Ambiguous Noun Phrases in Logical Form

Mary P. Harper*
Purdue University

In this paper, logical form representations for pronouns, singular definite noun phrases (NPs), and singular indefinite NPs are developed. These representations allow decisions about the precise meaning of a sentence to be postponed until the required information becomes available. Three computational constraints for this logical form are proposed: compactness, modularity, and formal consistency. Initially, NPs are represented using a composite representation for all allowable meanings, conforming with the compactness constraint. This representation is provided using only syntactic and sentence level information, consistent with the modularity constraint. When an ambiguity can be resolved, the precise behavior is specified in a way compatible with the initial representation, conforming with the formal consistency constraint. The scope of this approach is demonstrated by using a wide variety of examples, and a computer implementation is described. Related approaches are also discussed.

1. Introduction

A goal of natural language research is to provide a computer model capable of generating an internal representation for the meaning of each sentence processed. The building of this representation can be approached in two ways.

The first approach requires the generation of an unambiguous internal representation for each sentence before attempting to represent subsequent sentences. There are, however, several problems with this approach. First, the enumeration and testing of each possible reading of an ambiguous sentence to determine which is correct can require significant computational resources. Second, an application may not require the precise meaning for a sentence; hence, determining a single meaning would be a waste of resources. Third, many times an ambiguous sentence cannot be disambiguated until information contained in subsequent sentences has been processed.

Another approach, the one adopted in this paper, is to use an intermediate representation called **logical form** (LF) (Schubert and Pelletier 1984; Allen 1987; Harper 1988) to avoid immediately committing to a single meaning of an ambiguous sentence. LF partially specifies the meaning of a sentence based on syntactic and sentence-level information, without considering the effects of pragmatics and context. This partial specification of meaning allows us to process additional sentences before further limiting the meaning of an ambiguous sentence. Later, as relevant information becomes available (from a context processing module), the intermediate representation of the ambiguous sentence can be incrementally updated. This process can continue until all of the ambiguities are resolved and an unambiguous internal representation of the sentence is generated (although this is not a requirement of the approach). The process of mapping a sentence to an unambiguous internal representation is shown in Figure 1.

Because LF is a component of a computer model for language comprehension, it must be designed with two goals in mind. First, it should accurately model the

---

* School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

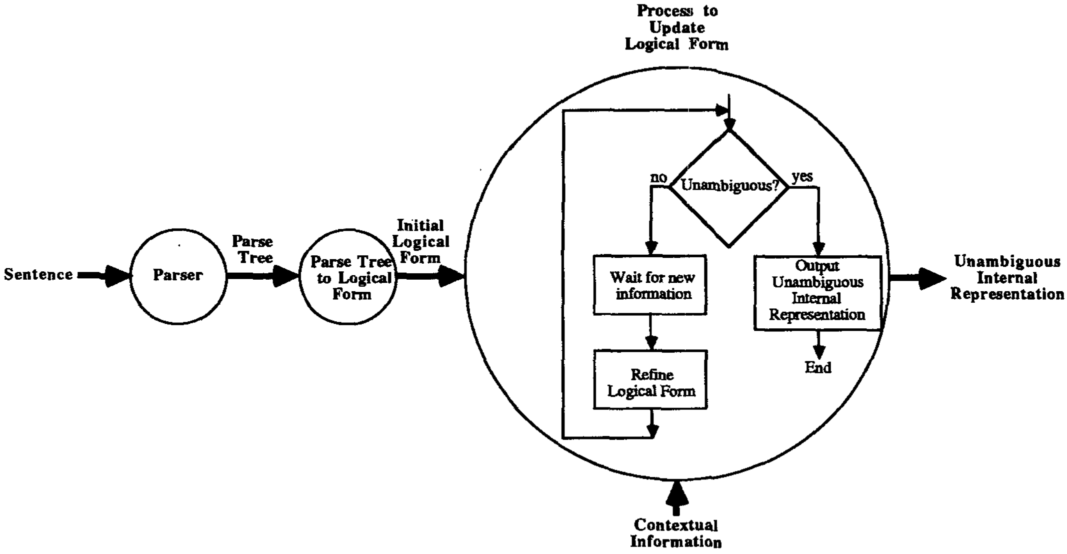

Figure 1
An architecture for language processing.

linguistic behavior of language. Second, it should be computationally tractable. In fact, we have defined three constraints for using LF in a computational framework (Harper 1988, 1990).

1. **Compactness Constraint:** The compactness constraint, which captures the spirit of LF as presented by Allen (1987), is important in a computer model of language comprehension because of the need to postpone decisions about ambiguity without large storage requirements. LF should compactly represent the underdetermined meaning of a sentence—underdetermined in that ambiguities and anaphora are not resolved. Ideally, LF should only require polynomial (in the number of words in a sentence) space.
2. **Modularity Constraint:** The modularity constraint requires LF to be initially computable from syntax and local (sentence-level) semantics only. In particular, it should not be initially dependent on contextual information, which requires significantly more computational resources to extract.
3. **Formal Consistency Constraint:** The formal consistency constraint requires that any update to the meaning of LF should be a refinement of the original meaning. Initially, LF provides a composite representation for a sentence. However, as more information becomes available, the meaning of the sentence is incrementally updated (without contradicting the composite meaning) until all ambiguity is resolved. To obey the formal consistency constraint, it is important to explicitly detail the syntax and semantics of the LF (see Appendix A for the syntax and semantics of the LF we are about to describe) and to indicate how formal consistency would be violated.

In the next section, we describe how LF has been used to represent quantifier scope ambiguity and how this representation conforms to our computational constraints. Then, we introduce our LF representations for pronouns, singular definite NPs, and singular indefinite NPs, considering how to update each representation when new information becomes available (though not how to determine that information). We also discuss an implementation that uses our representations and compare our approach with related work.

2. Quantifier Scope Ambiguity in Logical Form

Quantifier scope ambiguity has been handled by some researchers by using an intermediate scope-neutral LF (e.g., Hobbs 1983; Schubert and Pelletier 1984; Allen 1987) for the initial representation of sentences. Hence, sentences like *Someone loves everyone* are initially represented without committing to one particular meaning, as shown in Example 1.

Example 1
Someone loves everyone.
Possible Meanings: 1. $\exists x\ \forall y$ (love x y)
2. $\forall y\ \exists x$ (love x y)
Scope-neutral Form: (love [$\exists x$ x] [$\forall y$ y])

Initially, in scope-neutral LF, the quantifiers are stored in the predicate argument-structure with no scoping preference indicated, hence the representation does not commit to a specific meaning for the sentence; it is simply a compact way of expressing the set of all possible readings. Another type of scope-neutral form initially gives universal quantifiers scope over all existentials. This solution provides a general reading that in many cases subsumes readings where the existential is outside of the scope of the universal. However, if we are to handle NPs containing pronouns and sentences with verb phrase ellipsis, this solution is inadequate.

An LF that avoids committing to a single meaning must be capable of being updated once information is available to limit the scoping possibilities. There are a variety of mechanisms for expressing scoping decisions. For example, Schubert and Pelletier (1984) indicate quantifier scoping by extracting and linearly ordering the quantifiers to the left of the predicate-argument structure. Allen (1987) indicates for each pair of quantifiers which one has scope over the other; a method that is not limited to a linear sequence of operators. Finally, Creaney and McTear (1990) use constraints to limit scope possibilities. Hintikka (1979) has noted that the linear ordering of quantifiers is not sufficient to capture all possible meanings of a sentence when four or more quantifiers occur in the sentence. Consider a sentence with four quantifiers, two universals and two existentials: *Every boy$_i$ wanted every girl$_j$ to introduce a friend of his$_i$ to a friend of hers$_j$*. It should be possible for *every boy* to have scope over *a friend of his*, without having scope over *a friend of hers*. Similarly, *every girl* could have scope over *a friend of hers*, but not *a friend of his*. There is no way to express this with a linear quantifier scoping string, but Allen and Creaney and McTear have no trouble indicating nonlinear scoping using their approaches.

A scope-neutral LF compactly represents a sentence. Since the size of the scope-neutral representation is directly proportional to the length of the sentence, it is consistent with the compactness constraint. This representation also avoids committing to a single meaning by separating the process of making a scoping decision from the process of constructing the initial representation for a sentence and its constituents (this separation is consistent with the modularity constraint). Later, when enough information becomes available to make a scoping decision, there are several different methods for indicating which quantifier has scope over the other. These modifications commit to one of the meanings encoded in the scope-neutral form and so are compatible with the formal consistency constraint.

There are other types of semantic ambiguities in addition to quantifier scope ambiguity that can be handled by using LF. For example, the meanings of pronouns, singular definite NPs, and singular indefinite NPs often cannot be determined without additional contextual information. Postponing decisions about the precise meanings of sentences containing these types of constituents could be extremely useful, as the following example illustrates: *Every man showed a boy a picture of his mother*. The precise meaning of the sentence cannot be specified until information is available to select the pronoun's antecedent and to determine the quantifier scoping. The pronoun *his* can have many different antecedents, including: *every man, a boy,* some entity introduced in other sentences, or some individual in the environment of the speaker or hearer. The meaning of *his mother* is ambiguous because it depends on the pronoun's antecedent. Also, *a picture of his mother* is ambiguous; its meaning cannot be determined until we determine the meaning of *his mother* and decide whether the universally quantified NP has scope over the indefinite. Notice that quantifier scoping decisions are affected by the choice of antecedent for *his*. If *every man* is the antecedent then it must have scope over *a picture of his*.

