}\}$ where $P_{191} = \{\text{LMM,LMP,LMB,LPM,LPP,LPB}\}$ and $P_{192} = \{\text{RMX, RPX}\}$
$B_3 = \{P_{31}, P_{32}, P_{33}, P_{34}\}$ where $P_{31} = \{\text{LL0,LL1,LL2}\}$, $P_{32} = \{\text{LR0,LR1,LR2}\}$, $P_{33} = \{\text{RL0,RL1,RL2}\}$ and $P_{34} = \{\text{RR0,RR1,RR2}\}$
B_2 is similar to B_1 and B_4 is similar to B_3. (That is, the "female" atoms are partitioned like the "male" atoms.)

The tree indexing for this extended basis for the kinship space is shown in Figure 21.

Some idea of the embedding in a complete space of lexical meaning is conveyed by the semantic domains defined by Nida for classification of lexical meaning of Koine Greek [10]. This classification may not be ideal data for construction of an extended basis. It is possible that it would yield a poorly structured space as did the initial classification of c-kinship. It does however illustrate the embedding of semantic domains starting with the most general.

The following list includes only the highest levels. For more detail one can consult [10] and further references given there.

1. Entities
   (a) Inanimate
      i. Natural
         A. Geographical
         B. Natural substances
         C. Flora and plant products
      ii. Manufactured or constructed
         A. Artifacts
         B. Processed substances

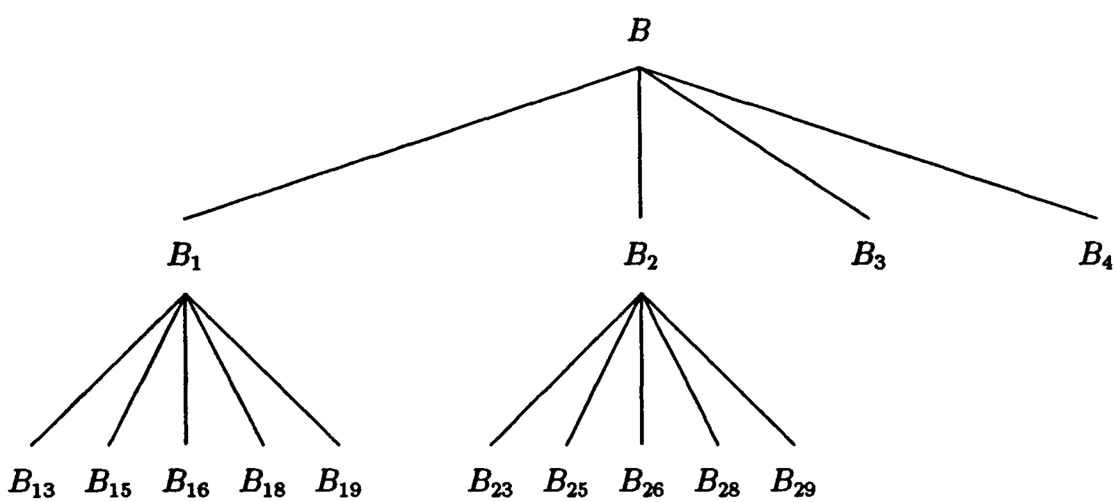


Figure 21: Tree Indexing of the Extended Basis for Kinship

C. Constructions

(b) Animate

i. Animals, birds, insects

ii. Humans

A. Generic and distinctions by age and sex

B. Kinship

C. Groups

D. Body, body parts and body products

iii. Supernatural powers or beings

2. Events

3. Abstracts

4. Relationals

Of course even a space of lexical meaning of the scope suggested by this classification will not be able to express certain kinds of knowledge about lexical entities. Much of this additional knowledge might be called "encyclopedic" information. But if the basis is adequate to distinguish the meanings denoted by the lexical items in the vocabulary, then the unique representations for these meanings can provide links to further (encyclopedic) knowledge which could be represented in, say, a logic program.

