Research Report AI–1994–04

An Implementation of Plurality in Discourse Representation Theory

Laurel Kathleen Graham

Artificial Intelligence Center
The University of Georgia
Athens, Georgia 30602–7415 U.S.A.

Abstract

Covington, Nute, Schmitz, and Goodman (1988) wrote a program which takes natural language sentences as input, represents them using Discourse Representation Structures, then translates the structures into Prolog clauses. The clauses may be asserted into the database or used to query it.

My thesis, PluralDRT, extends this program to handle plurality. PluralDRT is meant to be a first attempt at implementing collective and distributive readings of plural sentences. Although, PluralDRT focuses on these types of readings, it also deals with plural anaphora, generic statements, and definite descriptions acting as anaphors.

Acknowledgments

I would first like to thank Dr. Michael Covington for his time, effort, reading, and re-reading of my thesis. I also appreciate Dr. Donald Nute and Dr. Krys Kochut for being on my committee. I would next like to thank Andrew S. Lech for proof reading my thesis far and beyond the call of duty, and Daniel Brown for his ideas on DRT and the endless supply of late-night wit. I would also like to thank my family for their continual support during this rather trying time. Finally, I would like to thank Alisa Elmore, who made all of this possible by telling me there was a way to combine linguistics and computer science.

Contents

Chapter 1

Introduction

Plural sentences can have several different types of readings. Consider the following sentences:

(1.1) (a) *The lawyers jointly hired a secretary.*

(b) *The lawyers each filed a brief.*

In example (1.1a) the lawyers are acting jointly—as a collective—to hire the secretary, while in example (1.1b) the lawyers are acting separately in filing their briefs. We call these the Collective and Distributive readings of *the lawyers* respectively. Another type of reading is seen in the sentence

(1.2) *Five lawyers hired three secretaries.*

where a total of five lawyers hired a total of three secretaries, but some of the lawyers may have gotten together and hired one secretary, while the rest each hired their own secretaries (the Cumulative or Conjunctive reading).

Singular sentences do not have these different readings. This is because singular terms denote only individuals, rather than groups or collections of these individuals as plurals do. For example,

(1.3) *The lawyer hired the secretary*

can only mean that one individual who is a lawyer hired another individual who is a secretary.

An ambiguity arises when it is not clear which reading a plural has. An example is:

(1.4) *The lawyers sang.*

It is not clear whether the lawyers sang together, they sang separately, or a group of the lawyers sang together, while the rest sang alone. A singular sentence of the same form, such as *The lawyer sang*, lacks this inherent ambiguity because it is clear that only the one individual sang.

Plural terms also seem to take on different roles in the various readings. Consider the sentences from example (1.1). The plural terms in these sentences act oppositely to one another even though they are the same term. *The lawyers* in example (1.1a) is a sort of plural individual where all the lawyers act as a collective, whereas *the lawyers* in example (1.1b) refers to some individuals who act independently of one another.

In sentences with distributive readings, singular terms may refer to several individuals. This causes difficulties with anaphora. Consider the three sentences

(1.5) (a) *A man sang a song.*

(b) *The men sang a song together.*

(c) *The men each sang a different song.*

Suppose a second sentence *It was beautiful* is added to the three sentences. There would not be any conflicts with the first two sentences. *It* would refer to the song sung by the man in (1.5a) and the song the men sang together in (1.5b). However, in (1.5c), *it* does not seem to refer to anything since there were multiple songs sung. Using *they* instead of *it* also seems problematic. *They* refers to some abstract group of songs not explicitly mentioned, rather than referring to each individual song. Therefore, there seems to be no direct way of referring separately to these individuals, such as referring to the song each of the men sang.

Due to these ambiguities and complexities, representing plural sentences is a difficult task. However, a formal representation of plurals is useful in further understanding them and necessary to implement them into a computer program, which is essential for a program which adequately uses natural language.

Discourse Representation Theory (DRT) is a suitable representation system for implementing plurals on the computer because it both handles the problems associated with plurality and makes for a smoother transition to computer implementation. DRT provides a representation for each reading of a sentence, using a representation scheme which allows for plurals to act as individuals or as parts of a collection, which is similar to their function in natural language, and by using precise and simple structures which may be combined in a building block fashion to form intricate representations. Furthermore, DRT is an appropriate intermediary language for the translation of natural language to Prolog structures. This is because DRT is a logic-based system which can be easily implemented in a logic-based computer language such as Prolog.

With this in mind, I created PluralDRT, an extension of an existing implementation of DRT (Covington, Nute, Schmitz, Goodman 1988). PluralDRT translates plural sentences into Prolog structures, which can then be asserted into the database or used to query existing information in the database. PluralDRT operates under Quintus Prolog and relies on GULP (Covington 1987, 1993), an extension of Prolog for unification–based grammar.

In the remainder of this thesis, I shall discuss theories of plurality, plurality in DRT, and PluralDRT. An elementary knowledge of Prolog, feature structures, and unification–based grammar is assumed.

Chapter 2

Discourse Representation Theory (DRT)

Kamp (1981) introduced Discourse Representation Theory (DRT) in an attempt to fill what he saw as a gap in semantic theory. This gap formed as a result of two different views about meaning. The first view, found in formal semantics, sees meaning as simply specifying truth conditions, while the second, predominant in psycholinguistics, sees meaning as what the language user comprehends when he understands the oral or written words of his language. DRT tries to link these two views by combining a formal description of truth conditions with semantic representations that are similar to the language user's mental representations.

2.1 Simple DRSs

The semantic representations used in DRT are called Discourse Representation Structures (DRSs). Each DRS consists of an ordered pair $<$**U,Con**$>$ where **U** is the Universe of Discourse, a set of discourse entities, and **Con** is a set of conditions that describe properties or relations which the referents must satisfy. For example, the sentence

(2.1) *Chester barks.*

is represented as:

$$\begin{array}{|l|}\hline X \\ \hline \mathit{chester}(X) \\ \mathit{barks}(X) \\ \\ \hline \end{array}$$

where X is the discourse referent for the entity which must satisfy the condition of being (named) *Chester*[1] and *barking*.

A DRS can represent a discourse that consists of more than one sentence. It is constructed by adding the semantic information one sentence at a time. This incremental building preserves semantic cohesion in natural language. That is, DRSs maintain the logical flow of meaning from one sentence to the next. This can be seen in the following two sentences:

(2.2) *Pedro owns a cat. He feeds it.*

where the pronouns *he* and *it* depend upon information in the previous sentence. The DRS for these sentences look like:

$$\begin{array}{|l|}\hline X\ Y\ U\ V \\ \hline \mathit{pedro}(X) \\ \mathit{cat}(Y) \\ \mathit{owns}(X,Y) \\ U = X \\ V = Y \\ \mathit{feeds}(U,V) \\ \\ \hline \end{array}$$

The referents are passed from one sentence to the next through the equations $U = X$ and $V = Y$.

[1]Various versions of DRT use different representations for proper names. I am using *chester(X)* to represent the condition that X has the name *Chester* rather than simply introducing some constant "Chester" because there is possibly more than one individual with the name *Chester*. Because Kamp and Reyle use chester(X), I shall continue to do so in DRSs. However, PluralDRT uses `named(X,'Chester')`, which seems to better express the idea that proper names refer directly to the individual rather than being a condition of that individual.

2.1.1 Representing Simple Sentences

Simple sentences such as

(2.3) *Chester chases a cat.*

are added to a DRS as follows:

1. Introduce two different discourse markers for the entities *chester* and *a cat.*
   U = {X,Y}

2. Introduce the conditions which the discourse referents must satisfy.
   Con = {chester(X),cat(Y),chases(X,Y)}

The DRS now looks like this:

X Y
chester(X) *cat(Y)* *chases(X, Y)*

The referent of a proper name or singular term refers to some existing individual—or set of individuals—which satisfies the conditions associated with the referent. In terms of classical logic, there is an implicit existential quantifier associated with referents of proper names or singular terms. In example (2.3), a referent, *X*, is assigned to refer to the individual who satisfies the condition of being (named) *Chester*, and another referent, *Y*, is assigned to refer to the individual who is a *cat*. If these individuals do not exist, then the referent cannot refer to anything, and the DRS cannot be true.

Intuitively, a DRS is true if each referent in its Universe of Discourse can be mapped onto an individual in the real world such that all of the DRS-conditions are satisfied. Consider the sentence:

(2.4) *Heather owns a cat.*

This sentence is represented by the following DRS:

X Y
heather(X) *cat(Y)* *owns(X,Y)*

If some individuals, A and B, exist such that A has the name *Heather*, B is a *cat*, and A and B have the property that A owns B, then the above DRS is true.

Formally, a DRS is true in a model if it can be properly embedded in a model. A model is a domain, or a universe of things which can be talked about, together with an interpretation function which maps things from the language to the domain. Thus, a DRS is properly embedded if it maps its Universe of Discourse to members of the model such that all conditions associated with the referents are satisfied. So in our above example, the domain include the individuals A and B, and the embedding should preserve the properties of the individuals, namely A being named *Heather*, B being a *cat*, and A *owning* B.

2.1.2 Pronouns

DRT analyzes pronouns in relation to other existing discourse referents rather than in relation to existing noun phrases. Thus when resolving anaphora, DRT will look in the Universe of Discourse for a suitable antecedent referent. While linguistics generally sees pronouns as pointers to noun phrases, DRT sees pronouns as representing referents which refer to (or equal) other referents.

Anaphoric pronouns are incorporated into DRSs in the following manner:

1. Introduce a discourse referent for the anaphoric pronoun.
2. Locate the referent of the antecedent of the anaphor.
3. Introduce the condition that the pronoun's discourse referent equals its antecedent's referent.

For example, the sentences

(2.5) *Heather owns a cat. She loves it.*

is represented as follows:

$X\ Y\ Z\ W$
$heather(X)$
$cat(Y)$
$owns(X,Y)$
$Z = X$
$W = Y$
$loves(Z,W)$

The pronoun referents, Z and W, map onto the same elements as the referents of their antecedents, X and Y respectively.

In analyzing English, DRT has gender and number conditions for each referent introduced into the DRS to aid in anaphora resolution. The referent of the anaphor must satisfy these conditions as well as other conditions in the DRS. In the above example (2.4), if number and gender conditions were added, the DRS would look like this:

$X\ Y\ Z\ W$
$heather(X)$
$gender(X,f)$
$number(X,sg)$
$cat(Y)$
$gender(Y,n)$
$number(Y,sg)$
$Z = X$
$W = Y$
$loves(Z,W)$

However, in this thesis, for brevity, gender and number information are usually assumed and not explicitly displayed in DRSs.