3. Pronouns in Logical Form

Pronouns are a source of underspecification in a sentence: the antecedent of a pronoun cannot be determined using syntactic information alone, but requires a combination of syntactic, semantic, and contextual information. We divide the process of determining the meaning of a pronoun into two phases. First, we provide the pronoun's LF, using only syntactic and sentence-level information. This LF constrains the possible antecedents to be those NPs that are consistent with this local information. The LF is also a flag indicating that the sentence is underspecified because more than one antecedent for the pronoun is possible. Later, when the antecedent is determined, a task that often requires contextual information found in surrounding sentences, we provide a way to update our LF to include this information.

In the rest of this section, we discuss the linguistic behaviors of pronouns we want to model, introduce their LF representation, discuss how that representation is updated once contextual information isolates antecedents for the pronouns, and describe how the approach models verb phrase ellipsis (VPE).

3.1 Pronouns: Linguistic Behavior

Pronouns either have linguistic antecedents or refer to salient objects in the environment of the speaker or hearer (deictic use). Pronouns with linguistic antecedents can be categorized in two ways: either their antecedents occur in the same sentence (*intrasentential reference*[1]) or in other sentences (*intersentential reference*).

---

[1] By reference, we do not mean that the pronoun denotes its linguistic antecedent; rather, that it adopts the behavior of its antecedent.

When a pronoun's intrasentential antecedent is a universal NP, that pronoun takes on the behavior of the universal's variable. Consider the sentence *Fred showed every girl her picture.* Given that the antecedent for *her* is *every girl,* the pronoun adopts the behavior of a variable bound by the universal quantifier of its antecedent, as shown in Example 2.

Example 2

Fred$_i$ showed every girl$_j$ her$_j$ picture.
∀x (if (girl x) (show Fred (x's picture) x))

In contrast, if a pronoun has an intersentential antecedent, then that pronoun cannot act like a bound variable, since quantifiers do not have scope across sentences in English. Consider the following two sentences: *Fred likes everyone. But, he doesn't return the sentiment.* The pronoun in the second sentence cannot be bound by the quantifier corresponding to *everyone* in the first sentence. However, a quantified NP in one sentence can be the antecedent for a pronoun in another sentence, as in *Fred likes everyone. But, they don't return the sentiment.* In this case, *they* adopts the discourse entity for the group of individuals that *everyone* quantifies over. Webber (1978) discusses how to construct discourse entities for nonanaphoric NPs in a sentence (both for quantified NPs and definite NPs that are quantified over) once that sentence has been disambiguated. A discourse entity is a designator for the entity or set of entities the NP evokes in the discourse model of the speaker or hearer. If an NP in one sentence is the antecedent for a pronoun in another, the pronoun is replaced with the antecedent's discourse entity.

Pronouns have also been classified as bound variable pronouns or referential pronouns. The antecedent for a bound variable pronoun occurs in the same sentence as the pronoun and the meaning of the pronoun is represented as a variable bound by the operator associated with its antecedent. In contrast, the meaning of a referential pronoun is the discourse entity evoked by its antecedent. The bound versus referential dichotomy divides the world of pronouns differently than does the intrasentential–intersentential dichotomy. Pronouns with intersentential antecedents are typically referential.[2] However, pronouns with intrasentential antecedents can be bound or referential (Webber 1978; Reinhart 1983).

The bound–referential dichotomy doesn't cover the entire range of behaviors possible for pronouns. There is another category of pronoun that Evans (1980) dubs *E-type pronouns* and that appears to be a bound variable, but on closer inspection is not. Donkey sentences (originally noticed by Geach [1962]) can be used to demonstrate this difficulty, for example: *Every miner who owns a donkey beats it.* Given that *every miner* has scope over *a donkey,* the indefinite cannot be referential and the existential operator is blocked from binding the pronoun because of the scope island; quantified NPs embedded in a relative clause attached to an NP cannot bind pronouns outside of the relative clause environment.[3] However, *a donkey* can be the antecedent for the pronoun.

---

[2] The only exception are pronouns like those in *paycheck sentences* (first noticed by Kartunnen [1969]). Consider the sentence *Fred gave his paycheck to his wife. George gave it to his mistress.* The pronoun *it* is not referential. For that matter, it is not bound. The pronoun seems to take *his paycheck* as its antecedent where the pronoun is instantiated to a different individual than in the original sentence.

[3] This is related to the Complex NP Constraint introduced by Ross (1967), which prevents wh-movement out of a relative clause attached to an NP.

Donkey sentences provide evidence that all of the following cannot be simultaneously true (adapted from Heim [1982, p. 102]).

1. Indefinites should be represented using existential quantifiers.
2. Indefinites obey the same scope-island restriction as universals.
3. Pronouns are either bound variables or referential.

Many researchers have attacked one or more of these assumptions, but we prefer to modify the third by adding an additional type of pronoun: pronouns that adopt the functional behavior of their antecedents. The first two assumptions, together with our modification of the third, allows us to build a simpler LF for a sentence and also handle donkey sentences (as described in Section 5.2).

Pronouns are a source of ambiguity in verb phrase ellipsis (VPE). To signal a VPE, a full verb phrase (VP) is replaced with an auxiliary, as in the second sentence of Example 3. A sentence with VPE is called an **elided sentence**. The index on *Fred* and *his* indicates that they are co-referential.

Example 3

Trigger Sentence:	Fred$_i$ loves his$_i$ wife.
Elided Sentence:	George$_j$ does too.
Possible Meanings:	1. George loves Fred's wife. (strict meaning)
	2. George loves George's wife. (sloppy meaning)

The elided sentence has little meaning independent of the first sentence, called a **trigger sentence**. Hence, before determining the meaning of the elided sentence, the meaning of the trigger sentence must be completely determined. Even though the antecedent of the pronoun in the trigger sentence is *Fred*, the meaning of that pronoun is still ambiguous because pronouns in a trigger VP can refer to a subject NP either directly or indirectly. This example indicates that care is needed to design a pronoun representation capable of handling VPE since the representation must be compatible with the two behaviors of a pronoun whose antecedent is the syntactic subject of a trigger sentence. It also demonstrates how the meaning assigned to the trigger VP limits the possible meanings of the elided sentence. Though the meaning of the elided sentence is ambiguous, it cannot mean *George loves some other person's wife* (other than Fred's or George's).

3.2 Pronouns: The Initial Representation

Before introducing our LF for pronouns, we briefly describe the LF for the rest of a sentence. A sentence is represented as a predicate-argument structure, with subjects lambda abstracted to handle VPE (following Sag [1976]; Williams [1977]; Webber [1978]; and Partee and Bach [1981]). By lambda abstracting syntactic subjects in LF, a pronoun whose antecedent is a syntactic subject can refer to that subject in two different ways, either directly by using a value depending on the type of the subject NP or indirectly by using the subject's lambda variable. The logical roles of all NPs in a sentence are indicated by position in LF (logical subject first, logical object second, logical indirect object third, etc.). Following Webber [1978], we represent universal NPs as universally quantified and restricted variables (as in 4a) and existentially quantified NPs as existentially quantified and restricted variables (as in 4b). The colon between the quantifier and its restriction expands differently depending on the type of the quantifier.

Example 4

a. Sentence: Every man is happy.
   Representation: ∀x: (man x) (happy x)
   Meaning: ∀x (if (man x) (happy x))
b. Sentence: A man is happy.
   Representation: ∃x: (man x) (happy x)
   Meaning: ∃x (and (man x) (happy x))

Quantifier scoping is handled in the same way as in Section 2. Initially, quantifiers are placed in the predicate-argument structure for the sentence, except for subjects, which are necessarily abstracted. Abstraction of a quantified subject does not imply that it must have scope over quantifiers placed in the lambda function corresponding to the VP. Later, when information becomes available for making scoping decisions, quantifier scoping is indicated using a method similar to Allen's (1987) (described in Section 5.2). Possessive NPs are represented as functions of the possessive nouns (following Webber [1978]) and proper nouns as skolem constants (i.e., skolem functions without arguments). These representations will be replaced with a general representation for definite NPs in Section 4.

The LF representation for a pronoun must be compatible with our computational constraints. To be consistent with the modularity constraint, a pronoun's representation must be generated without utilizing the contextual information needed to select its antecedent. To obey the compactness and formal consistency constraints, a pronoun must be represented using a single representation that is consistent with the ways the pronoun can act given its position in a sentence. To conform with these constraints, we represent a pronoun as a **pronoun function** in LF. This representation of pronouns is similar in spirit to the representation of pronouns as unique skolem constants in Charniak and McDermott (1985). Their representation allows the construction of basic logical structure of the sentence to precede pronoun resolution, a division consistent with our compactness and modularity constraints. However, because a constant is incompatible with a variable, their pronoun representation is incompatible with a bound variable meaning.