It seems reasonable to assume that the lexicon, as defined here, will be only one component of the total knowledge of a natural language faculty. An encyclopedic knowledge base and a facility to accept and process contextual information will also be necessary components.

To see the role of the lexicon, consider again the sentences **Mary loves every man such that he loves her** and **An actor loves Mary**. Their translations in IL are $\forall x[\mathbf{man}'(x) \wedge \mathbf{love}'(x,m) \rightarrow \mathbf{love}'(m,x)]$ and $\exists x[\mathbf{actor}'(x) \wedge \mathbf{love}'(x,m)]$.

For this discussion the denotation mapping ψ is assumed to be extended in the usual way to a homomorphism of arbitrary expressions of IL. If x and y are IL expressions, $x \leq_{IL} y$ is defined to be equivalent to $\psi(x) \subseteq \psi(y)$. Since for basic expressions of IL $x \leq_{IL} y$ is equivalent to $v(x) \leq v(y)$, and in all usage the domain will be clear, $x \leq_{IL} y$ will be written simply $x \leq y$. Similarly if PTQ expressions x and y have translations x' and y' and $\psi(x') \subseteq \psi(y')$, then $x \leq_{PTQ} y$. Again since no confusion can result, $x \leq_{PTQ} y$ will be written simply $x \leq y$.

Assume a lexicon v such that $v(\mathbf{actor}') \leq v(\mathbf{man}')$. In general, it follows from the definition of $\leq$ that if $x \leq y$ then (1) x and y are functors[9] of the same type,

[9]For conciseness, individuals and formulas will be considered functors of zero arity.

and (2) for all $z \in \mathbf{dom}(x)$ $(= \mathbf{dom}(y))$, $x(z) \leq y(z)$. For example, **actor′** and **man′** are functors that take individuals as argument and yield a formula. Therefore **actor′** $\leq$ **man′** implies that for every individual z, $\mathbf{actor'}(z) \leq \mathbf{man'}(z)$. That is, $\mathbf{actor'}(z)$ entails $\mathbf{man'}(z)$ or $\forall z[\mathbf{actor'}(z) \rightarrow \mathbf{man'}(z)]$.

From $\forall x[\mathbf{actor'}(x) \rightarrow \mathbf{man'}(x)]$, $\exists x[\mathbf{actor'}(x) \wedge \mathbf{love'}(x, m)]$ and $\forall x[\mathbf{man'}(x) \wedge \mathbf{love'}(x, m) \rightarrow \mathbf{love'}(m, x)]$, the desired result $\exists x[\mathbf{actor'}(x) \wedge \mathbf{love'}(m, x)]$ can be deduced in IL using the axioms of first order logic. (See Appendix B for details.)

Similarly, assuming **adore′** $\leq$ **love′**, **Bill adores Mary** can be shown to entail **Bill loves Mary**. Thence the translation of **Mary loves Bill** can be deduced using the axioms of first order logic.

The above illustrates deduction when x and y are in the role of functors and $x \leq y$. Now consider x and y in the role of arguments. Let $x \leq y$ and let w be a functor such that $x, y \in \mathbf{dom}(w)$. Then w is said to be *isotone* if and only if $w(x) \leq w(y)$ for all such x and y; w is said to be *antitone* if and only if $w(y) \leq w(x)$ for all such x and y. For example, **a** translates to an isotone functor. By contrast, **every** translates to an antitone functor.[10] **a man** translates to an isotone functor and in fact, it can be shown that all PTQ terms translate to isotone functors (see [11]).

The role of the isotone/antitone property in deduction can be simply illustrated. From the corresponding lexicon entries, it would follow that **actor** entails **man**. Given that the translation of **a** is isotone, it follows immediately that **an actor** entails **a man**. Again, given that the translations of **Mary** and **love** are isotone, it is deduced that **Mary loves an actor** entails **Mary loves a man**. Similar logic yields the result that **Mary loves every man** entails **Mary loves every actor**.

It would be important then to make the partition {**isotone,antitone,neither**} a part of a basis of any space of functors.