A pronoun may sometimes have several possible antecedents. For example, in the sentence

(2.6) *The boys give the girls some flowers. They love them.*

the meaning of the second sentence is ambiguous as to whether it refers to the boys loving the girls, the girls loving the flowers, or even the flowers loving the boys or the girls. Further, deictic pronouns are not representable in the current version of DRT since they require extra-linguistic information such as a nod of the head or pointing of the finger. Even anaphoric pronouns which language users understand intuitively may be ambiguous, such as

(2.7) *Heather gives Sage a gift. She is happy.*

Intuitively we would think *she* refers to *Sage* since we know that most people are happy upon receiving a gift. However, if the context *Heather is generally happiest when she gives people gifts* is given, then *she* refers to *Heather*.

DRT makes the assumption that an antecedent can be found for every pronoun. Its concern lies with the conditions that the pronoun and its antecedent must satisfy rather than the mechanism for finding the antecedent.

2.2 Complex DRSs

2.2.1 Conditionals and Universally Quantified Sentences

DRSs which represent conditionals contain sub-DRSs. For example,

(2.8) *If a girl owns a cat, then she loves it.*

is represented as:

$$\begin{array}{|c|}\hline \quad \\ \hline \begin{array}{|l|}\hline X\ Y \\ \hline girl(X) \\ cat(Y) \\ owns(X,Y) \\ \hline\end{array} \Rightarrow \begin{array}{|l|}\hline U\ V \\ \hline U = X \\ V = Y \\ loves(U,V) \\ \hline\end{array} \\ \hline \end{array}$$

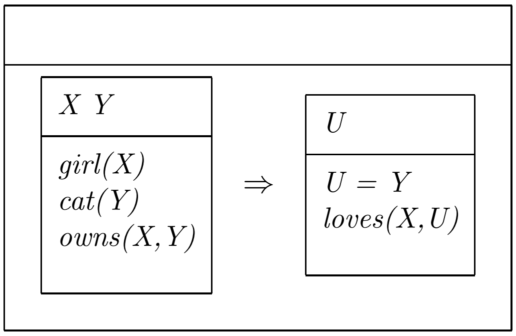
where the antecedent and consequent of the substructure match those of the conditional in natural language. In terms of DRT, the conditional structure is one condition contained in a larger DRS. It contains two complete DRSs within it.

In terms of truth conditions, a conditional structure is a hypothetical situation. While a simple DRS is considered true if each referent refers to *some* individual in the domain such that all the DRS-conditions are satisfied, the conditional structure is true if *every* mapping which satisfies the antecedent can be extended to satisfy the consequent. Thus in example 2.7, for all individuals A and B such that A is named *Heather*, B is a *cat*, and A *owns* B, it is also true that A *loves* B.

Universally quantified sentences are treated as conditionals also. The sentence *Every girl who owns a cat loves it* is treated as *If a girl owns a cat, then she loves it.* DRT does this because of the antecedent's implicit universal quantifier, and the consequent's implicit existential quantifier, which is contingent on the antecedent. So,

(2.9) *Every girl who owns a cat loves it*

is correctly represented by



All X and Y which satisfy the conditions *girl(X), cat(Y)* and *owns(X, Y)* must also satisfy the condition X *loves* Y. It is important to note that *cat* has no one real-world referent; rather *cat(Y)* refers to potentially different cats for each cat-owning girl. Being able to handle sentences like this was one of the early successes of DRT.

2.2.2 Negation

DRSs that represent negated sentences also contain a sub-DRS, but the sub-structure is headed by a negation symbol. The sentence

(2.10) *A girl does not own a cat.*

can be represented as

$$\begin{array}{|l|}\hline X \\ \hline \neg \; \begin{array}{|l|}\hline X \;\; Y \\ \hline \mathit{girl}(X) \\ \mathit{cat}(Y) \\ \mathit{owns}(X,Y) \\ \\ \hline\end{array} \\ \\ \hline\end{array}$$

This DRS expresses the meaning that *It is not the case that there exists a girl who owns a cat.* For it to be true, there must be no individuals, A and B, in the domain such that A is a *girl*, B is a *cat*, and A *owns* B. Formally, the condition $\neg K$, where K is an arbitrary DRS, is satisfied if there is no embedding that makes K true.

Because some negated sentences are ambiguous, it is possible to have more than one DRS for these sentences. Example 2.10 has two possible meanings associated with it:

1. *It is not the case that there exists some girl who owns a cat.*

2. *It is the case that there exists some girl who does not own a cat*

The second meaning corresponds to the DRS:

$$\begin{array}{|l|}\hline X \\ \hline \mathit{girl}(X) \\ \neg \; \begin{array}{|l|}\hline Y \\ \hline \mathit{cat}(Y) \\ \mathit{owns}(X,Y) \\ \\ \hline\end{array} \\ \\ \hline\end{array}$$

Kamp and Reyle (1993) usually treat negated sentence as having the first meaning with everything in them negated.

However, sentences with proper names lack this ambiguity. This is because the referents and conditions associated with proper names must always rise to the top DRS.[2]

(2.11) *Heather does not own a cat*

is represented as:

$$\begin{array}{|l|}\hline X \\ \hline \mathit{heather}(X) \\ \neg \begin{array}{|l|}\hline Y \\ \hline \mathit{cat}(Y) \\ \mathit{owns}(X,Y) \\ \\ \hline \end{array} \\ \\ \hline \end{array}$$

Thus, there can be only one DRS for example (2.11).

However, the ambiguity in meaning still exists with example (2.11). The reading that *It is not the case that there exists an individual named Heather who owns a cat* still exists. Either this reading cannot be represented by a DRS, or the DRS for example (2.11) now inherits the ambiguity in meaning. Either way, this seems to be a failing in DRT.

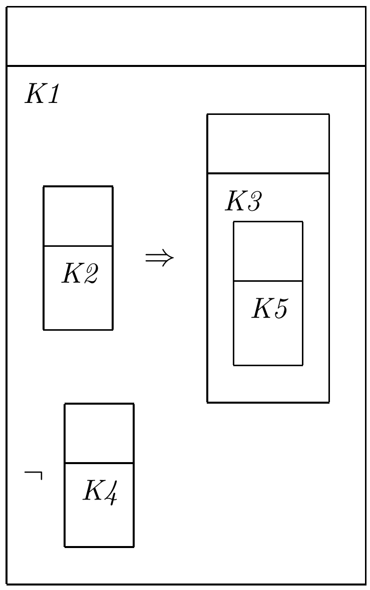
2.2.3 Other DRS Sub-Structures

Besides conditionals and negation, DRS sub-structures can be used to represent other features of natural language. Kamp and Reyle (1993) present methods of constructing DRSs for disjunctions, conjunctions, and tense and aspect. PluralDRT, my plural extension to the Covington, Nute, Schmidt, and Goodman (1988) DRT program, does not deal with these phenomena, hence they will not be further discussed in this thesis. Instead, see Brown (forthcoming), Kamp (1981), Roberts (1987) and Spencer-Smith (1987). In the next few chapters, I will discuss the semantics of plurals and DRS sub-structures representing plurality and generalized quantifiers.

[2]This is in keeping with the analysis put forth by philosophers such as Kripke (1972) that proper names directly refer to the person.

2.2.4 Anaphora Resolution and DRS Accessibility

In anaphora resolution, the search for the antecedent's referent is limited by the accessibility relating among sub-DRSs. DRT specifies that the search begins with the current DRS's universe of discourse and searches up through superordinate DRSs, checking the Universe of Discourse for matching referents. DRS_1 is superordinate to DRS_2 if DRS_1 either contains DRS_2 or is the antecedent of a conditional which has DRS_2 as the consequent.



In the above figure, *K1* is superordinate to everything. *K2* is superordinate to *K3* and *K5*. *K3* is superordinate to *K5*. *K5* and *K4* are superordinate to nothing. Thus, if the pronoun *he* were encountered while constructing DRS *K1*, the anaphora resolver could only search *K1*'s Universe of Discourse. If *he* were in *K2*, DRT could search *K2*'s and *K1*'s Universes of Discourse. If *he* were in *K3*, DRS could search *K3*'s, *K2*'s, and *K1*'s Universes of Discourse.

Chapter 3

Plurality

3.1 What Do Plurals Denote?

What is a plural? Or rather what do plurals denote? It is generally agreed that singular terms such as *Bill*, *a cat*, and *the dog*, denote individuals in the domain of discourse—or at least singleton sets of individuals.[1] Are plurals simply sets of these individuals, or are they entities themselves? Several approaches to this question have been taken.

One approach to plurals, taken by Bennett (1975), Hoeksema (1983), and Scha (1981), is based on set theory. This approach sees singular terms as denoting individuals in the domain, and plurals as sets of individuals. For example, the word *men* denotes a set of sets of two or more men. Thus, if the domain contained the individuals *Bill*, *Bob*, and *Dave*, then the word *men* would denote the set {{Bill, Bob}, {Bill, Dave}, {Bob, Dave}, {Bill, Bob, Dave}}, and the word *man* would denote the set {Bill, Bob, Dave}.

Several objections to this approach were brought up by Link (1983):

1. A set-theoretic account treats plurals and singular terms as two different types of objects.[2] However, natural language in many ways treats them as the same type. For example, questions can easily be answered with a singular or plural term. The sentence *Who chased the cat up the tree?* can be answered with either *Chester* or *the dogs*.

[1] In Bennett's (1975) theory, singular terms denote individuals, while Scha (1981) says that singular terms denote singleton sets of these individuals.

[2] While Scha makes the improvement of treating singular and plural terms as the same type, Link's remaining objections apply to Scha's theory.

2. Plurals are not treated as concrete entities, as singular terms are, but rather as abstract objects that do not exist in the domain. Landman (1989) gives a situational example to illustrate this. Suppose Link's children, having just messed up a room, leave it. Link finds the room in a total mess, and he wants to place the blame on someone. That is, he wants to blame his children. If one child had messed the room up, he could place the blame on that individual, but he finds it hard to blame a set.