A pronoun function is a composite representation reminiscent of a skolem function. Its role is to limit the range of possible antecedents for the pronoun without committing to one in particular. Each pronoun function is assigned a unique name (supplied by adding a unique number to the pronoun to distinguish it from other pronouns), and its argument list is specified using only syntax and sentence-level semantics to avoid violating the modularity constraint.

A pronoun should be represented as a function of all the variables corresponding to quantified NPs that can affect its meaning (because they are representations for possible antecedents or can affect the meanings of other nonquantified antecedents such as definite NPs). By concentrating on variables of possible antecedents, we automatically include those variables that affect potential nonquantified antecedents. The argument list should also contain the variables of lambda operators that have scope over the position that the pronoun function fills in LF (in order to allow sloppy readings of elided sentences).

To provide an algorithm for automatically generating the LF for a pronoun, we must develop a mechanism for specifying its argument list. This mechanism should not automatically assign all of the variables associated with quantified NPs in the sentence to the argument list of a pronoun function because, in English, some quantified NPs are syntactically incompatible antecedents for the pronoun. Consider, for example, *He loves every man,* in which the antecedent for *he* cannot be *every man.* To determine

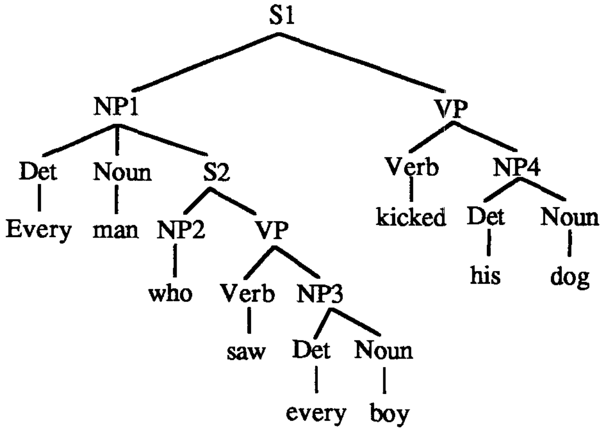

Figure 2
Parse tree for *Every man who saw every boy kicked his dog.*

which NPs can bind a pronoun in a sentence, we adapt Reinhart's (1983) c-command (or constituent-command), which is a relation on nodes in the sentence's parse tree.

> Node A c(constituent)-commands node B iff the branching node α_1 most immediately dominating A either dominates B or is immediately dominated by a node α_2 which dominates B, and α_2 is of the same category type as α_1. (Reinhart 1983, p. 23)

To illustrate the concept of c-command, consider the parse tree for the sentence in Example 5, shown in Figure 2.

Example 5
Every man who saw every boy kicked his dog.

In Figure 2, NP1 c-commands *his,* but NP2 and NP3 do not. Reinhart claims that a pronoun can be bound by an NP if and only if the NP c-commands the pronoun. Hence, *every man who saw every boy* can bind *his,* but *every boy* cannot.

C-command is very useful for ruling out incorrect antecedents for pronouns, as illustrated by Example 5. Unfortunately, it also makes some incorrect predictions. For example, it does not allow *every man's* to bind *him* in *Every man's mother loves him* or *each candidate* to bind *him* in *A friend of each candidate supported him,* even though the universal variables provide reasonable meanings for the pronouns. The difference between Example 5 and the previous two sentences is that in 5 the quantified NP in question is contained in a relative clause attached to an NP, whereas in the other two sentences they are not. A quantified NP is prevented from binding any pronoun outside of the clause even though the NP containing it c-commands the pronoun. Hence, we adapt Reinhart's binding rule[4] to allow quantified NPs to bind a pronoun if they c-command the pronoun or are embedded in another c-commanding NP but not contained in a relative clause.

---

[4] We could have also adapted the binding constraints found in Bach and Partee (1980).

Armed with this syntactic rule for determining which quantified NPs can have an impact on the meaning of a pronoun, we can now specify a pronoun's representation. A pronoun is represented as a uniquely named function of all the variables corresponding to operators that can have scope over it. These operators include:

- any lambda operator that has scope over the pronoun function in LF,
- any nonsubject quantified variable corresponding to an NP that can bind the pronoun given our modification of Reinhart's binding rule.

Because a quantified subject's variable is abstracted by the sentence's lambda operator, the lambda variable subsumes the quantified variable as the only subject-related variable required in the argument list. The lambda variable makes available both the direct quantified variable meaning as well as the indirect lambda variable meaning since we can choose to apply the quantified variable or not, depending on our need.

Now that the initial representation for pronouns has been specified, consider a series of examples, beginning with Example 6:

Example 6
Fred loves himself.
Fred_{22}, $\lambda(x)(\text{love } x\ (\text{himself}_1\ x))$

The subject *Fred* is represented as the skolem constant $\textit{Fred}_{22}$ and the VP as the lambda function, $\lambda(x)(\textit{love } x\ (\textit{himself}_1\ x))$. To create a representation for the entire sentence, we apply the subject to the lambda function; this is indicated by placing the subject to the left of the function. Though it is more traditional to apply the term to the right of the function, we chose this method to make it easier to see the correspondence between the sentence and its representation (following Sag [1976]). The logical subject fills the first slot after the predicate in the VP and the logical object fills the second. Since the sentence in Example 6 contains no universal or indefinite NPs, *himself* is represented simply as a pronoun function of the lambda variable x. And as we mentioned previously, the name of this function is created by concatenating a unique integer onto the end of the pronoun string (i.e., $\textit{himself}_1$).

In the next example, the representation of the pronoun is affected by universal and indefinite NPs in the sentence.

Example 7
Every man showed a boy his picture.
$\forall x\text{: (man } x)\ x,\ \lambda(y)(\text{show } y\ (\text{picture-of } (\text{his}_2\ y\ z))\ [\exists z\text{: (boy } z)\ z])$

The syntactic subject of the sentence is universally quantified, the indirect object is existentially quantified, and the logical direct object *his picture* is represented as a function of the pronoun. As discussed above, the subject's lambda variable subsumes the universal variable; hence *his* is represented as a function of the subject's lambda variable plus the existential variable for *a boy*. The reader should note that there is no order implied by quantifiers in our LF (following Allen [1987]) and so the quantifiers can be ordered in two different ways to provide two possible final meanings for the sentence.

Next, consider Example 8.

Example 8

Fred believes he must speak to every woman.

Fred$_{22}$, λ(x)(believe x [(he$_1$ x), λ(z)(speak z (to [∀y: (woman y) y]))])

Although this sentence contains a universal NP, *he* is represented as a pronoun function of the lambda variable x alone. The pronoun function's argument list does not include the variable y because *every woman* neither c-commands nor is embedded in an NP that c-commands the pronoun.

Finally, consider a sentence containing only pronouns and definite NPs.

Example 9

Fred showed his mother her picture.

Fred$_{22}$, λ(x)(show x (picture-of (her$_1$ x)) (mother-of (his$_2$ x)))

Both of the pronouns are represented as functions of the lambda variable x.

3.3 Pronouns: Updating Logical Form

When a pronoun's antecedent is known,[5] the LF containing it must be updated in a way compatible with its initial representation (to conform with the formal consistency constraint). To augment LF with antecedent information without creating an ill-formed LF (i.e., a LF with unbound variables), we assert equality statements in the lambda environment containing the pronoun function and limit the types of updates allowed. A pronoun function constrains its possible antecedents, and depending on the type and location of its antecedent, the pronoun function is equated with four different values.

Pronoun Update Cases:

1. If a universal or indefinite NP is the intrasentential antecedent for the pronoun, and its variable is an argument of the pronoun function (or is abstracted by a lambda operator whose variable is an argument of the function), then equate the pronoun function with that quantified variable.
2. If an NP represented as a function (i.e., a pronoun or definite) is the intrasentential antecedent for a pronoun, and its argument list is compatible with that of the pronoun function (either immediately or after more is known about its meaning), then equate the pronoun function with that function.
3. If a syntactic subject of a sentence is the intrasentential antecedent for the pronoun, and the lambda variable is an argument of the pronoun function, then equate the pronoun function with either the subject's lambda variable or some other value depending on the subject's type.
4. If an NP in a different sentence or some nonlinguistic entity is the antecedent for the pronoun (and it is compatible in number and gender with the pronoun), then equate the pronoun function with the discourse entity created for the antecedent.

---

[5] In this paper, we are not concerned with how the correct antecedent for a pronoun is determined. However, our LF provides useful hooks for an implementation to request information of either a user or a pronoun resolution module in order to resolve ambiguity.

Some updates to logical form must be postponed until more is known about the meaning of a possible antecedent (as we will show in Section 4.3.2).