[10] For a discussion of English determiners see [9]

7 Conclusion

The completely formalized nature of Montague's language theory makes it particularly applicable to the problem of natural language understanding by machine. However, deduction in a system based on Montague's theory is severely limited by the absence of lexical semantics. A relational structure, for example a logic program, can be used to specify lexical semantics. However it is computationally unwieldy. An alternative (or, more precisely a partial alternative) called a *semantic space* is described in this paper. A lexicon is defined to be a map from a vocabulary of lexical items to a semantic space.

Unique representations called *normal forms* are defined for the subspaces of a semantic space. A Boolean algebra of normal forms is then defined. Since subspaces of a semantic space correspond to lexical meanings, inclusion between subspaces corresponds to meaning inclusion or entailment.

The lexicon that results from this approach can provide a lexical extension to Montague semantics based on the Intensional Logic. Further it seems that this lexical semantics is compatible with other semantic theories similar to that of Montague.

It is conjectured that the conceptual simplicity of such a lexicon will result in computational simplicity. Further, and perhaps more important, the independence of the dimensions of a semantic space permits a high degree of computational parallelism.

References

[1] Brachman, Ronald J.: 1979, "On the Epistemological Status of Semantic Networks," in N. V. Findler (ed.), *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press.

[2] Deliyanni, Amaryllis and Robert A. Kowalski: 1979, "Logic and Semantic Networks," *Communications of the ACM* 22, 184-192.

[3] Dowty, David R.: 1978, "Applying Montague's Views on Linguistic Metatheory to the Structure of the Lexicon," in Donka Farkas et al. (eds.), *Parasession on the Lexicon*, Chicago Linguistic Society.

[4] Dowty, David R.: 1979, *Word Meaning and Montague Grammar*, D. Reidel Publishing Company.

[5] Dowty, David R., Robert E. Wall and Stanley Peters: 1981, *Introduction to Montague Semantics*, D. Reidel Publishing Company.

[6] Gallin, Daniel: 1975, *Intensional and Higher-Order Modal Logic with Applications to Montague Semantics*, North-Holland.

[7] Hausser, Roland R.: 1984, *Surface Compositional Grammar*, Wilhelm Fink Verlag.

[8] Keenan, Edward L. and Leonard M. Faltz: 1985, *Boolean Semantics for Natural Language*, D. Reidel Publishing Company.

[9] Keenan, Edward L. and Jonathan Stavi: 1986, "A Semantic Characterization of Natural Language Determiners," *Linguistics and Philosophy* 6, 447-478.

[10] Nida, Eugene A.: 1975, *Componential Analysis of Meaning*, Mouton.

[11] Purdy, William C.: 1987, *A Theory of Lexical Semantics*, Report CIS-87-7, School of Computer and Information Science, Syracuse University.

[12] Thomason, Richard H. (ed.): 1974, *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press.

A Details of the Example of Section 1.2

Montague provides an IL meaning postulate which asserts that, in an extensional context,

love$'$ $(m, \lambda P[P(x_1)])$,	i.e., Mary loves an individual of whom a set of predicates is true,
if and only if $\lambda P[P(x_1)]$ $(\lambda z$ [**love** $'(m,z)])$,	i.e., one of the predicates true of this individual is being loved by Mary.

By λ-reduction, such an expression can often be simplified:

$$\lambda z\ [\mathbf{love}'\ (m,z)]\,(x_1) = \ \mathbf{love}'\ (m,x_1).$$

More generally, for predicate Q,

$$Q(y, \lambda P[P(x)]) \leftrightarrow (\lambda P[P(x)])(\lambda z[Q(y,z)]).$$

This will be referred to in the following as "MP1." It corresponds to Dowty's MP1 with the intensional operators eliminated.