3. If plurals are sets, then the following sentences should be correct:

   (a) *Heather, Sage, and Chester has three elements.*

   (b) *My children has two elements, or my children is a two element set.*

   However, these sentences sound odd.[3]

Link (1983) introduced an alternative account of plurals that treats both singular and plural terms as individuals in the domain. Plural individuals consist of parts, and hence are divisible. Singular individuals are indivisible, having only themselves as parts. For example, in the sentence

(3.1) *Bill and Bob lift the stone*

the plural individual *Bill and Bob*, denoted *Bill+Bob*, is made up of the parts *Bill* and *Bob*. However, in the sentence

(3.2) *Bill lifted the stone*

Bill is the singular individual that consists of only one part, *Bill*. Plural individuals form a lattice-like structure in the domain of individuals. Adding another man, *Dave*, to example (3.1), we get a structure of the form:

In Link's theory, the word *men* denotes the set of plural individuals who are men. That is, if the domain of men consist of *Bill*, *Bob*, and *Dave*, then

[3]Landman points out that Link's argument can be applied to his theory as well. The sentence

Peter, Paul and Mary is the join of three atoms

sounds just as odd. Landman attributes this to Link's blurring of use-mention words. That is, Link is using mathematical notation with non-mathematical entities.

the word *men* refers to the set of plural individuals: $\{Bill{+}Bob,\ Bill{+}Dave,\ Bob{+}Dave,\ Bill{+}Bob{+}Dave\}$.

Link's theory deals with mass terms more appropriately than set theory does. Set-theoretic approaches try to make mass terms into some kind of sets. This approach falls apart because words such as *water* are not understood in terms of indivisible elements. Link proposes treating such terms as plural individuals. The only difference between mass terms and other plurals is that the other plurals can be divided into indivisible parts, while the mass terms have only divisible parts. Since mass terms are not used in Kamp and Reyle's (1993) version of DRT nor in my PluralDRT, they will not be discussed any further, but it is important to point out that Link's treatment of mass nouns corresponds more closely to natural language than a set-theoretic treatment would.

Kamp and Reyle continue to use *sets* to denote divisible individuals, and *elements* to denote indivisible individuals, and I shall do the same. This is legitimate because Kamp and Reyle show that models in their version of Link's theory can be converted to set-theoretic models. Thus, from this point of view, the only advantage of Link's theory is it corresponds more closely to natural language.

3.2 Collective and Distributive Plurals

3.2.1 Introduction

Plural sentences can have an inherent ambiguity in their reading. Consider the sentence

(3.3) *Five men sang.*

This sentence is true in at least three situations:

1. Five men jointly sang.

2. Five men each took turns singing.

3. Three men sang together, while the other two sang separately.

The first reading is called the *collective* reading, the second is called the *distributive* reading, and the third is called the cumulative (Scha 1981), or sometimes conjunctive (Webber 1983) reading. In the cumulative reading, it is possible for any number of subsets of the *men* to have sung together or as individuals as long as the number in the subsets *and* the number of individuals adds up to 5. This reading seems to be very important because it encompasses all other readings (Franconi 1993). That is, both collective and distributive readings seem to be special cases of the cumulative. However, PluralDRT does not deal with the cumulative readings, and hence they will not be discussed any further here.

Some plural sentences lack this ambiguity. The following sentence has only a collective reading:

(3.4) *The men gathered in the square.*

This is because the verb *gathered* only takes a plural subject. Thus, *a boy* could not be used as a subject, unless the verb *gathered* is used in a different sense such as

(3.5) *A boy gathered bricks in the square.*

Link (1983) gives an example of a sentence with only a distributive reading:

(3.6) (a) *John, Paul, George, and Ringo are pop stars.*
(b) *Paul is a pop star.*

Example (3.6a) automatically entails example (3.6b). This is because the predicate, *be a pop star*, is true for individuals only. This is also seen in the reverse, *John, Paul, George, and Ringo are a singing group* does not then entail *John is a singing group.* Thus, *be a singing group* has a strictly collective reading.

3.2.2 Collective

Various factors cause a sentence to have a collective reading. The verb can force a collective reading as in example (3.4). Adverbials such as *together* can also force a sentence to have a collective reading. The sentence *The men sang* can mean that the men sang separately, but

(3.7) *The men sang together*

no longer has a distributive reading.

3.2.3 Distributive

Factors which can cause sentences to have strictly distributive readings are predicates, adverbials and determiners. Example (3.6) illustrates a predicate which forces distributivity. An example of an adverb causing distributivity is the following sentence:

(3.8) *The men each sang the song.*

The adverb *each* acts as a quantifier with a scope of the entire sentence. Its effect is that the predicate, *sang a song*, applies to every member of the subject individually rather than as a group. Finally, a determiner that causes a sentence to be strictly distributive is *most* in the following sentence:

(3.9) *Most cats hunt at night.*

Even if the cats in question hunt in packs, the statement in example (3.9) is being made about the individual cats, not the packs. For in-depth discussion of theories of both distributive and collective readings in plural sentences, see Bennett (1974), Hoeksema (1983), Link (1983), Roberts (1987), and Scha (1981).

3.2.4 DRT Representations

Collective Representations

Kamp and Reyle deal with collective—and distributive—readings of plural sentences by introducing discourse referents for sets of individuals. For example, the sentence

(3.10) *Five dogs chased a cat*

is represented as:

$X\ y$
$dogs(X)$ $\lvert X \rvert = 5$ $cat(y)$ $chased(X,y)$

where X refers to a set of dogs and $|X| = 5$ shows that there are five of these dogs.[4] *chased(X,y)* then shows that all five dogs chased the cat, because the set referent, X, is used.

Truth conditions for collective readings rely on proper embeddings in the model. Thus, the set referent must refer to a set of individuals in the domain such that all conditions associated with the referent are satisfied by the set of individuals. So, in example (3.10), if there exists a set of individuals, A, in the domain where A has five members, all of them dogs, and there exists another individual b in the domain such that b satisfies the condition of being a *cat* and further A *chases* b, then the DRS for example (3.10) is true.

This method works for any sentence in which a plural is explicitly mentioned, e.g., *dogs*. But consider the sentence

(3.11) *Bill, Bob, and Dave lifted the piano.*

This sentence clearly has a collective plural reading, but there is no plural explicitly mentioned. Thus DRT cannot introduce a discourse referent for the set of individuals.

In order to handle sentences such as (3.11), DRT uses a process called *summation*. It creates a referent for the set of individuals, *Bill*, *Bob*, and *Dave*, which contains their individual referents. Thus a DRS for example (3.11) looks like:

$x\ y\ u\ v\ W\ Z$
bill(x) *bob(y)* *dave(u)* *piano(v)* $W = x \oplus y \oplus u$ $Z = W$ *lifted(Z,v)*

$W = x \oplus y \oplus u$ denotes the summation of the referents x, y, and u, where the $\oplus$ is essentially the $+$ in Link's theory.

[4]From now on, capital letters will denote sets, and lower-case letters will denote individuals.

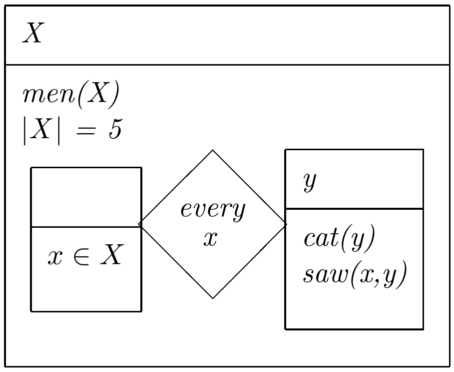
DRSs which use summation must satisfy extra conditions in order to be true. The set formed by summation must refer to a set in the domain, and each referent in the formed set must refer to an individual in the domain set that satisfies all conditions associated with the referent. Thus, the DRS above is true if in the domain there exists a set of individuals $\{a, b, c\}$ and another individual d, such that a has the name *Bill*, b has the name *Bob*, c has the name *Dave*, and d is a *piano*. Further, the set $\{a, b, c\}$ must satisfy the condition of having lifted d.

Distributive Readings

DRT uses sub-structures called *duplex conditions* to represent distributive readings. The sentence

(3.12) *Five men each saw a cat*

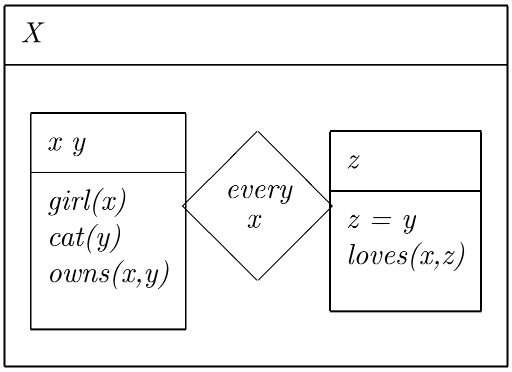
is represented as



The above duplex condition is essentially the if–then condition introduced earlier. Further, if–then conditions will be represented in this form for the remainder of this thesis. For example the sentence

(3.13) *Every girl who owns a cat loves it* or *If a girl owns a cat, then she loves it*

is represented as:



A duplex condition with *every* in the diamond[5] is true if every embedding of the left hand side can be extended to satisfy the right hand side. That is, any individual in the domain which satisfies the conditions in the left hand side must also satisfy the conditions in the right hand side. To illustrate this, consider example (3.12). Any individual who satisfies the condition in the left hand side (i.e., membership in X) must also satisfy the condition of seeing another individual who is *a cat*. Suppose there exists a set of five individuals $\{a, b, c, d, e\}$ such that the individuals all satisfy the condition of being in the group of five men, which means they must satisfy the condition of being men. Further, a satisfies the condition of *seeing* another individual f_a, which satisfies the condition of being a *cat*, and so on for all members of the set. If every member of the group satisfies these conditions, then the DRS for (3.12) is true.

3.3 Plural Anaphora

3.3.1 Anaphora and Collective Readings

Simple Anaphora

Collective readings of simple sentences with explicit anaphora are treated like singular anaphora.[6] Consider the discourse:

(3.14) *Five men bought a pizza. They ate it.*

[5]Duplex conditions which have other quantifiers in the diamond will be introduced later.

[6]The term simple sentence refers to sentences which have no hypothetical conditions in them.

The DRS for this discourse looks like:

$X\ y\ Z\ u$
men(X) $\|X\| = 5$ *pizza(y)* *bought(X,y)* $X = Z$ $y = u$ *ate(Z,u)*

where Z and u are the discourse referents for *they* and *it* respectively. The anaphors *they* and *it* are treated alike except that one denotes a set and the other denotes an individual.

Summation

DRT uses summation in sentences which have conjoined proper names or conjoined singulars as the antecedents of plural anaphors. For example, the discourse

(3.15) *Bill, Bob and Dave bought a pizza. They ate it.*

has the plural anaphor *they* referring to the set {*Bill*, *Bob*, *Dave*}. Because the set is not explicitly mention, it has no discourse referent. However this problem is solved automatically since DRT processes discourses one sentence at a time. When the first sentence is processed, a referent for the set *Bill*, *Bob*, and *Dave*, is created using summation. This referent is then the antecedent for the referent of *they* when the second sentence is processed.