Consider some examples of how LF is refined following pronoun resolution. Suppose, for example, that we decide that the antecedent for *his,* in Example 7, is *a boy,* then the LF for the sentence is modified as follows:

Example 10
Every man$_i$ showed a boy$_j$ his$_j$ picture.
$\forall$x: (man x) x, λ(y)(and (show y (picture-of (his$_2$ y z)) [$\exists$z: (boy z) z])
(= (his$_2$ y z) z))

Simplification:
$\forall$x: (man x) x, λ(y)(show y (picture-of z) [$\exists$z: (boy z) z])

To indicate that the antecedent for *his* is *a boy,* the pronoun function *(his$_2$ y z)* is equated with the existentially quantified variable *z* (by pronoun update case 1). Notice that the equality statement is placed in the environment of the $\lambda(y)$ operator. If we had placed it outside of this environment, the variable *y* would have been unbound. The variable *z* is compatible with the pronoun's initial representation because we are limiting the function of *y* and *z* to be the identity function on *z*. Once the equality statement is asserted, we can simplify the LF as shown above. Notice that the meaning of the sentence is still underspecified since quantifier scoping has not been determined.

Next, consider how the representation in Example 6 is augmented, given that the antecedent for *himself* is the subject *Fred.*

Example 11
Fred$_i$ loves himself$_i$.
Fred$_{22}$, λ(x)(and (love x (himself$_1$ x))
(or$_a$ (= (himself$_1$ x) x) (= (himself$_1$ x) Fred$_{22}$)))

The pronoun can refer to the subject either directly or indirectly, so its function is equated with *Fred*$_{22}$ (by pronoun update case 2) or *x* (by case 3), respectively. Notice that we use *or*$_a$ as a meta-or to annotate our logical form with multiple possible meanings for the pronoun (outside of the logical system for mapping to truth value, see Appendix A). No simplification of the LF is possible until one of the alternatives is chosen, but by using this device we can at least compactly represent the ambiguous ways that pronouns refer to syntactic subjects. In fact, if there are *n* pronouns whose antecedents are subjects, we can specify this ambiguity with O(n) updates, compared with providing 2^n different representations for the sentence. This allows us to more easily put off decisions about the pronoun's intended meaning until we process the required information, in contrast to an approach that uses an exponential amount of memory.

Finally, consider how the LF in Example 9 is updated given that *Fred* is the antecedent of *his* and *his mother* is the antecedent for *her.*

Example 12
Fred$_i$ showed (his$_i$ mother)$_j$ her$_j$ picture.
Fred$_{22}$, λ(x)(and (show x (picture-of (her$_1$ x)) (mother-of (his$_2$ x)))
(or$_a$ (= (his$_2$ x) x) (= (his$_2$ x) Fred$_{22}$))
(= (her$_1$ x) (mother-of (his$_2$ x))))

Possible Meanings:
1. Fred_{22}, λ(x) (show x (picture-of (mother-of x)) (mother-of x))
2. Fred_{22}, λ(x) (show x (picture-of (mother-of Fred_{22})) (mother-of Fred_{22}))

Note that *(his_2 x)* is equated with *Fred_{22}* (by case 2) or *x* (by case 3) since it is compatible with those values. Also *(her_1 x)* is equated with *(mother-of (his_2 x))* (by case 2) since the functions are compatible.

3.4 Pronouns and Verb Phrase Ellipsis

In this section, we discuss how to handle VPE, using Example 3 to illustrate our approach.

Example 3

Trigger Sentence:	Fred_i loves his_i wife.	
Elided Sentence:	George_j does too.	
Possible Meanings:	1. George loves Fred's wife.	(strict meaning)
	2. George loves George's wife.	(sloppy meaning)

The trigger sentence in Example 3 is initially represented as shown below.

Example 13
Fred loves his wife.
Fred_{22}, λ(x)(love x (wife-of (his_1 x)))

Because the meaning of an elided VP is constrained by the meaning of the trigger VP, the trigger's meaning must be fixed before we provide the meaning of the elided VP. In particular, we must locate the antecedent for *his*; otherwise, the final meaning of the pronoun function in the trigger cannot limit the meaning of the pronoun in the elided sentence. Given that the antecedent for *his* is *Fred*, we augment the LF as shown in 14.

Example 14
Fred_i loves his_i wife.
Fred_{22}, λ(x)(and (love x (wife-of (his_1 x)))
(or$_a$ (= (his_1 x) x) (= (his_1 x) Fred_{22})))

The initial representation for the elided sentence from Example 3 contains a place holder for the missing VP, as shown below.

Example 15
George does too.
George_{35}, λ(y)(Dummy_2 y)

To determine the intended meaning of the elided sentence, we must locate the trigger sentence, select a single meaning for that sentence, and replace the dummy lambda function with the lambda function representing the trigger VP. The trigger sentence's VP, shown in Example 14, contains a meta-or of equality statements signalling an ambiguity in the pronoun's meaning; hence, before providing the meaning of the elided VP, we must select a single meaning for the pronoun.

The procedure for choosing between the two meanings is beyond the scope of this work; consequently, we will demonstrate that for each choice, we are able to provide a reasonable meaning for the elided sentence. If the pronoun *his* refers indirectly to the subject of the trigger sentence, the first disjunct in 14 is selected. This choice provides the sloppy reading of the elided sentence, as shown below.

Example 16
Trigger Sentence Representation:
Fred_{22}, λ(x)(and (love x (wife-of (his_1 x))) (= (his_1 x) x))
Elided Sentence Representation:
George_{35}, λ(x)(and (love x (wife-of (his_1 x))) (= (his_1 x) x))

On the other hand, if the pronoun *his* refers directly to *Fred*, the second disjunct is chosen, giving the strict reading of the elided sentence.

Example 17
Trigger Sentence Representation:
Fred_{22}, λ(x)(and (love x (wife-of (his_1 x))) (= (his_1 x) Fred_{22}))
Elided Sentence Representation:
George_{35}, λ(x)(and (love x (wife-of (his_1 x))) (= (his_1 x) Fred_{22}))

There is no way to determine whether a particular sentence will be a trigger sentence until an elided sentence is processed, so all sentences are treated as potential trigger sentences. Later, when an elided sentence is detected, its trigger sentence must be located and disambiguated. Contextual information is needed to select trigger sentences, to choose antecedents for pronouns, and to select a single meaning for a pronoun function whose antecedent is a syntactic subject.

We have described our LF representation of pronouns, which allows us to model pronoun behaviors both in normal use and in VPE and is compatible with our computational constraints. Next, we discuss the LF for singular definite NPs, which is slightly more complex than the logical form for pronouns.

4. Singular Definite NPs in LF

In this section, we develop an LF representation for singular definite NPs. As with pronouns, we wish to obey our computational constraints while providing a model of definite behavior. First, we discuss the behaviors we wish to model.

4.1 Definites: Linguistic Behavior

Like pronouns, definite NPs can be anaphoric. Anaphoric definites either depend on linguistic antecedents or denote salient individuals in the environment of the speaker/hearer. An anaphoric definite's antecedent can be found in previous sentences as in *Fred$_i$ saw (his$_i$ cat)$_j$. The cat$_j$ was chasing a mouse*, or within the same sentence as in *Every boy$_i$ who loves (his$_i$ cat)$_j$ takes care of the animal$_j$*. In the first example, the antecedent for *the cat* is *his cat*, found in the previous sentence, hence, *the cat* adopts the discourse entity assigned to *his cat*. In the second, *the animal* cannot take a discourse entity as its meaning because its antecedent is *his cat*, which does not denote a particular cat.

Definites, unlike pronouns, can have a complex syntactic structure. A definite NP's meaning can be affected by embedded NPs. While simple nonanaphoric definites (i.e., they contain no embedded NPs) act like constants when included in sentences with universal NPs, as in *Every boy loves the woman*; definite NPs containing pronouns often cannot be described as constants, as in *Every boy loves his mother*. The meaning of *his mother* depends on how the pronoun is resolved. If the antecedent for *his* is found in another sentence, then *his mother* could be represented as a constant, but if *every* *boy* is the antecedent for *his*, then the universal quantifier corresponding to *every boy* distributes over *his mother*. When a quantifier distributes over a definite, the definite denotes different entities depending on the values assigned to the quantified variable. Any pronoun embedded in a definite NP can affect the definite's meaning: a possessive pronoun, one contained in a prepositional phrase (PP), or one contained in a relative clause attached to the definite.

A quantified possessive in a definite NP always distributes over the NP, preventing it from acting like a constant, as in *Every man's mother loves him*. It can also bind any pronouns the definite NP c-commands. Quantified NPs contained in a PP attached to a definite NP can also distribute over the definite, as in *The head of every public authority in New York is rich*, though the meaning of the definite NP is ambiguous. If the universal distributes over *the head of every public authority in New York*, then its denotation depends on which public authority is considered. But if the universal does not distribute over the definite, then there is one particular person who heads all of the public authorities. Our initial representation must be compatible with either possibility.

Not all embedded quantified NPs can distribute over a definite. Quantified NPs embedded in relative clauses attached to a definite NP are unable to distribute over the definite. This constraint prevents *every boy* from quantifying over *the child who cares for every man*, so the definite can only denote one particular child. Universal NPs that cannot distribute over a definite NP are also unable to bind a pronoun outside that phrase, as noted by May (1985) and Roberts (1987).