The structural analysis of **Mary loves every man such that he loves her** is given in Figure 22. The translation is as follows. The numbers in parentheses refer to translation rules and correspond to the numbers of the formation rules shown in Figure 22.

love him$_1 \Rightarrow_T$ **love**$'$ $(\lambda P[P(x_1)])$ **(5)**

he$_0$ **loves him**$_1 \Rightarrow_T \lambda Q[Q(x_0)](\mathbf{love}'\ (\lambda P[P(x_1)])) =$ **(4)**
$\mathbf{love}'\ (\lambda P[P(x_1)])(x_0) =$
$\mathbf{love}'\ (x_0, \lambda P[P(x_1)]) =$
$\lambda P[P(x_1)](\lambda y[\ \mathbf{love}'\ (x_0,y)]) =$ **(MP1)**
$\lambda y[\mathbf{love}'\ (x_0,y)](x_1) =$
$\mathbf{love}'\ (x_0,x_1)$

man such that he loves him$_1 \Rightarrow_T$ $\lambda x_0\ [\mathbf{man}'\ (x_0) \wedge \mathbf{love}'\ (x_0,x_1)]$ **(3,0)**

every man such that he loves him$_1 \Rightarrow_T$

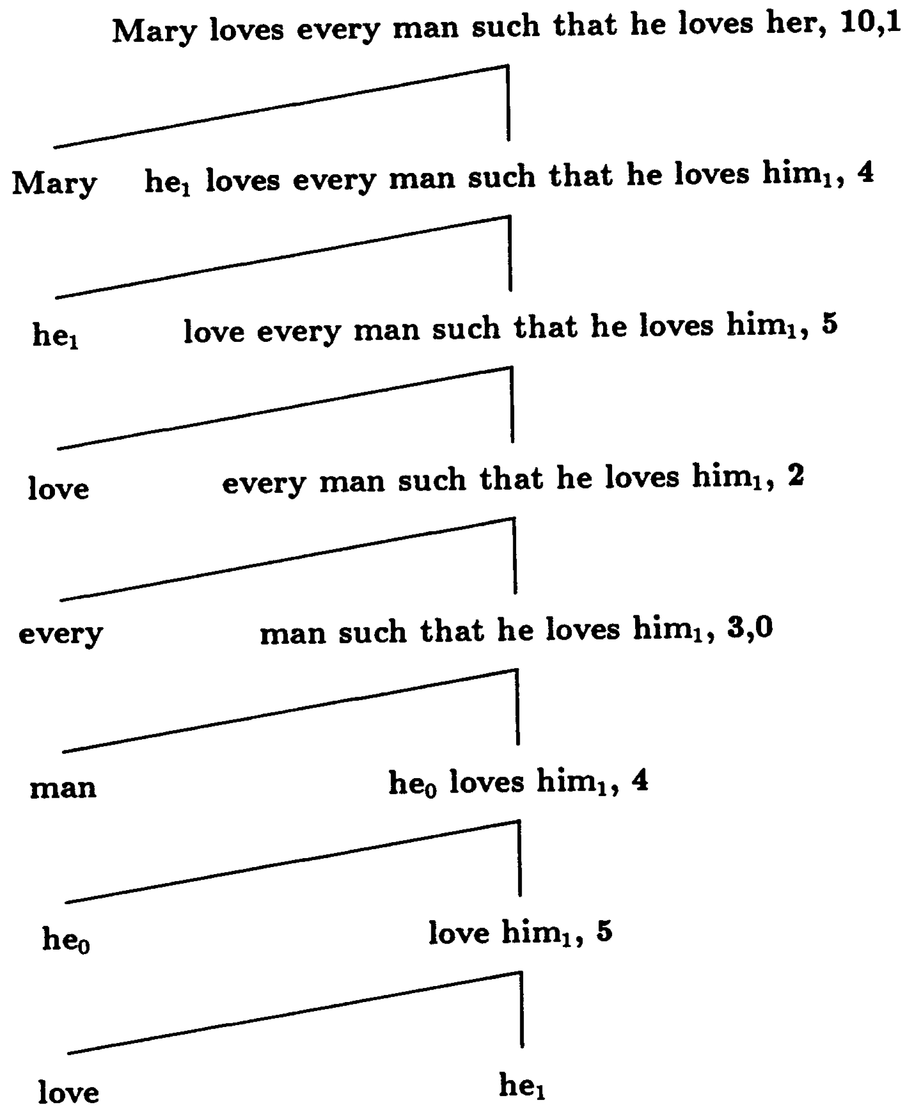


Figure 22: Structure of Mary loves every man such that he loves her

$$\lambda P \lambda Q \forall y[P(y) \rightarrow Q(y)](\lambda x_0 \text{ [man}' (x_0) \wedge \text{ love}' (x_0, x_1)]) =$$
$$\lambda Q \forall y[\lambda x_0[\text{man}' (x_0) \wedge \text{ love}' (x_0, x_1)](y) \rightarrow Q(y)] = \qquad (2)$$
$$\lambda Q \forall y \text{ [man}'(y) \wedge \text{ love}' (y, x_1) \rightarrow Q(y)]$$

love every man such that he loves him$_1$ $\Rightarrow_T$

$$\text{love}' (\lambda Q \forall y \text{ [man}'(y) \wedge \text{love}' (y, x_1) \rightarrow Q(y)]) \qquad (5)$$

he$_1$ loves every man such that he loves him$_1$ $\Rightarrow_T$

$$\lambda P \text{ } [P(x_1)](\text{love}' (\lambda Q \forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow Q(y)])) = \qquad (4)$$
$$\text{love}' (\lambda Q \forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow Q(y)])(x_1) =$$