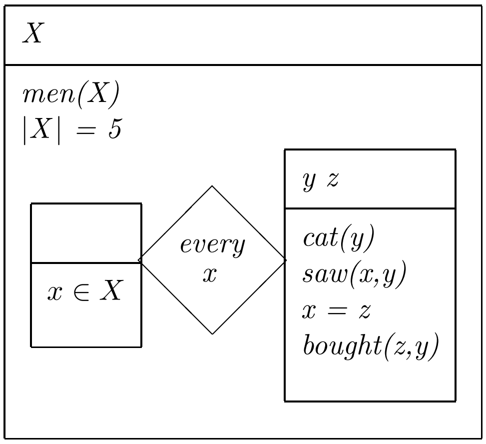
3.3.2 Anaphora and Distributivity

Anaphora Inside the Sentence Scope

Anaphors used inside the scope of sentences with a distributive reading do not seem to present any difficulties. This is because DRT has relatively few problems accessing the antecedent's referent. An example is the sentence

(3.16) *Five men each saw a cat which they bought.*

The DRS for the distributive reading is



This example also shows a case of a plural anaphor referring to an individual rather than a set of individuals. The relative clause *which they bought* has the meaning that each individual man bought the cat he saw. This differs from example (3.14) in that the anaphor must match the number of the antecedent, but *they* is plural and x (in the DRS) is singular. Kamp and Reyle solve the problem by giving a singular referent introduced from a plural referent, $x \in X$, a plural superscript, x^{pl}, which marks it as able to be the antecedent of a plural anaphor.[7]

Anaphoric Subordination

Sometimes a singular noun phrase can only be referred to by a plural pronoun in a subsequent sentence, because its reference is dependent on a distributively understood plural. Consider the following discourse

(3.17) *Five men each saw a cat. They fed them.*

where the first sentence has the distributive reading of the men individually seeing a cat,[8] and the second sentence has two possible readings, a *strict* and *sloppy* reading.[9] The strict reading is the five men got together and jointly

[7] For brevity, the superscript will usually be assumed and not explicitly shown.

[8] The men do not necessarily see the same cat. Although it is possible for them to have seen the same cat, it is not relevant for our discussion.

[9] This is similar to what Reinhart (1983) calls *sloppy identity*.

fed all the cats, while the sloppy reading is that each man feed the cat he saw. Either way, in subsequent sentences, the cats (implicitly one for each man) can only be referred to as *they*, not *it*. In effect, *cat* picks up plurality from *men*. I shall call this phenomenon *anaphoric subordination*.[10]

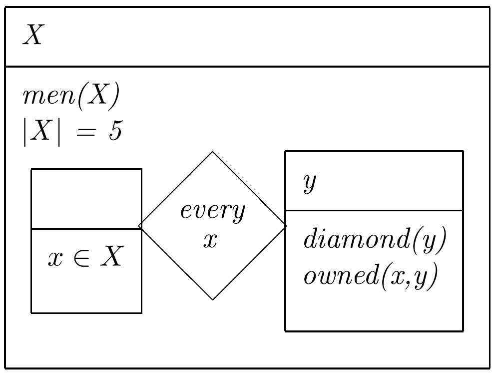
Sentences which have the object of the verb as the distributed plural do not have this problem with anaphor reference. For example,

(3.18) *A man saw five buildings. He liked them.*

The anaphors are able to directly refer to both *a man* and *five buildings*. Here the plurality of *buildings* has narrower scope than the singularity of *man*; that is, *man* is not Skolem dependent on *buildings*.

A further problem is caused by DRT structures. The duplex conditions used in representing distributive readings may cause accessibility problems in anaphor resolution. An example of this problem is the following sentence and DRS:

(3.19) *Five men each owned the diamond*



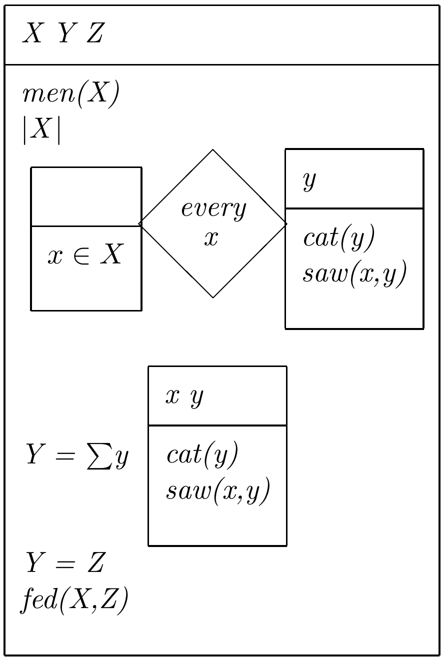
In English, *They loved it* can follow example (1.19) without conflict, but DRT is unable to access the right hand side of the duplex condition for the antecedent of *it*.

Handling Strict Readings in DRT

In order to handle strict readings, DRT needs to collect the referents for the individuals denoted by the object of the verb, *a cat* in example (3.17),

[10] At the suggestion of Dr. Michael Covington.

into a set, so that it may be referred to. Kamp and Reyle use a process called *abstraction* to form a referent for the set of individuals that satisfy the conditions prescribed by the antecedent. That is, in example (3.17) a referent is formed for the set of cats who were seen by the five men. So the strict reading of example (3.17) has the following DRS:



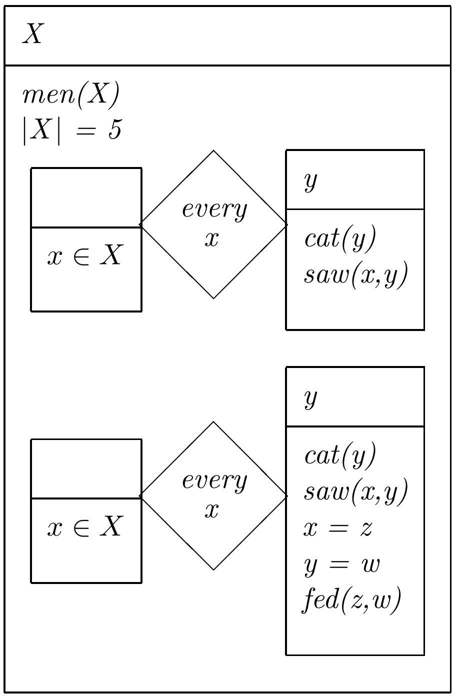
where $Y = \sum y \begin{array}{|l|}\hline x\ y \\ \hline cat(y) \\ saw(x,y) \\ \hline\end{array}$ is abstraction.[11] This simply collects all the referents of individuals who satisfy the conditions of being *a cat* and having been seen by at least one of the individuals in X into a set, and then gives this set a referent, i.e., Y.

[11]This method is odd because x in the duplex condition can be accessed by the DRS used in abstraction, which it should not be able to do. This can be fixed in two different ways. First, putting the condition $x \in X$ in the abstraction DRS, or second, putting the abstraction DRS in a second duplex condition which copies the information of the first.

For the abstraction condition to to be true, there must exist at least one individual that satisfies the conditions in the sub-DRS involved in the abstraction. Thus if individuals a, b, $\ldots$ have the property of being a *cat*, and each cat has been seen by at least one of the individuals in the set of five men, then the above DRS is true. If there is only one individual in the abstracted set, then each member is distributed to the same object. In the example above, this means that each man saw the same cat.

Handling Anaphoric Subordination in DRT

Kamp and Reyle handle cases of anaphoric subordination by creating a second duplex condition which contains the information of the first duplex condition plus the additional information from the second sentence. The DRS for the sloppy reading of example (3.17) would then look like

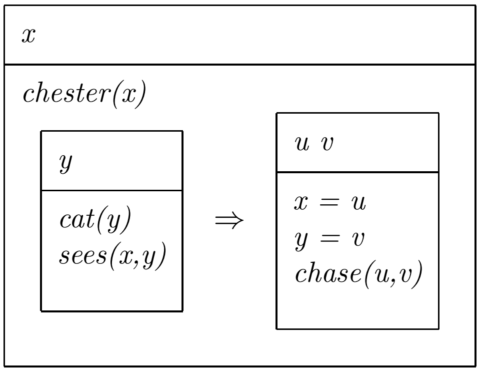


This method seems wrong. The additional information, *fed(z,w)*, should really go in the right hand side of the first duplex condition, but it cannot since the right hand side is not accessible once the scope of the sentence has ended. Kamp and Reyle use two different duplex conditions to handle this. This method seems faulty for at least two reasons. First, there is a repetition of information which does not happen in natural language. Since DRT is trying to represent natural language, it seems inherently wrong to repeat something when it is not done in the natural language sentence. Second, this method seems to imply that two conditions could have different truth values, which is not the case.[12]

Anaphoric subordination is similar to another phenomena involving the accessibility of antecedents. *Modal subordination* (Roberts 1987) occurs when the mood (modus) of a sentence in a discourse causes it to be subordinate to another sentence, acting as a clause to that sentence.[13] An example of modal subordination is the following discourse:

(3.20) *If Chester sees a cat, he will chase it. It will soon grow tired.*

A hypothetical situation is formed in the first sentence; the second sentence continues this situation by using the modal *will*, thus adding more information. However, DRT limits the scope of the hypothetical situation to the first sentence making it inaccessible to the rest of the DRS. The DRS for the first sentence looks like:



This is similar to the problems associated with anaphoric subordination. Roberts gives a solution which she calls the *insertion approach*. The insertion approach simply takes the information of the second sentence and inserts it into the right hand side of the sub-DRS. Roberts seems to have

[12]This was suggested by Dr. Michael Covington.

[13]This phenomenon was also noticed by David Goodman (1988). He called it *Multi-sentence consequents*.

the better solution to this problem and anaphoric subordination as well. Her solution does not involve unnecessary repetition of information, nor are there any problems with truth–values.

3.4 Definite Descriptions

The type of definite descriptions discussed in this section are noun phrases with the determiner *the* heading them. Other definite descriptions, proper names, anaphora, etc., are discussed in the sections pertaining to them. DRT primarily focuses on these definite descriptions as anaphora or as dependent plurals.