We must also consider the behavior of definite NPs in VPE. The meaning of a definite NP is ambiguous whenever it contains a pronoun whose antecedent is the subject of the sentence,[6] as in the following example.

Example 18
The postman$_i$ saw his$_i$ dog.
The policeman$_j$ did too.
Possible Meanings: 1. The policeman saw the postman's dog. (strict reading)
2. The policeman saw his own dog. (sloppy reading)

We must also provide a good representation for a definite subject, one that will account for the differences between universal and definite subjects in VPE (compare Example 19 with 18).

Example 19
Every postman$_i$ saw his$_i$ dog.
Every policeman$_j$ did too.
Possible Meanings: Every policeman saw his own dog. (sloppy reading only)

Universal quantifiers cannot bind across sentences, so the only possible meaning for the elided sentence in 19 is the sloppy one. However, definite subjects support both sloppy and strict meanings (as shown in 18). If we choose a quantified variable to represent a definite subject, we would have to allow its quantifier to bind across sentences.

The meaning of a definite NP is affected by its structure and the meanings of embedded NPs, as well as its potential anaphoric use. In the remainder of this section, we introduce our LF representation for definites, describe ways to update this

---

[6] Or whenever it contains an embedded indefinite. We consider sloppy indefinites in Section 5.3.

representation once ambiguity is resolved, and discuss how the representation is used in VPE.

4.2 Definite NPs: An Initial Representation

We represent definite NPs as functions of all of the variables that can affect their meanings. This representation satisfies our constraints by combining the advantages of definite descriptions (Russell 1971) (discussed in Section 7.2.1) with the functional notation we introduced to represent pronouns. A **definite function** is assigned a unique name (i.e., *def* with a unique integer subscript) to distinguish two occurrences of the same definite NP, has a restriction consisting of a single predicate or a conjunction of predicates derived from information contained in the NP, and has a (possibly empty) list of arguments containing:

- any variables associated with lambda operators that have scope over it,
- any variables associated with nonsubject quantified NPs that could bind a pronoun in that position,
- any quantified variables associated with embedded quantified NPs that are not also embedded in a relative clause.[7]

To illustrate our representation for definite NPs, consider the initial representation of the following sentence.

Example 20
Every man showed a boy his picture.
$\forall$x: (man x) x, λ(y)(show y ((def$_1$ y z) | (and (picture (def$_1$ y z))
(possess (his$_2$ y z) (def$_1$ y z))))
[$\exists$z: (boy z) z])

This representation is very similar to Example 7 except for the definite NP, *his picture,* which is represented as the function *(def$_1$ y z).* The definite function's argument list consists of the variables *y* and *z,* just like the pronoun *his.* As in the representation of pronouns, we omit the variable *x* from the argument list because the lambda operator for *y* abstracts *x,* so *y* is the more general argument. Anything that affects the meaning of the pronoun also affects the meaning of the definite. The function's restriction is the conjunction of statements following the vertical bar. The vertical bar in the function serves two purposes: it is used to distinguish the function's definition (on the left) from references to it (on the right), and it indicates that the function's restriction should be expanded just like the restriction on an existential operator.

This representation for definite NPs accounts for quantified NPs embedded in a definite. There are three cases to consider. The first is exemplified by *Every man's mother loves him.* The NP *every man's mother* does not denote a single mother; *every man* distributes over the definite noun phrase. A possessive quantified NP embedded in a definite NP always distributes over the definite. Hence, its variable must be included in the argument list of the definite function plus any other functions that the quantifier can affect (e.g., the pronoun function for *him*). Additionally, its quantifier must be moved to indicate that it has scope over the function, as shown below.

---

[7] We should also add that a sententially attached PP with a quantified object can quantify over a definite as well (e.g., *In every car, the driver turned the steering wheel,* in which the universal distributes over both definites).

Example 21
Every man's mother loves him.
∀x: (man x) ((def$_1$ x) | (and (mother (def$_1$ x)) (possess x (def$_1$ x)))), λ(y)(love y (him$_2$ x y))

The second case concerns quantified NPs contained in a relative clause attached to a definite NP. They cannot have scope over the definite so their variables are automatically excluded from the definite's argument list. Furthermore, they cannot affect the meaning of any other NP outside of the relative clause. The following example uses these facts to represent *The child who cares for every man visits him.*[8]

Example 22
The child who cares for every man visits him.
((def$_1$) | (and (child (def$_1$)) ((def$_1$), λ(x)(care x (for [∀y: (man y) y]))))), λ(z)(visit z (him$_2$ z))

The third case concerns quantified objects of PPs attached to a definite NP. These quantified NPs optionally have scope over the definite, and so our representation must be consistent with both possibilities. To avoid making decisions about whether a quantified object of a PP attached to a definite distributes over the definite, we include its variable in the function's argument list but leave the quantifier inside the function's restriction. This representation is shown below:

Example 23
The head of every public authority in New York is rich.
((def$_1$ x) | (head-of [∀x: (and (public-authority x) (in x New York)) x] (def$_1$ x))), λ(y)(rich y)
Possible Meanings:
1. ((def$_1$) | (head-of ∀x: (and (public-authority x) (in x New York)) x (def$_1$ x))), λ(y)(rich y)
2. ∀x: (and (public-authority x) (in x New York)) ((def$_1$ x) | (head-of x (def$_1$ x))), λ(y)(rich y))

Notice that the quantifier is placed inside the restriction of *def*$_1$, and the variable *x* is placed in the argument list (for the semantics of such a function, see Appendix A). Later, after we decide whether or not the quantifier distributes over the definite, the initial representation will be updated, as discussed in the next section.

The decision about whether a quantified object of a PP attached to a definite distributes over it cannot be made at the level of LF, though it has an impact on the quantifier's ability to bind pronouns (or anaphoric definites) in the sentence. Quantified objects of prepositions attached to a definite NP can bind pronouns in the sentence only when they distribute over the definite (Roberts 1987[9]; May 1985). For example, in *The secretary of every spy keeps an eye on him*, the NP *every spy* can bind the pronoun *him* only when it has scope over the definite NP, giving it a distributive reading. However, we cannot make our representation of the pronoun *him* contingent on quantifier scoping decisions. Hence, we must include the variable in the argument list of the pronoun, and add a constraint to the pronoun resolution module preventing a pronoun function from being bound by a quantifier unless it distributes over the NP containing it.

Because a definite function is initially a composite representation for all possible meanings of a definite NP, as appropriate information becomes available, we repeat-

---

[8] We do not provide an explicit representation for *who*; instead we represent it by borrowing the relative head's representation. In 22, *who* is represented as a definite function. If the relative head was quantified, we would have represented the relative pronoun using the quantified variable.

[9] Roberts modifies the definition of c-command to allow a PP-attached quantified NP to optionally c-command the same NPs as the containing NP.

edly update the function, refining its range of possible meanings. This process continues until there is no longer any ambiguity in the intended meaning of the definite. In the next section, we will discuss two methods for achieving this.

4.3 Definite NPs: Two Ways to Update the Initial Representation

4.3.1 Updating Anaphoric Definites. If a definite is used anaphorically, it is equated with some value depending on its antecedent (as in the case of the pronoun function). For example, if the antecedent for a definite noun phrase occurs in another sentence and they are compatible in number and gender, then we would equate the definite function with the antecedent's discourse entity. Otherwise, the update is not allowed.

Antecedents for definite NPs can also be found in the same sentence. For example, consider the initial representation of the following sentence.

Example 24
The owner of every dog is afraid of the animal.
((def$_1$ x) | (and (owner (def$_1$ x)) (of (def$_1$ x) [$\forall$x: (dog x) x]))),
λ(y)(afraid y (of ((def$_2$ x y) | (animal (def$_2$ x y)))))

Because the representation of *every dog* is formally consistent with the definite function and additional definite constraints hold (e.g., number and gender agreement holds, the antecedent does not c-command the anaphoric definite, and the universal quantifier distributes over the subject so it can bind the anaphoric definite), the definite function can be equated with the antecedent's representation, as shown in 25.

Example 25
(The owner of every dog$_i$)$_j$ is afraid of the animal$_i$
$\forall$x: (dog x) ((def$_1$ x) | (and (owner (def$_1$ x)) (of (def$_1$ x) x))),
λ(y)(and (afraid y (of ((def$_2$ x y) | (animal (def$_2$ x y))))) (= (def$_2$ x y) x)))

This example would be problematic for approaches using either definite descriptions or definite quantifiers, which provide no mechanism for handling bound variable, anaphoric definites.

4.3.2 Updating Structurally Complex Definites. To determine the meaning of a structurally complex definite NP (i.e., an NP containing embedded pronouns and quantified NPs), we must: 1) determine the meanings for all embedded NPs and 2) decide whether quantifiers corresponding to embedded quantified NPs not contained in relative clauses distribute over the definite. Given this information, we can refine the meaning of a definite function using the behavior of definite descriptions (Russell 1971) as our model. Any definite description that does not contain variables bound by outside quantifiers acts like a constant because of the uniqueness assumption (see Section 7.2.1 for a discussion of definite descriptions). On the other hand, if a quantifier has scope over the definite description (either because an embedded quantified NP distributes over it or the antecedent for an embedded pronoun is quantified), it denotes different individuals depending on the instantiation of that variable.