$$\text{love}' (x_1, \lambda Q \forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow Q(y)]) =$$
$$\lambda Q \forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow Q(y)](\lambda z \text{ [love}' (x_1, z)]) = \qquad (\text{MP1})$$
$$\forall y \text{ [man}' (y) \wedge \text{love}'(y, x_1) \rightarrow \lambda z \text{ [love}' (x_1, z)](y)] =$$
$$\forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow \text{love}' (x_1, y)]$$

Mary loves every man such that he loves her$\Rightarrow_T$

$$\lambda P \text{ } [P(m)](\lambda x_1[\forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow \text{love}' (x_1, y)]]) = \qquad (10,1)$$
$$\lambda x_1 \text{ } [\forall y \text{ [man}' (y) \wedge \text{love}' (y, x_1) \rightarrow \text{love}' (x_1, y)]](m) =$$
$$\forall y \text{ [man}' (y) \wedge \text{love}' (y, m) \rightarrow \text{love}' (m, y)]$$

The structural analysis of **John is a man** is given in Figure 23. The translation follows.

a man$_1$ $\Rightarrow_T \lambda P \lambda Q \exists x[P(x) \wedge Q(x)](\text{man}') = \qquad (2)$

$$\lambda Q \exists x \text{ [man}'(x) \wedge Q(x)]$$

be a man $\Rightarrow_T \lambda \mathcal{P} \lambda z[\mathcal{P}(\lambda w[z = w])](\lambda Q \exists x[\text{man}'(x) \wedge Q(x)]) = \qquad (5)$

$$\lambda z \text{ } [\lambda Q \exists x[\text{man}'(x) \wedge Q(x)](\lambda w[z = w])] =$$
$$\lambda z \exists x \text{ [man}' (x) \wedge \lambda w[z = w](x)] =$$
$$\lambda z \exists x \text{ [man}' (x) \wedge z = x]$$

John is a man$\Rightarrow_T \lambda P[P(j)](\lambda z \exists x[\text{man}'(x) \wedge z = x]) = \qquad (4)$

$$\lambda z \exists x[\text{man}' (x) \wedge z = x](j) =$$
$$\exists x[\text{man}'(x) \wedge j = x]$$

Finally, the structural analysis of **John loves Mary** is given in Figure 24 and its translation is given below.