3.4.1 Anaphors Masquerading as Definite Descriptions

Definite descriptions can act as anaphors by referring back to individuals or groups of individuals previously mentioned. An example is the discourse

(3.21) *Five men saw a cat. The men fed it.*

The men in the second sentence acts as a discourse referent for *five men* in the previous sentence. Thus, DRT treats *the men* as anaphora. A DRS for example (3.21) looks like:

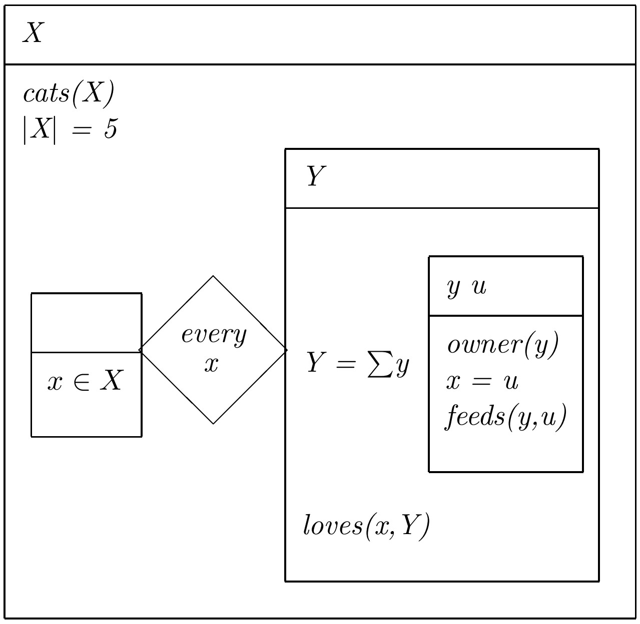
$X\ y\ Z\ v$
men(X)
$\lvert X \rvert = 5$
cat(y)
saw(X,y)
$X = Z$
$y = v$
feed(Z,v)

3.4.2 Dependent Plurals

A plural definite description can sometimes depend on another noun phrase. Consider the sentence

(3.22) *Five cats love the owners who feed them.*

where *the owners* is dependent on the noun phrase *five cats*. Although there may be only one owner per cat, the plurality of *five cats* forces *the owners* to also be plural. Further, the referent for *the owners* is contingent on the individual referents of the five cats, since each cat loves the owner who feeds it. This is regardless if the owner owns that particular cat or not. The DRS then looks like:



Y can contain one or more individuals. Hence one or more owners can feed a particular cat.

3.4.3 Other Roles of Definite Descriptions

There have been several views of definite descriptions taken. One such view, held by Russell (1905) sees definite descriptions as denoting specific, unique objects in the universe. For example,

(3.23) *The queen of England has grey hair*

refers to the one and only queen of England. If there is not an individual in the universe which fits this description, then the sentence is considered to be false. This can be extended to include plural definite descriptions. Consider the sentence:

(3.24) *The boys in that gang are mean.*

The boys refer to the unique set of individuals who are the boys in that gang.

Strawson (1950) said that a statement was not false if it failed to refer to an individual. Instead the sentence failed the presupposition that it referred to an existing object, and since it does not, it has no truth value.

Donnellan (1966) introduced another view of definite descriptions. Not all definite descriptions are used referringly; some are used in an attributive manner. Suppose the president of AA has been told that there is a man with a glass of champagne at the meeting. The president might reply, "The man drinking champagne should not be at an AA meeting." The president does not have to have a particular individual in mind when he says this; he is simply *attributing* it to some person who he thinks is drinking champagne at an AA meeting.

Kamp and Reyle do not generally deal with definite descriptions other than as anaphors and dependent plurals. Instead they are simply inserted into the DRS along with the noun phrase they determine. For example, the sentence

(3.25) *The cats claw a chair*

has the DRS

X y
the cats(X) *chair(y)* *claw(X,y)*

where Kamp and Reyle do not further analyze *the cats*.

3.5 Generics

Generic statements are statements that give general rules about the world. They are general because they can sometimes be contradicted without making them false. Statements which contradict these rules are called *exceptions*. An example of a generic statement is

(3.26) *Cats claw.*

An exception for this might be *My cat has been declawed and hence cannot claw*. *My cat* is an example of a cat who does not fit this generic statement. However, the rule still holds for cats in general.

Generic statements are intuitive rules that language speakers make in every day life and are usually based on empirical evidence. These statements are notable because they can be headed by such phrases as *in general* and *generally*.

Rules which have no exceptions are called *strict rules*. An example of such a rule is

(3.27) *Cats are mammals.*

There is no cat that exists which is not a mammal. This rule cannot be defeated and is therefore strict.

Generic statements can be expressed in several different forms, both singular and plural. Some examples (Izzo 1993) of these forms are:

(a) *A cat is a hunter.*

(b) *Cats are agile.*

(c) *Cats meow.*

Kamp and Reyle (1993:294) represent these generic statements as follows:

$$\boxed{\begin{array}{c} \quad \\ \hline \boxed{\begin{array}{l} x \\ \hline cat(x) \end{array}} \rightsquigarrow \boxed{\begin{array}{l} \quad \\ \hline hunter(x) \end{array}} \end{array}}$$

DRT treats these conditions similarly to the if–then conditions with the exception that these rules are not universal, but hold to normal or general cases. The $\rightsquigarrow$ signifies that the rule is a loose rule rather than the strict interpretations associated with $\Rightarrow$.[14]

[14] Izzo (1993) does not use this exact notation, but the idea is the same.

3.6 Generalized Quantifiers

Generalized quantifiers include not only ∀ and ∃, but also other quantifiers such as *most* or *five*.

3.6.1 Generalized Quantifiers are Relations among Sets

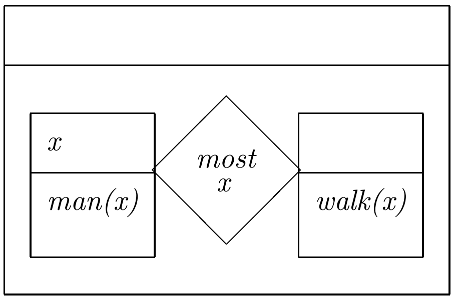
The sentence

(3.28) *Most men walk*

seems to be asserting that the men in the domain who are men that walk are more than half of all the men. This idea can be generalized to: Most Qs are P if and only if the set of Qs that are P is more than half the entire set of Qs. Analogous definitions can be given for *few*, *some*, *not all*, and so forth. Thus quantifiers are seen as a relation among sets, where the quantifier compares the set of Qs that are P to the set of Qs (Kamp and Reyle 1993:315).[15] This definition of quantifier holds for ∀ and ∃. The quantifier ∀ Qs are P is true iff the set of Q's that are P is the entire set of Qs. ∃ Q is P is true iff the set of Qs that are P share at least one element with the set of Qs.

3.6.2 Generalized Quantifiers in DRT

Duplex conditions represent generalized quantifiers. Example (3.30) is represented as follows:



The diamond containing *most x* represents the quantifier relating the set denoted by the right hand side of the duplex condition to the set denoted by

[15]Barwise and Cooper (1981) introduced a version of generalized quantifiers in natural language. However, I am describing Kamp and Reyle's version of generalized quantifiers, which differs from Barwise and Cooper's.

the left hand side. The middle part of the condition is called the *quantifier* with the left hand side its *restrictor* and the right hand side its *scope*. That is, this is a picturization of a generalized quantifier.

The duplex condition is true if *most* of the embeddings of the left hand side also satisfies (can be extended to satisfy) the right hand side. If the individuals who satisfy the conditions of the left hand side and the right hand side are more than half of all the individuals who satisfy the left hand side, then the duplex condition is true. Suppose the domain contains five individuals, $\{a, b, c, d, e\}$ who are *men*. Further, only individuals a, c, and d satisfy the condition of *walking*. Then, the duplex condition is considered to be true.

Chapter 4

PluralDRT

PluralDRT uses a top-down parser with definite-clause grammar (DCG) notation. Feature structures are used during parsing to pass syntactic and semantic information along. Number, case, and subject-verb agreement are enforced through the unification of these feature structures.

4.1 Representing Plurals in the Database

Singular terms are represented as specified in the original program (Covington, Nute, Schmitz, and Goodman 1988). *A man* is represented in the database as `man(1)` where `1` is a constant which denotes the individual. Notice that the individual exists in the database only in terms of the conditions associated with it.

Collective Plurals

Following Link's theory, PluralDRT represents plural terms, as well as singular terms, as concrete individuals in the domain. Collective plurals, in particular, are denoted as a single entity. *Five men* is represented as

(4.1)
```
men(1).
card(1,5).
```

`men` is left in the plural form to denote that this is a collective individual. `1` is the constant which represents the individual, and `card(1,5)` shows that `1` has five members or parts. In (4.1) PluralDRT is not concerned with the individual members of `1`, only the necessary information about it.

Distributive Plurals

Distributive plurals have a more complex representation. This is because it is necessary to represent both the individual members as well as the set itself. *Five men* has the representation:

(4.2)
```
man(1).          element(1,6).          card(6,5).
man(2).          element(2,6).
man(3).          element(3,6).
man(4).          element(4,6).
man(5).          element(5,6).
```

The individuals of the set are represented as singular terms, while `element` acts as a pointer linking each individual member to the set. Thus `element(1,6)` gives the information that `man(1)` is an element of the set, `6`. `Card` gives the cardinality of `6` that it has 5 members. This representation scheme is based on Covington (1994).

4.2 Building DRSs

DRSs are of the form `drs(U,Con)` in Prolog, where `U` is a list of the referents, and `Con` is a list of all the conditions associated with the referents. *Blackie meows* would be represented in Prolog as: `drs([_01234], [blackie(_01234),meows(_01234)])`.

PluralDRT uses feature structures to build DRSs during parsing. These feature structures collect semantic information, which is added to either **U** or **Con**. Thus, as a sentence in the discourse is parsed, the rules pass information from one rule to the next using unification. The parse rule for **s** demonstrates this:

(4.3)
```
s(S,H1,H2) -->
  { NP = sem:A,
    S  = sem:A,
    NP = sem:scope:C,
    NP = syn:index:D,
    NP = syn:case:nom,
    NP = syn:number:Num,
    VP = sem:C,
    VP = syn:number:Num,
    VP = syn:arg1:D },
    np(NP,H1,H2), vp(VP,H2,H3).
```

Briefly, the **s** rules pass the incoming semantic information to the **np** rules through unification of the feature structure, `sem`, `S` and `NP`.This also means that the semantics of the noun phrase is the semantics of the sentence. The **np** rules parse the noun phrase and add the information to the DRS in progress. This information is also passed to the **vp** rules through the unification of `VP`'s `sem` with `NP`'s `scope`. Since the `VP` is the scope of the `NP`, it is also the scope of the sentence. The **vp** rules parse the verb phrase, adding the information to the current DRS, as the **np** rules did. Further, when the verb phrase is completely parsed, `S` will contain the semantic information of the sentence. Feature structures containing syntactic information are used for agreement during parsing.