Once we determine the meanings of the NPs contained in a definite function's restriction and decide whether any extractable embedded quantifiers should distribute over it, we examine the function's restriction to determine whether it contains any **necessary arguments**, i.e., variables bound by operators outside of the restriction. When the necessary arguments for a definite function are determined, its meaning can be refined in two different ways. A definite function can be anaphoric only if it does not contain any necessary arguments. Hence, if a definite function contains no necessary arguments and is anaphoric, then it is equated with its antecedent (as in the previous section). However, if the definite function contains any necessary arguments or if it contains no necessary arguments but is nonanaphoric, then we limit the argument list to precisely the necessary arguments. By equating the original function with a new function over the necessary arguments, a process which we call **argument simplification**, we limit the initial composite representation of a nonanaphoric definite NP to its final meaning.

For example, consider the initial representation of the sentence in Example 20. Notice that *def*$_1$ is defined as a function of all of the variables that could potentially cause it to change. Given that the antecedent for *his* is *a boy*, the LF is updated as follows.

Example 26

Every man showed a boy$_i$ his$_i$ picture.
∀x: (man x) x, λ(y) (show y ((def$_1$ y z) | (and (picture (def$_1$ y))
(possess (his$_2$ y z) (def$_1$ y z))
(= (his$_2$ y z) z)))
[∃z: (boy z) z])

After *(his*$_2$ *y z)* is replaced with the variable z, the only necessary argument for *(def*$_1$ *y z)* is z. Since the restriction of the function is bound by an outside operator, the definite cannot be anaphoric. Hence, to provide the final meaning of the definite, we apply argument simplification to replace the function *(def*$_1$ *y z)* with a more precise function of z, as shown in Example 27. Because of the meanings of equality and the vertical bar in the restriction of the function, this representation is simplified, as shown below.

Example 27

Every man showed a boy$_i$ his$_i$ picture.
∀x: (man x) x, λ(y)(and (show y ((def$_1$ y z) | (and (picture (def$_1$ y z))
(possess (his$_2$ y z) (def$_1$ y z))
(= (his$_2$ y z) z)))
[∃z: (boy z) z])
(= (def$_1$ y z) (def$_3$ z)))
Simplification:
∀x: (man x) x, λ(y)(and (show y (def$_3$ z) [∃z: (boy z) z]) (picture (def$_3$ z)) (possess z (def$_3$ z)))

Also consider how we update the initial representation of the sentence in Example 23. If *every public authority in New York* distributes over the definite function, then the universal quantifier ∀*x* is extracted from the restriction prior to applying argument simplification, as shown below.

Example 28

The head of every public authority in New York is rich.
∀x: (and (public-authority x) (in x New York)) ((def$_1$ x) | (head-of x (def$_1$ x))), λ(y)(rich y)

Given this scoping decision, the variable *x* is free in the restriction of *(def*$_1$ *x)* and so the function must retain the argument. On the other hand, if *every public authority in New York* does not distribute over the function, then the quantifier remains in the restriction, as shown in 29.

Example 29

The head of every public authority in New York is rich.
((def$_1$ x) | (head-of [∀x: (and (public-authority x) (in x New York)) x] (def$_1$ x))), λ(y)(rich y)

Because the restriction of *(def$_1$ x)* contains no free variables, we must decide whether the definite is anaphoric or not. Assuming it is not, argument simplification is applied as shown below.

Example 30

The head of every public authority in New York is rich.
(and ((def$_1$ x) | (head-of [∀x: (and (public-authority x) (in x New York)) x] (def$_1$ x))),
λ(y)(rich y)
(= (def$_1$ x) (def$_2$)))
Simplification:
((def$_2$) | (head-of [∀x: (and (public-authority x) (in x New York)) x] (def$_2$))),
λ(y)(rich y)

Hence, we can systematically derive both of the possible meanings for the definite NP *the head of every public authority.*

The availability of a definite NP as an antecedent for a pronoun depends on its intended meaning, which cannot be determined using only syntactic information. Hence, c-command does not always correctly predict when definites are accessible as antecedents for anaphoric expressions. To determine the intended meaning of a definite, we must determine the meanings of all embedded NPs and decide whether any embedded quantified NPs distribute over the definite. Consider Example 31.

Example 31

Fred told the teacher who discussed every student with his mother to examine her educational history.
((def$_1$) | (name (def$_1$) Fred)),
λ(x)(tell x ((def$_2$ x) | (and (teacher (def$_2$ x))
(def$_2$ x), λ(y)(discuss y [∀(z): (student z) z]
(with ((def$_3$ x y z) |
(and (mother (def$_3$ x y z))
(possess (his$_4$ x y z)
(def$_3$ x y z)))))))
[(def$_2$ x), λ(w)(examine w ((def$_5$ x w) | (and (ed-history (def$_5$ x w))
(possess (her$_6$ x w) (def$_5$ x w)))))])

What are the legal antecedents for *her* in this sentence? Certainly, *the teacher* is a candidate, but consider *his mother.* We cannot immediately determine whether *his mother* is a legal antecedent for *her* because *(her$_6$ x w)* is not immediately compatible with the representation for *his mother* (i.e., *(def$_3$ x y z)*). We must first determine the meaning of *his mother* by selecting the antecedent for *his.* Depending on the outcome, the final meaning of *his mother* may or may not be accessible to the pronoun. If the antecedent for *his* is *Fred* or *the teacher,* then *his mother* can be the antecedent for *her* (following argument simplification). However, if the antecedent is *every student,* then *his mother* cannot be the antecedent for *her.*

4.4 Definites and Verb Phrase Ellipsis

To handle VPE, we first determine the meanings of definite functions contained in a trigger VP before providing the meaning of an elided sentence. Consider the following example in which *the dog* must denote the same dog in the trigger and elided sentences.

Example 32
Fred saw the dog.
George did too.
Possible Meanings: George saw the same dog that Fred saw.

The initial representation of the trigger sentence is as follows.

Example 33
Fred saw the dog.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(see x ((def$_2$ x) | (dog (def$_2$ x)))

Before deriving the meaning of the elided sentence from this representation of the trigger, we must apply argument simplification to the definite function (*def*$_2$ *x*) (assuming that it is nonanaphoric).

Example 34
Fred saw the dog.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(and (see x ((def$_2$ x) | (dog (def$_2$ x)))) (= (def$_2$ x) (def$_3$)))
Simplification:
((def$_1$) | (name (def$_1$) Fred)), λ(x)(and (see x (def$_3$)) (dog (def$_3$)))

Once the meaning of the definite function is determined, we derive the meaning of the elided sentence by using the VP representation from 34.

Example 35
George did too. (George saw the same dog as Fred did.)
((def$_4$) | (name (def$_4$) George)), λ(x)(and (see x ((def$_2$ x) | (dog (def$_2$ x))))
(= (def$_2$ x) (def$_3$)))

Simplification:
((def$_4$) | (name (def$_4$) George)), λ(x)(and (see x (def$_3$)) (dog (def$_3$)))

Because the final meaning for *the dog* is (*def*$_3$) in 34 and 35, it denotes the same dog.

Consider how we can use definite functions to handle Example 3, discussed in Section 3.4. The initial representation of the trigger sentence is shown in 36.

Example 36
Fred loves his wife.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x)))))

If the pronoun resolution module determines that the antecedent for *his* is *Fred*, then the trigger LF form is modified as follows.

Example 37
Fred$_i$ loves his$_i$ wife.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x))
(or$_a$ (= (his$_3$ x) (def$_1$))
(= (his$_3$ x) x)))))

Depending on the meaning selected for the pronoun *his*, there are two different readings for the elided sentence.

If *his* refers indirectly to *Fred*, the intended meaning for the trigger sentence is shown in 38.

Example 38
Fred$_i$ loves his$_i$ wife.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x))
(= (his$_3$ x) x))))
Simplification:
((def$_1$) | (name (def$_1$) Fred)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess x (def$_2$ x)))))

Notice that *def*$_2$'s restriction contains a free variable *x* and so its argument list is unchanged by argument simplification. Hence, the representation of the VP in 38 is used to derive the sloppy reading of the elided sentence shown in 39.

Example 39
George does too. (George loves George's wife.)
((def$_5$) | (name (def$_5$) George)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x))
(= (his$_3$ x) x))))
Simplification:
((def$_5$) | (name (def$_5$) George)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess x (def$_2$ x)))))

Notice that the function *def*$_2$ denotes a different individual in the trigger and elided sentences, depending on the value of *x*.

On the other hand, suppose that *his* refers directly to *Fred*. Then the intended meaning of the trigger sentence is shown in 40.

Example 40
Fred$_i$ loves his$_i$ wife.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x))
(= (his$_3$ x) (def$_1$)))))
Simplification:
((def$_1$) | (name (def$_1$) Fred)), λ(x)(love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (def$_1$) (def$_2$ x)))))

Notice that once the pronoun function is replaced by (*def*$_1$), the restriction of (*def*$_2$ *x*) contains no free variables except those in the argument list of the function itself. Hence we update the LF as follows.