love Mary $\Rightarrow_T \text{love}'(\lambda P[P(m)]) \qquad (4)$

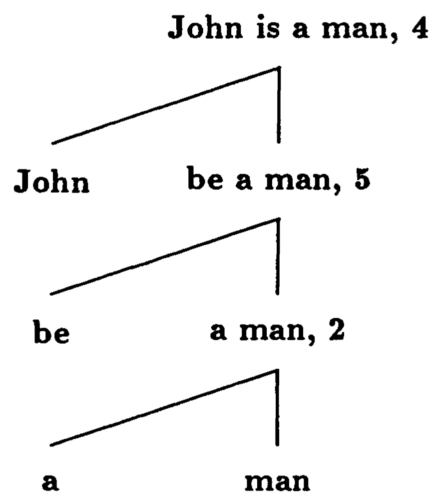


Figure 23: Structure of **John is a man**

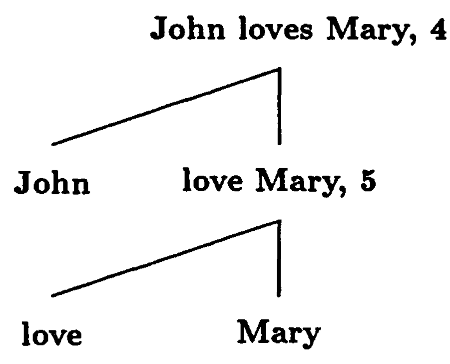


Figure 24: Structure of **John loves Mary**

$$\textbf{John loves Mary} \Rightarrow_T \lambda Q[Q(j)]\ (\textbf{love}'(\lambda P[P(m)])) = \tag{5}$$

$$\textbf{love}'(\lambda P[P(m)])(j) =$$

$$\textbf{love}'(j, \lambda P[P(m)]) =$$

$$\lambda P[P(m)](\lambda y[\textbf{love}'(j,y)]) = \tag{MP1}$$

$$\lambda y[\textbf{love}'(j,y)](m) =$$

$$\textbf{love}'(j,m)$$

Similarly, **Mary loves John** $\Rightarrow_T$ **love**$'(m,j)$.

This results in the IL expressions:

(1) $\forall y[\textbf{man}'(y) \wedge \textbf{love}'(y,m) \rightarrow \textbf{love}'(m,y)]$

(2) $\exists x[\textbf{man}'(x) \wedge j = x]$

(3) **love**$'$ (j,m)

(4) From (2) one can deduce **man**$'(j)$

(4) From (1), (3), and (4) one can deduce

$$\textbf{man}'(j) \wedge \textbf{love}'(j,m) \rightarrow \textbf{love}'(m,j)$$

$$\textbf{man}'(j)$$

$$\textbf{love}'(j,m)$$

hence **love**$'(m,j)$

i.e., **Mary loves John**

In clausal form:

(1) -**man**$'(x)$, -**love**$'(x,m)$, **love**$'(m,x)$

(1) **man**$'(a)$

(3) $j = a$

(4) **love**$'(j,m)$

(5) **man**$'(j)$ (substitution)

(6) **love**$'(m,x)$ (unification and resolution)

B Details of the Example of Section 6

Given: $\forall x$ [**actor**$'(x) \rightarrow$ **man**$'(x)$]
$\exists x$ [**actor**$'(x) \wedge$ **love**$'(x, m)$]
$\forall x$ [**man**$'(x) \wedge$ **love**$'(x, m) \rightarrow$ **love**$'(m, x)$]

Clausal form:
(1) -**actor**$'(x)$, **man**$'(x)$
(2) **actor**$'(a)$
(3) **love**$'(a, m)$
(4) -**man**$'(x)$, -**love**$'(x, m)$, **love**$'(m, x)$
(5) -**actor**$'(x)$, -**love**$'(x, m)$, **love**$'(m, x)$ (resolution)
(6) **love**$'(m, a)$ (unification and resolution)

By existential generalization on a:

$\exists x$ [**actor**$'(x) \wedge$ **love**$'(m, x)$]