Feature structures are of the form:

$$\begin{bmatrix} \textit{syn:} & \begin{bmatrix} \textit{index:}\ldots \\ \textit{case:}\ldots \\ \textit{class:}\ldots \\ \textit{number:}\ldots \\ \textit{type:}\ldots \\ \textit{arg1:}\ldots \\ \textit{arg2:}\ldots \end{bmatrix} \\ \textit{sem:} & \begin{bmatrix} \textit{in:}\ldots & \\ \textit{res:}\ldots & \begin{bmatrix} \textit{in:}\ldots \\ \textit{out:}\ldots \end{bmatrix} \\ \textit{scope:}\ldots & \begin{bmatrix} \textit{in:}\ldots \\ \textit{out:}\ldots \end{bmatrix} \\ \textit{out:}\ldots & \end{bmatrix} \end{bmatrix} \tag{4.4}$$

The `syn` information is used for agreement, passing information and distinguishing types of plurals. `Index` is the discourse referent, which is a variable in Prolog. `Case`, `class`, and `number` are the syntactic information for agreement. `Case` is nominative or accusative, `class` is proper or common, and `number` is singular or plural. `Type` is for plural common nouns and distinguishes whether the noun is treated as a collective or distributed. `Arg1` and `arg2` are the referents for the subject and objects of the verb respectively.

The `sem` information is used for DRS building. `In` contains prior semantic information coming into the rules. The information can consist of DRSs of previous sentences or semantic information from previous parts of the sentence which have been parsed already. This is used to give semantic cohesion to the DRSs by providing information from previous sentences of the discourse. `Out` is the complete information once the sentence is parsed. `Res` and `scope` are the semantic information used in forming the logical structure of the sentence.

4.2.1 The Role of the Determiner

The role of the determiner in PluralDRT, as with the original program, is to shape the logical form of the sentence. The determiner acts as a quantifier over the sentence with two arguments: the restrictor and the scope. The restrictor is the rest of the noun phrase, and the scope is the verb phrase. PluralDRT does this in the determiner parse rules.

Naturally these rules differ for each determiner, but they also differ for the type plural headed by the determiner. Sentences with collective plurals have a very different logical structure than sentences with distributive plurals. Collective plurals are often treated like singular sentences, which is reflected in the logical structure, and distributive plurals are treated similarly to universally quantified statements. The difference being that the distributive plurals act within a certain domain, whereas universally quantified statements range over the entire domain. Thus, there are two different rules for the same determiner. For example, here are the rules for the cardinal number *five*:

(4.5) (a)
```
det(DET) --> [five],
{  Det = syn: (number:pl ::
               type:col  ::
               det: (type:card ::
                     amt:5)),
   Det = sem:in:A,
   Det = sem:res:in:A,
   Det = sem:res:out:B,
   Det = sem:scope:in:B,
   Det = sem:scope:out:C,
   Det = sem:out:C }.
```

(b)
```
det(DET) --> [five],
{  Det = syn: (number:pl ::
               type:ind  ::
               det: (type:card ::
                     amt:5)),
   Det = sem:in:A,
   Det = sem:res:in:[drs([],[])|A],
   Det = sem:res:out:B,
   Det = sem:scope:in:[drs([],[])B],
   Det = sem:scope:out:[Scope,Res,drs(U,Con)|Super],
   Det = sem:out:[drs(U,dist(Res,Scope)|Con])|Super] }.
```

(4.4a) is the rule for collective plurals, and (4.4b) is the rule for distributive.

The determiner *the* in rule (4.4a) works as the determiner *a* does (Covington and Schmitz 1988), except that cardinality information is passed along with the DRS in progress. The rule takes information coming in from previous sentences—other DRSs—and passes it to the restrictor. The rules dealing with the restrictor then add the semantic information to the feature structures, and pass all the information to the scope. The rules for the scope add it's information to the feature structures, and this forms the semantic information of the sentence. This determiner rule is very simple; there are no complex substructures.

Rule (4.4b) is much more complex and does involve a sub–structure. The rule here is very similar to the parse rule for *every* (Covington and Schmitz 1988). The process works nearly the same as rule (4.4a) until the scope output. The output of the scope is formed into a list where the first three elements are a list containing the information from the scope, a list containing the information from the restrictor, and the previous DRSs. The scope and restrictor are then placed in a sub-structure which is headed by the predicate `dist`. This notifies PluralDRT during the translation process that the substructure is a distributive plural, and PluralDRT can treat it accordingly. When there is no determiner, the noun phrase rules act in the determiner's place.

4.3 Prolog Structures from DRSs

4.3.1 Asserting

Simple DRSs

Since DRS conditions are similar to Prolog predicates, translating from DRT to Prolog can be quite easy. Consider a simple example such as

(4.6) *A dog chases a cat.*

$$\begin{array}{|l|}\hline x\ y \\ \hline dog(x) \\ cat(y) \\ chases(x,y) \\ \\ \hline\end{array}$$

First, two constants would be randomly chosen to represent the individuals x and y. Then, the conditions, with the constants instantiated for the variables, would be asserted into the database as:

```
dog(1).
cat(2)
chases(1,2).
```

Translating DRSs with collective verbs is a very similar process. Constants replace variables for sets and individuals. The cardinality information is also added. These conditions are then asserted into the database. For example, (4.7) *Five dogs chase a cat.*

$X\ y$
dogs(X) $\|X\| = 5$ *cat(y)* *chases(X,y)*

would be translated as:

```
dogs(1).
card(1,5).
cat(2).
chases(1,2).
```

Thus, PluralDRT will give the following output if given the collective sentence:

(4.8) *Two dogs chase a cat.*

```
Sentence: Two dogs chase a cat.

[two,dogs,chase,a,cat,.]
--------------------------------------------------
Originally constructed DRS:
--------------------------------------------------
_8863
_9202
number(_8863,pl)
gender(_8863,n)
case(_8863,nom)
dogs(_8863)
|_8863|=2
number(_9202,sg)
gender(_9202,n)
case(_9202,acc)
```

```
cat(_9202)
chases(_8863,_9202)

--------------------------------------------------
Cleaned-up (simplified) DRS:
--------------------------------------------------
_8863
_9202
dogs(_8863)
|_8863|=2
cat(_9202)
chases(_8863,_9202)
--------------------------------------------------
Asserting: card(1,2)
Asserting: dogs(1)
Asserting: cat(0)
Asserting: chases(1,0)
--------------------------------------------------
```

Complex DRSs

Translating conditions which use sub–DRSs requires the use of an operator defined in both PluralDRT and the original program, `::-`. This operator allows a Prolog clause to have multiple consequents. It would be used in the following manner:

(4.9) `a,b ::- c,d.`

where `a`, `b`, `c`, and `d` are Prolog predicates. Since this operator cannot be asserted into the database, clauses headed by `::-` are changed into several clauses using `:-`. This is *distributing the consequents*. Example (4.8) is asserted as:

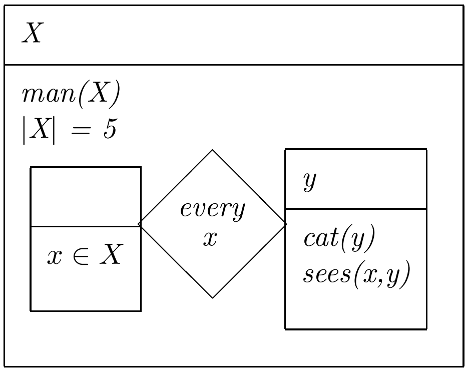
(4.10)
```
a :- c,d.
b :- c,d.
```

This operator is especially useful on any complex if-then statements, sentences with *every*, sentences with distributive plurals, and generics. I refer the reader to Covington and Schmitz (1988) and Covington, Nute, Schmitz, and Goodman (1988) for further discussion of translating if-then statements and sentences which contain universal quantifiers into Prolog predicate. I shall now focus on DRSs with distributive plurals and generics.

Distributive Plurals

Since the duplex conditions for distributivity are essentially the if-then conditions, the Prolog clauses for distributivity will be similar to those for if-then structures defined in the original program. An example is

(4.11) *Five men each see a cat.*



Since everything in the right side of the duplex condition is Skolem dependent on the left side, the Prolog structures must reflect this. Thus, the DRS is first translated into the following structures:

(4.12)
```
man(1).                    element(1,6).                 card(6,5).
man(2).                    element(2,6).
man(3).                    element(3,6).
man(4).                    element(4,6).
man(5).                    element(5,6).

sees(_11735, [7,_11735]), cat([7,_11735]) ::- man(_11735),
                                               element(_11735,6),
                                               card(6,5).
```

The consequents of the last structure are distributed, and the following clauses are asserted into the database with the first eleven facts:

(4.13)
```
sees(_11735, [7,_11735]) :- man(_11735), element(_11735,6),
                             card(6,5).

cat([7,_11735]) :- man(_11735), element(_11735,6),
                   card(6,5).
```

This representation causes a logic error to occur. PluralDRT gives the following output if given the sentence *Five dogs each see a cat*:

(4.14)
```
Sentence: Five dogs each see a cat.

[five,dogs,each,see,a,cat,.]
--------------------------------------------------
Originally constructed DRS:
--------------------------------------------------
_11006
number(_11006,pl)
gender(_11006,n)
case(_11006,nom)
dog(_11006)
|_11006|=5

For each x in _11006
  _11374
  sees(x,_11374)
  cat(_11374)
  case(_11374,acc)
  gender(_11374,n)
  number(_11374,sg)

--------------------------------------------------
Cleaned-up (simplified) DRS:
--------------------------------------------------
_11006
dog(_11006)
|_11006|=5

For each x in _11006
```

```
  _11374
  sees(x,_11374)
  cat(_11374)
--------------------------------------------------
Asserting: dog(2)
Asserting: dog(3)
Asserting: dog(4)
Asserting: dog(5)
Asserting: dog(6)
Asserting: element(2,7)
Asserting: card(7,5)
Asserting: element(3,7)
Asserting: card(7,5)
Asserting: element(4,7)
Asserting: card(7,5)
Asserting: element(5,7)
Asserting: card(7,5)
Asserting: element(6,7)
Asserting: card(7,5)
Asserting: sees(_13867,[8,_13867]):-dog(_13867),
           element(_13867,7),card(7,5)
Asserting: cat([8,_13867]):-dog(_13867),
           element(_13867,7),card(7,5)
--------------------------------------------------
```

Generic Statements

PluralDRT does not make a distinction between strict rules and generics rules. Generic rules are treated exactly as universally quantified statements. Thus sentences like:

(a) *A cat is a mammal.*

(b) *Cats are mammals.*

(c) *Cats hunt.*

are considered to be of the form:

(a),(b) *If an individual is a cat, then it is a mammal.*

(c) *If an individual is a cat, then it hunts.*

These then have the following Prolog structures asserted into the database:

(a),(b) `mammal(_11735) :- cat(_11735).`

(c) `hunts(_11735) :- cat(_11735).`

Generics are treated in this manner because in order to deal with them correctly, defeasible reasoning would be needed, which is outside the scope of this thesis. I refer the reader to Izzo (1993) for a program using defeasible reasoning with DRT.