Example 41
Fred$_i$ loves his$_i$ wife.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(and (love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x))
(= (his$_3$ x) (def$_1$)))))
(= (def$_2$ x) (def$_4$)))
Simplification:
((def$_1$) | (name (def$_1$) Fred)), λ(x)(and (love x (def$_4$)) (wife (def$_4$)) (possess (def$_1$) (def$_4$))

Using this representation of the VP, we derive the strict reading of the elided sentence.

Example 42
George does too. (George loves Fred's wife.)
((def$_5$) | (name (def$_5$) George)), λ(x)(and (love x ((def$_2$ x) | (and (wife (def$_2$ x))
(possess (his$_3$ x) (def$_2$ x))
(= (his$_3$ x) (def$_1$)))))
(= (def$_2$ x) (def$_4$)))
Simplification:
((def$_5$) | (name (def$_5$) George)), λ(x)(and (love x (def$_4$)) (wife (def$_4$)) (possess (def$_1$) (def$_4$))

Notice that (def_4) denotes the same individual in the trigger and elided sentences. Hence, our general LF representation of definite NPs allows us to derive both the sloppy and strict readings of the elided sentence.

We have introduced a composite representation for definite NPs along with a way to update its meaning as more information becomes available. Our approach has several strengths: It is consistent with the three computational constraints discussed in Section 1; it handles a variety of definite behaviors with one mechanism; and it provides useful constraints on intrasentential antecedents for definites (in addition to traditional constraints like number and gender agreement) and a more flexible mechanism than c-command for determining whether a definite noun phrase is a possible antecedent for an anaphoric expression.

5. Indefinites in Logical Form

In this section, we develop an initial LF representation for singular indefinite NPs and provide a mechanism for updating it once additional information is processed.

5.1 Indefinites: Linguistic Behavior and Initial Representation

Singular indefinite NPs share many behaviors with singular definites, including the fact that the final meanings of both are affected by the meanings of embedded NPs. However, unlike definites, the meanings of indefinites are affected by negation, and so they cannot be initially represented as functions in LF. For example:

Example 43
Fred did not see a woman.
Possible Meanings:
a. $\exists$x: (woman x) Not(see Fred x)
b. Not $\exists$x: (woman x) (see Fred x) $\equiv$ $\forall$x: (woman x) Not(see Fred x)

Whenever there is negation in a sentence containing an indefinite, two meanings of the sentence are possible. If the negation does not have scope over the indefinite, then the indefinite is represented as an existential outside the scope of the negation as shown in 43a (and could be represented as a function). In contrast, if the negation has scope over the indefinite, then it has scope over the existential operator making it equivalent to a universal (as shown in 43b). If we represent the indefinite in 43 as a function before deciding whether the negation has scope over it, then the second reading could not be expressed. Thus, we represent indefinites initially as existentially quantified and restricted variables, as in Example 44.

Example 44
Fred saw a dog.
((def$_1$) | (name (def$_1$) Fred)), λ(x)(saw x [$\exists$y: (dog y) y])

To provide a scope-neutral form, we place the quantified term $[\exists y : (dog\ y)\ y]$ directly into the predicate-argument structure (see Appendix A). This initial representation of the indefinite is provided using only syntactic information and knowledge about how to map arguments into the predicate-argument structure, obeying the modularity constraint. Once quantifier scoping information is available, we update the sentence's LF, using a mechanism that also allows us to account for several interesting indefinite behaviors.

5.2 Indefinites: Updating the Initial Representation

An approach that models indefinites solely as existentially quantified variables cannot account for the variety of behaviors of indefinite NPs, since quantifiers cannot have scope over variables in other sentences. This accounts for the fact that the quantified NP cannot be the antecedent for the pronoun *she* in: *The boy kissed every girl. She slapped him.* However, in *The boy kissed a girl. She slapped him,* the antecedent for *she* can be *a girl.*

A similar problem arises in VPE. If the trigger VP contains a pronoun whose antecedent is an indefinite subject, two possible meanings for the elided sentence are possible, as in Example 45.

Example 45
A postman$_i$ saw his$_i$ dog.
A policeman$_j$ did too.

Possible Meanings:	a. A policeman saw the postman's dog.	(strict reading)
	b. A policeman saw his own dog.	(sloppy reading)

When the antecedent for *his* is *a postman,* the elided sentence has two possible meanings, but a quantified representation for the indefinite can only account for one of them.[10] Given that the antecedent for *his* is *a dog,* there are two possible representations for the trigger VP. The pronoun function is either replaced with the lambda or the existential variable corresponding to the subject. If we use the first representation of the pronoun, we are able to derive the sloppy reading. But, if we use the second, then the meaning for the elided sentence would contain an unbound variable because existential quantifiers do not have scope across sentences.

Donkey sentences (originally noticed by Geach [1962]) suffer a similar difficulty. A typical donkey sentence is *Every miner who owns a donkey$_i$ beats it$_i$.* Though the existential

---

[10] This example is in sharp contrast with *Every postman$_i$ saw his$_i$ dog. Every policeman did too.* The elided sentence can only mean *Every policeman saw his own dog.*

operator corresponding to *a donkey* cannot have scope over the pronoun *it*, the NP *a donkey* can be its antecedent. In contrast, in *Every miner who brushed every donkey beat it*, the NP *every donkey* cannot be the antecedent for *it*.

These examples indicate that our initial representation of a singular indefinite is insufficient for modeling the variety of linguistic behaviors exhibited. However, the existential operator is only necessary until we can determine that the indefinite is not in the scope of a negation and can decide what has scope over the indefinite. After this information is available, it is desirable to transform the initial representation into a form more compatible with the behaviors of indefinites, especially if that transformation refines the meaning of the indefinite (conforming with the formal consistency constraint). Our solution is to eliminate the existential operator and replace the existential's variables with skolem functions.

Once scoping is specified, each existentially quantified variable is replaced by a function whose argument list consists of all of the universally quantified variables that have scope over the existential operator. To demonstrate how existential variables are replaced by functions during skolemization, consider Example 46.

Example 46

Some man saw every woman.

Possible Meanings:

1a. $\exists$x: (man x) $\forall$y: (woman y) (see x y)

b. $\forall$y (and (man (indef$_{34}$)) (if (woman y) (see (indef$_{34}$) y)))

2a. $\forall$y: (woman y) $\exists$x: (man x) (see x y)

b. $\forall$y (if (woman y) (and (man (indef$_{35}$ y)) (see (indef$_{35}$ y) y)))

There are two meanings for the sentence in 46, indicated in 1a and 2a, and these meanings are preserved when the existential variables are replaced by skolem functions, as shown in 1b and 2b.

When an indefinite is represented as an existentially quantified variable and is not in the scope of negation, it can be replaced by an **indefinite function**. However, additional information about the indefinite noun phrase must be gathered before such a transformation is performed. We need to:

1. Determine the antecedents of embedded pronouns and anaphoric definites.
2. Determine the meanings of embedded definite and indefinite NPs.
3. Determine whether any embedded universally quantified NPs, not contained in a relative clause, distribute over the indefinite. (The operators that bind the variables contained in the restriction of the existential operator, but that are not themselves contained in the restriction, necessarily have scope over the existential operator, as in the case of definite functions).
4. Determine whether quantifiers within the sentence have scope over the indefinite, even if those quantifiers do not bind a variable contained in the indefinite's restriction. (The final meaning of an indefinite is affected by quantifiers that could never affect the meaning of a definite. Consider the sentence, *His mother saw every boy*. Despite the fact that *his* needs an antecedent, syntactic constraints eliminate *every boy* from the list of

possible candidates. Hence, *his mother* acts as a constant in the sentence. Compare this sentence with *A friend of his saw every boy*. Despite the fact that the antecedent for *his* cannot be *every boy, a friend of his* could still be in the scope of *every boy*).

5. Determine whether lambda operators have scope over the indefinite, even if they do not bind a variable contained in the indefinite's restriction. (We discuss why lambda operator scope over an indefinite is an issue for handling VPE in the next section).

To demonstrate how much information is necessary to determine the final meaning of an indefinite NP, consider the initial representation for the sentence *Every man showed every boy a picture of his mother.*

Example 47

Every man showed every boy a picture of his mother.
∀x: (man x) x, λ(y)(show y [∃w: (and (picture w)
(of w ((def$_1$ w y z) | (and (mother (def$_1$ w y z))
(possess (his$_2$ w y z)
(def$_1$ w y z)))))) w]
[∀z: (boy z) z])

Before determining the final meaning of the indefinite, we must determine the meaning of the definite NP *his mother* and decide whether ∀*x*, ∀*z*, or λ*(y)* has scope over it. If the antecedent for *his* is *every boy*, the updated LF is shown in Example 48.

Example 48

Every man showed every boy a picture of his mother.
∀x: (man x) x, λ(y)(show y [∃w: (and (picture w)
(of w ((def$_1$ w y z) | (and (mother (def$_1$ w y z))
(possess (his$_2$ w y z)
(def$_1$ w y z))
(= (his$_2$ w y z) z))))) w]
[∀z: (boy z) z])

Since the restriction on *(def$_1$ w y z)* contains only the unbound variable *z*, we replace it with a function of *z*.