PluralDRT gives the following output if given the sentence *Dogs are happy*:

(4.15) `Sentence: Dogs are happy.`

```
[dogs,are,happy,.]
--------------------------------------------------
Originally constructed DRS:
--------------------------------------------------
IF:
  _15090
  number(_15090,pl)
  gender(_15090,n)
  case(_15090,nom)
  dog(_15090)
THEN:
  happy(_15131)

--------------------------------------------------
Cleaned-up (simplified) DRS:
--------------------------------------------------
IF:
  _15090
  dog(_15090)
THEN:
  happy(_15131)
--------------------------------------------------
Asserting: happy(_15131):-dog(_15090)
--------------------------------------------------
```

4.3.2 Querying

Questions are often represented using sub-DRSs to distinguish that there is a query occurring. They have the form:

(4.16) *Do five men buy a pizza?*

<table>
<tr><td></td></tr>
<tr><td>QUERY: <table><tr><td>X y</td></tr><tr><td>men(X)
$|X| = 5$
pizza(y)
buy(X,y)</td></tr></table></td></tr>
</table>

The Prolog version for DRSs containing queries is `drs([],[query(drs(...))])`. The DRS is then translated into a structure of the form:
`query([men(_12345),card(_12345,5),pizza(_14141),sees(_12345,_14141)])`, and PluralDRT queries the database with `query`'s argument.

Suppose *Five dogs chase a cat* is asserted into the database. PluralDRT then gives the following output when queried *Do five dogs chase a cat?*

(4.17) `Sentence: Do five dogs chase a cat?`

```
[do,five,dogs,chase,a,cat,?]
--------------------------------------------------
Originally constructed DRS:
--------------------------------------------------
QUERY:
  _14260
  _14615
  number(_14260,pl)
  gender(_14260,n)
  case(_14260,nom)
  dogs(_14260)
  |_14260|=5
```

```
  number(_14615,sg)
  gender(_14615,n)
  case(_14615,acc)
  cat(_14615)
  chases(_14260,_14615)

--------------------------------------------------
Cleaned-up (simplified) DRS:
--------------------------------------------------
QUERY:
  _14260
  _14615
  dogs(_14260)
  |_14260|=5
  cat(_14615)
  chases(_14260,_14615)
--------------------------------------------------
Querying: card(1,5),
dogs(1),
cat(0),
chases(1,0),
yes
--------------------------------------------------
```

Querying differs slightly for Prolog structures of distributive plurals. When querying the database with

(4.18)
```
man(_00001).       element(_00001,_00006).        card(_00006,5).
man(_00002).       element(_00002,_00006).
man(_00003).       element(_00003,_00006).
man(_00004).       element(_00004,_00006).
man(_00005).       element(_00005,_00006).

sees(_11735, [_00006,_11735]) :- man(_11735), element(_11735,_12345),
                                 card(_12345,5).

cat([_00006,_11735]) :- man(_11735), element(_11735,_12345),
                        card(_12345,5).
```

PluralDRT checks to see that every predicate of `man` is instantiated to a different fact in the database. This insures that there are indeed five different men who see the cat.

Suppose the sentence *Five dogs each see a cat* is asserted into the database. Then, given *Do five dogs each see a cat?*, PluralDRT outputs:

(4.19) `Sentence: Do five dogs each see a cat?`

```
[do,five,dogs,each,see,a,cat,?]
--------------------------------------------------
Originally constructed DRS:
--------------------------------------------------
QUERY:
  _55450
  number(_55450,pl)
  gender(_55450,n)
  case(_55450,nom)
  dog(_55450)
  |_55450|=5

  For each x in _55450
    _55832
    sees(x,_55832)
    cat(_55832)
    case(_55832,acc)
    gender(_55832,n)
    number(_55832,sg)

--------------------------------------------------
Cleaned-up (simplified) DRS:
--------------------------------------------------
QUERY:
  _55450
  dog(_55450)
  |_55450|=5

  For each x in _55450
    _55832
    sees(x,_55832)
    cat(_55832)
--------------------------------------------------
```

```
Querying: dog(2),
dog(3),
dog(4),
dog(5),
dog(6),
element(2,7),
card(7,5),
element(3,7),
card(7,5),
element(4,7),
card(7,5),
element(5,7),
card(7,5),
element(6,7),
card(7,5),
sees(_59548,[8,_59548]):-dog(_59548),
                          element(_59548,7),
                          card(7,5),
cat([8,_59548]):-dog(_59548),element(_59548,7),
                 card(7,5),
yes
---------------------------------------------------
```

Simple Inferences

PluralDRT may be queried based on simple inferences made about discourses asserted in the database. These inferences may be done on distributive and generic sentences. For example, suppose the following discourse is asserted into the database:

(4.20) *Cats are boojums. Blackie is a cat.*

asserted as:

```
boojum(_11735) :- cat(_11735).
named(1,'Blackie').
cat(1).
```

Then given the question *Is Blackie a boojum?*, PluralDRT will query the database and return a positive answer. Given a distributive sentence such as *Five men each see a cat*, PluralDRT will infer that one man saw a cat, two men each saw a cat, ... and will return a positive answer when given them as questions. PluralDRT does this by querying the database with the clauses for *Two men each see a cat* with the exception of `card`. Instead of querying `card`, PluralDRT checks to see if there is another `card`. If one is found, the second argument of that `card` is checked to see if it is greater than two. PluralDRT succeeds if the second argument is greater than two and fails if it is not.

PluralDRT will not do this for collective readings. This is because saying *A man bought a pizza* from the assertion *Five men bought a pizza* is misleading information.

4.4 What PluralDRT Can and Cannot Do

PluralDRT has restrictions placed on what it understands and limitations of what it can do. The primary goal of PluralDRT is to handle cases of collective and distributive readings, plural anaphora, and generics. In order to stay focussed on this goal and not get bogged down in the complexities of plurality, these restrictions and limitations are necessary.

4.4.1 Collective and Distributive

PluralDRT handles collective and distributive readings of plural sentences with cardinal determiners between 1–10. Cardinal determiners are a lot less complicated to deal with than other determiners which do not have a strict cardinality associated with them. An example plural sentence to PluralDRT might be:

(4.21) *Five men buy a cat.*

In order to successfully assert discourse into the database or query it, a restriction with regards to ambiguity is made. PluralDRT assumes that the subject is meant to have a collective reading unless the floating quantifier *each* is between the subject and the predicate. For example, PluralDRT will assume

Five men see a cat

is a collective reading, and

Five men each see a cat

is a distributive reading of the same sentence. Further, objects of the verb are assumed to always have a collective reading. These are, of course, not true in natural language, but for the small domain of PluralDRT, it works well.

4.4.2 Anaphora

While PluralDRT easily handles anaphora referring to collective plurals by treating them as it would anaphora referring to singular terms, there are anaphoric types which PluralDRT is unable to handle. PluralDRT, for example, cannot at this point completely handle anaphoric subordination.[1] It can do everything up to translating the DRSs into Prolog structures and asserting them into the database. Further, PluralDRT has a limitation of not being able to handle plural anaphors that refer to implicit sets. For example, PluralDRT cannot handle the following discourse:

(4.22) *Bill, Bob, and Dave order a pizza. They eat it.*

4.4.3 Definite Descriptions

Plural definite descriptions are assumed by PluralDRT to refer back to previously mentioned noun phrases, and hence treats them as anaphors. This is because definite descriptions taken referringly often need extra-linguistic information such as *The man standing right there*, or are proper names as in *The Queen of England.* Also, it is my observation that plural definite descriptions—unlike singular ones—seem to be used more often as anaphors.

PluralDRT does not handle dependent plural definite descriptions. This is because they are vague sets which are contingent on the subject. This is rather difficult to implement, and there was a time restriction.

[1] PluralDRT makes a further assumption that Anaphors referring to distributive sentences are the sloppy reading.

Chapter 5

Further Research

PluralDRT is not a finished project. There are many ways in which it can be extended or used as a basis for future projects. PluralDRT is meant to be a first attempt at demonstrating two readings of plurality and related anaphora. Due to this focus, the program is limited and thus has a large capacity for improvement in both the handling of plurality—and linguistics in general—and the features of the program. Further, many connected ongoing projects may be integrated with PluralDRT or use PluralDRT to build on.

5.1 Extending PluralDRT

5.1.1 Linguistic Extensions

PluralDRT should first be extended to handle the cumulative readings. It does not include all cases even though PluralDRT's focus was the handling of different readings in plurality. The cumulative reading is especially important because both the collective and distributive readings seem to be special cases of this reading. So rather than implementing different schemes for different readings, simply use one overall scheme to deal with all readings.

Other extensions might be the adding of more determiners, included in these are generalized quantifiers. The cardinal determiners were used in PluralDRT because they are very straightforward. However they are by no means the only determiners. These determiners include *some*, *both*, *most*, *few*, and *many*.

Modal subordination could be implemented into PluralDRT in two ways. The first would be to treat it similarly to the treatment of anaphoric subordination. That is, create a second duplex condition which copies the information of the first which is inaccessible with the additional information. The second is to use Roberts' insertion approach by sticking the new information in the right hand side of the duplex condition. As stated previously, I believe Roberts' solution to be the better one for both anaphoric subordination and modal subordination.

In the last chapter, I mentioned several areas in which PluralDRT was restricted or limited. Extensions might include adding these areas. However, it would remain important not to lose the focus of PluralDRT by getting caught up in the complexities of plurality.

5.2 Programming Extensions

The program itself can be extended in many ways. Covington and Schmitz (1988) and Covington, Nute, Schmitz, and Goodman (1988) noticed several. One such addition would be a table of identity so that when individuals are introduced into the discourse under various names, the different names can be linked to the same individual. For example, the following discourse should succeed but doesn't.

(5.1) *If a dog chases a cat, the dog is happy. Chester is a dog. Chester chases a cat. Is Chester happy?*

```
Sentence: If a dog chases a cat then the dog is happy.  Chester is a
  dog.  Chester chases a cat.  Is Chester happy?

[if,a,dog,chases,a,cat,then,the,dog,is,happy,.,
 chester,is,a,dog,.,chester,chases,a,cat,.,is,chester,happy,?]
--------------------------------------------------
Originally constructed DRS:
--------------------------------------------------
_4168
_4172
_4632
_4881
IF:
  _3346
  _3621
```

```
  number(_3346,sg)
  gender(_3346,n)
  case(_3346,nom)
  dog(_3346)
  number(_3621,sg)
  gender(_3621,n)
  case(_3621,acc)
  cat(_3621)
  chases(_3346,_3621)
THEN:
  happy(_3346)
case(_4168,nom)
number(_4168,sg)
gender(_4168,n)
named(_4168,Chester)
number(_4172,sg)
gender(_4172,n)
case(_4172,acc)
dog(_4172)
_4168=_4172
case(_4632,nom)
number(_4632,sg)
gender(_4632,n)
named(_4632,Chester)
number(_4881,sg)
gender(_4881,n)
case(_4881,acc)
cat(_4881)
chases(_4632,_4881)
QUERY:
  _5365
  case(_5365,nom)
  number(_5365,sg)
  gender(_5365,n)
  named(_5365,Chester)
  happy(_5365)

--------------------------------------------------
Cleaned-up (simplified) DRS:
```

```
--------------------------------------------------
_4168
_4168
_4632
_4881
IF:
  _3346
  _3621
  dog(_3346)
  cat(_3621)
  chases(_3346,_3621)
THEN:
  happy(_3346)
named(_4168,Chester)
dog(_4168)
named(_4632,Chester)
cat(_4881)
chases(_4632,_4881)
QUERY:
  _5365
  named(_5365,Chester)
  happy(_5365)
--------------------------------------------------
Asserting: happy(_3346):-chases(_3346,_3621),cat(_3621),dog(_3346)
Asserting: named(2,Chester)
Asserting: dog(2)
Asserting: named(1,Chester)
Asserting: cat(0)
Asserting: chases(1,0)
Querying: named(2,Chester),
happy(2),
named(1,Chester),
happy(1),
chases(1,0),
cat(0),
cat(0),
chases(1,0),
cat(0),
cat(0),
```

```
named(2,Chester),
happy(2),
named(1,Chester),
happy(1),
chases(1,0),
cat(0),
cat(0),
chases(1,0),
cat(0),
cat(0),
no
--------------------------------------------------
```

The query fails because the representations for the individual *Chester* in the second and third sentences are different.

PluralDRT currently does not handle negation. An extension to include negation would need to deal with a quirk of Prolog. The inference engine views *cat* and ∼*cat* as two separate facts, which is not the case in natural language.

Loops are created in PluralDRT by sentences which use noun phrases twice. For example:

(5.2) *Every long-tailed cat is a skittish cat.*

PluralDRT would assert such a sentence as

```
cat(X) :- cat(X), long_tailed(X).
skittish(X) :- cat(X), long_tailed(X).
```

Covington, Nute, Schmitz, and Goodman suggest simply adding a syntactic readjustment rule to remove such loops.

As mentioned before, anaphoric subordination is dealt with partially. It is handled linguistically, but PluralDRT fails to translate the DRSs to Prolog. In order to handle this, PluralDRT should treat discourses with anaphoric subordination as one condition rather than two. In the discourse

Five men each see a cat. They feed them.

cat is asserted as:

```
cat([6,_11735]) :- man(_11735),element(_11735,_11856),card(_11856,5).
```

Similarly, `feeds` should be asserted as:

```
feeds(_11735,[6,_11735]) :- man(_11735),element(_11735,_11856),card(_11856,5).
```

5.3 PluralDRT and Similar Projects

There have been several projects which can be integrated with PluralDRT. These projects include Brown (forthcoming) and Izzo (1993). Daniel Brown extends the Covington, Nute, Schmitz, and Goodman program to handle tense and aspect. PluralDRT and his project would integrate smoothly to create a larger program handling both plurality and tense and aspect. Gregory Izzo combines DRT and defeasible reasoning in a program which takes natural language and translates it into Prolog statements. Since PluralDRT and the Brown and Izzo Projects are based on the same program, combining PluralDRT with the Brown and Izzo projects would form a larger program which allows for better handling of English sentences in general.

Appendix A

User's Guide to PluralDRT

A.1 Loading PluralDRT into Prolog

PluralDRT is written in Quintus Prolog 3.1.4 and uses an extension to Prolog called Gulp3. In order to use PluralDRT, first load Gulp3 into Prolog. Then load PluralDRT in by typing:

`['pluraldrt.pl'].`

To start PluralDRT, simply type `go.`

A.2 Types of Sentences Used in PluralDRT

The following types of sentences may be used in PluralDRT:

- $S \rightarrow N\ V\ N$
  where N is a bare plural.
- $S \rightarrow N$ *[are] Adj*
  where N is a bare plural.
- $S \rightarrow NP\ VP$
- $S \rightarrow NP$ *[does not]* VP
- $S \rightarrow NP$ *[each]* VP
  where *each* denotes a distributive sentence.
- $S \rightarrow NP$ *[each] [are] Adj*

- $S \rightarrow NP$ *[are] [each] Adj*
- $S \rightarrow NP$ *[is] Adj*
  where NP is singular.
- $S \rightarrow NP$ *[are] Adj*
  where NP is plural.
- $S \rightarrow NP$ *[is not] Adj*
- $S \rightarrow NP$ *[are not] Adj*
- $S \rightarrow$ *[a] N [is] [a] N*
  where N is singular.
- $S \rightarrow$ *N [are] N*
  where N is a bare plural.
- $S \rightarrow NP$ *[is] NP*
  where NP is singular.
- $S \rightarrow NP$ *[is not] NP*
  where NP is singular.
- $S \rightarrow NP$ *[are] NP*
  where NP is plural.
- $S \rightarrow NP$ *[are not] NP*
  where NP is plural.
- $S \rightarrow$ *[if] S [then] S*

A.3 Assumptions Made By PluralDRT

PluralDRT makes certain assumptions about the reading of a sentence. PluralDRT assumes that a sentence which does not have the floating quantifier *each* between the subject and predicate is a collective reading. If *each* is present, the sentence is assumed to have a distributive reading for the subject. The object of the verb is always assumed to have a collective reading.

PluralDRT considers sentences of the following form to be generic statements:

- *A cat is a boojum.*

- *Cats are boojums.*
- *Dogs chase cats.*

A.4 PluralDRT's Lexicon

A.4.1 Proper Names

Pedro, Chiquita, Maria, Blackie, Chester, Kitty, Bill, Bob, Dave, Heather, Sage, Dana, Emma, Fox.

A.4.2 Common Nouns

bandersnatch, bandersnatches, boojum, boojums, cat, cats, dog, dogs, man, men, woman, women, donkey, donkeys, farmer, farmers, knight, knights, lady, ladies, knave, knaves, pizza, pizzas.

A.4.3 Adjectives

big, green, rich, old, happy, sad, morose, nasty.

A.4.4 Transitive Verbs

sees, see, loves, love, owns, own, has, have, beats, beat, feeds, feed, admires, admire, fights, fight, insults, insult, chases, chase, buys, buy, eats, eat.

A.4.5 Intransitive Verbs

bark, barks, eats, eat, brays, bray.

A.4.6 Determiners

a, an, one, two, three, four, five, six, seven, eight, nine, every, no, not every.

A.4.7 Miscellaneous

both, are, is, if, then, who, whom, which, each.

Bibliography

[1] Barwise, Jon, and Cooper, Robin (1981) Generalized Quantifiers and Natural Language. *Linguistics and Philosophy.* 4:159-219.

[2] Bennett, Michael (1975) *Some Extensions of a Montague Fragment of English.* Dissertation, UCLA. Cited by: Landman, F. (1989)

[3] Brown, Daniel (1994) *DRT with Extended Verb Semantics.* Master's thesis, University of Georgia.

[4] Covington, Michael (1987) *GULP 1.1: An extension of Prolog for Unification-based Grammar.* ACMC Research Report 01-0021. University of Georgia.

[5] Covington, Michael (1993) *Brief Notes on GULP3.* An unpublished manuscript. University of Georgia.

[6] Covington, Michael (1994) *Representing Plurals as Collections in Logic Programming.* An unpublished manuscript. University of Georgia.

[7] Covington, Michael, Nute, Donald, Schmitz, Nora, and Goodman, David (1988) *From English to Prolog via Discourse Representation Theory.* ACMC Research Reports 01-0024, University of Georgia.

[8] Donnellan, Keith (1966) Reference and Definite Descriptions. *Philosophical Review.* 75:281-304.

[9] Franconi, Enrico (1993) A Treatment of Plurals and Plural Quantifications based on a Theory of Collections. *Minds and Machines.* 3:453-474.

[10] Goodman, David (1988) *An Implementation of an Extension to Discourse Representation Theory: Translating Natural Language to Discourse Representation Structures to Prolog Clauses.* Master's thesis, University of Georgia.

[11] Hoeksema, Jack (1983) Plurality and Conjunction. In ter Meulen, Alice, eds., *Studies in Modeltheoretic Semantics.* Groningen-Amsterdam Studies in Semantics (GRASS) 1. Dordrecht:Foris Publications.

[12] Kamp, Hans (1981) A Theory of Truth and Semantic Representation. in: Groenendijk, J. A. G., Janssen, T. M. V., and Stokhof, M. B. J. eds. *Formal Methods in the Study of Language.* Amsterdam: Mathematical Centre Tract. Found in the University of Georgia's Artificial Intelligence Programs Library.

[13] Kamp, H. and Reyle, U. (1993) *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Dordrecht: Kluwer Academic Publishers.

[14] Kripke, Saul (1972) Naming and Necessity. In: Davidson, Donald, and Harman, Gilbert H. eds. *Semantics of Natural Language.* Dordrecht:Reidel.

[15] Landman, Fred (1989) Groups, I. *Linguistics and Philosophy.* 12:559-605.

[16] Link, Godehard (1983) The Logical Analysis of Plurals and Mass Terms: A Lattice-theoretical Approach. In Bäuerle, Rainer, Schwarze, Christoph, and von Stechow, Arnim, eds., *Meaning, Use, and Interpretation of Language.* Berlin:de Gruyter.

[17] Reinhart, Tanya (1983) Coreference and Bound Anaphora: A Restatement of the Anaphora Question. *Linguistics and Philosophy.* 6:47-88.

[18] Roberts, Craige (1987) *Modal Subordination, Anaphora, and Distributivity.* Ph.D. Thesis, University of Massachusetts at Amherst.

[19] Scha, Remko (1981) Distributive, Collective and Cumulative Quantification. In Groenendijk, J., Janssen, T., and Stokhof, M., eds., *Truth, Interpretation and Information.* Groningen-Amsterdam Studies in Semantics (GRASS) 2. Dordrecht:Foris Publications.

[20] Spencer-Smith, R. (1987) Semantics and Discourse Representation. *Mind and Language* 2:1:1-26.

[21] Webber, Bonnie (1983) So what can we talk about now? In Brady, M., and Berwick, R.C., eds., *Computational Models of Discourse.* 331-371.