Example 49

Every man showed every boy a picture of his mother.
∀x: (man x) x, λ(y)(and (show y [∃w: (and (picture w)
(of w ((def$_1$ w y z) | (and (mother (def$_1$ w y z))
(possess (his$_2$ w y z)
(def$_1$ w y z))
(= (his$_2$ w y z) z))))) w]
[∀z: (boy z) z])
(= (def$_1$ w y z) (def$_3$ z))

Because *z* is unbound in the restriction of the existential, ∀z must have scope over ∃*w*; however, we must still determine whether ∀*x* or λ*(y)* also have scope over the existential. If they do not, we update the LF by replacing the existential variable *w* with a function of *z*.

Example 50

Every man showed every boy a picture of his mother.
∀x: (man x) x, λ(y)(and (show y ((indef$_4$ z) | (and (picture (indef$_4$ z))
(of (indef$_4$ z)
((def$_1$ (indef$_4$ z) y z) |
(and (mother (def$_1$ (indef$_4$ z) y z))
(possess (his$_2$ (indef$_4$ z) y z)
(def$_1$ (indef$_4$ z) y z))
(= (his$_2$ (indef$_4$ z) y z) z))))))
[∀z: (boy z) z])
(= (def$_1$ (indef$_4$ z) y z) (def$_3$ z))

By replacing the existential variables with a function of z, we indicate that only ∀z has scope over the existential. This final meaning is compatible with the initial representation of the indefinite in 47, but we have constrained the initial meaning with additional information.

There are several advantages gained by replacing existential variables by functions. First, it provides a way to indicate quantifier scoping in a representation containing only universal and existential quantifiers. Second, the method of indicating scope is similar to Allen's (1987) method since we are not limited to expressing scope as a linear string of operators. Third, universal variables cannot be replaced with functions. Hence, skolemization may be useful for modeling the differences between universals and indefinites in English. Fourth, once quantifier scoping information is available, the replacement of existential variables with functions is a meaning preserving operation as required by the formal consistency constraint. Finally, a functional representation for an indefinite allows us to account for several behaviors that are poorly modeled using an existential variable representation of indefinites alone.

The functional representation for an indefinite provides a mechanism for determining whether a singular indefinite can be the antecedent for a singular pronoun in a subsequent sentence. Consider the example: *Every woman saw a dog. It bit the tallest woman.* The antecedent for the pronoun *it* can be *a dog* only if the universal operator corresponding to *every woman* does not have scope over the existential. Consider the initial representation of the first sentence.

Example 51

Every woman saw a dog.
∀x: (woman x) x, λ(y)(see y [∃z: (dog z) z])

Now, suppose that the universal has scope over the indefinite, then the LF would be updated as follows.

Example 52

∀x: (woman x) x, λ(y)(see y ((indef$_{34}$ x) | (dog (indef$_{34}$ x))))

A consequence of this scoping decision is that *a dog* cannot be the antecedent for *it*, without causing a violation of the formal consistency constraint.[11] On the other hand,

---

[11] Even if we construct a discourse entity for *a dog*, following Webber (1978), the discourse entity for (*indef*$_{34}$ *x*) would denote a set of dogs, and the pronoun resolution module would not allow a plural entity to be the antecedent for a singular pronoun.

if the universal does not have scope over *a dog,* then the LF for the first sentence (shown in 51) is updated as shown in 53.

Example 53
∀x: (woman x) x, λ(y)(see y ((indef$_{37}$) | (dog (indef$_{37}$))))

Because *a dog* is represented as a function with no arguments, it is compatible with the pronoun function representing *it.*[12]

The representation is also useful for handling the donkey sentence *Every miner who owns a donkey$_i$ beats it$_i$.* The antecedent for *it* is *a donkey,* yet in English, a quantified NP contained in a relative clause attached to an NP cannot bind a pronoun outside of that clause. Hence, *it* cannot be bound by the existential quantifier corresponding to *a donkey.* However, once the meaning of the indefinite has been determined it may become formally consistent with the pronoun function. Consider the initial representation of this sentence.

Example 54
Every miner who owns a donkey beats it.
∀x: (and (miner x) x, λ(y)(own y [∃z: (donkey z) z])) x, λ(w)(beat w (it$_{58}$ w))

The pronoun *it* is represented as a function of *w* only and cannot be equated with *z*. However, by replacing the existential term in the relative clause with a functional term, we will be able to assert that the antecedent for *it* is *a donkey.*

To replace the variables corresponding to the existential operator with a function, we must determine whether the existential quantifier is in the scope of negation. Assuming that the negation introduced by the restriction on the universal operator does not have scope over *a donkey,* we replace the existential term with a function whose arguments are the variables corresponding to operators that have scope over it. If only *∀x* has scope over the existential (i.e., *λ(y)* does not have scope), we can assert the anaphoric relationship between *it* and *a donkey.*

Example 55
Every miner who owns a donkey$_i$ beats it$_i$.
Skolemization:
∀x (if (and (miner x) x, λ(y)(own y ((indef$_{22}$ x) | (donkey (indef$_{22}$ x))))) x, λ(w)(beat w (it$_{58}$ w)))
Pronoun Update:
∀x (if (and (miner x) x, λ(y)(own y ((indef$_{22}$ x) | (donkey (indef$_{22}$ x)))))
  x, λ(w)(and (beat w (it$_{58}$ w)) (= (it$_{58}$ w) (indef$_{22}$ x))))

Since *(it$_{58}$ w)* is consistent with a function of *x* (because *λ(w)* abstracts the variable *x*), we can assert the anaphoric relationship, as shown above.[13]

It is important to note that the initial type of a quantified NP determines whether skolemization can make it accessible to a pronoun function whose argument list does not contain that NP's variable. If an NP is initially represented as a universal, then unless the universal variable is included in the argument list of the pronoun, it cannot be the antecedent for that pronoun even if it is in the scope of negation. For example,

---

[12] It could also be used to create a singular discourse entity compatible with the singular pronoun.

[13] Our solution has much in common with Webber's (1978) parameterized individuals. Webber introduces a parameterized individual (which looks much like an indefinite function) as the antecedent for *it.* However, she does not modify the initial representation of the indefinite.

in *Every miner who did not see every donkey beat it*, the antecedent for *it* cannot be *not every donkey*. In contrast, so long as an indefinite remains an existential, even if it cannot bind the pronoun, it may become accessible to the pronoun once we determine its precise behavior and convert it into a functional term.

5.3 Indefinites and Verb Phrase Ellipsis

The representation of an indefinite as a function is also very useful for modeling indefinite subjects in VPE. Consider Example 45 again. By converting existentially quantified variables into functions, we are able to provide the strict meaning for the elided sentence without creating an ill-formed representation; we simply replace the subject's existential variables with a skolem constant (assuming it is not in the scope of another quantifier).

To properly model indefinite NPs contained in a trigger VP for VPE, we must consider whether the lambda operators in a VP representation have scope over indefinites represented as existentially quantified variables. Consider Example 56.

Example 56
Fred saw a dog.
George did too.
Possible Meanings: 1. George saw the same dog that Fred saw.
2. George saw a different dog than Fred saw.

When an indefinite NP occurs in the trigger VP, the elided sentence is ambiguous. This is in contrast to Example 32 discussed in Section 4.4: *Fred saw the dog. George did too*, in which the elided sentence can only mean *George saw the same dog that Fred saw*. By ignoring lambda operators when converting existential variables to skolem functions, we would be unable to provide the second meaning for the elided sentence in this example.

To illustrate this point further, consider how we determine the three meanings of the elided sentence in Example 57.

Example 57
Fred$_i$ saw a friend of his$_i$.
George$_j$ did too.
Possible Meanings: 1. George saw the same friend of Fred's.
2. George saw a different friend of Fred's.
3. George saw a friend of George's.

The initial representation for the trigger sentence is shown in 58.

Example 58
Fred saw a friend of his.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x [∃y: (and (friend y) (possess (his$_{98}$ x) y)) y])

Given that the antecedent for *his* is *Fred*, this LF is updated as shown in 59.

Example 59
Fred$_i$ saw a friend of his$_i$.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x [∃y: (and (friend y)
(possess (his$_{98}$ x) y)
(or$_a$ (= (his$_{98}$ x) x) (= (his$_{98}$ x) (def$_{97}$)))) y])

Now, before converting the existential into its functional form, we must determine which meaning of the pronoun is intended.

Suppose that the pronoun *his* refers indirectly to the subject; then, the trigger LF is refined as shown in 60.

Example 60
Fred$_i$ saw a friend of his$_i$.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x [$\exists$y: (and (friend y)
(possess (his$_{98}$ x) y)
(= (his$_{98}$ x) x)) y])

Since $\lambda(x)$ must have scope over the existential to bind the variable x in its restriction, the existential must be a function of that variable and can be used to provide the third meaning of the elided sentence in 57, as shown in 61.

Example 61
Trigger: Fred$_i$ saw a friend of his$_i$.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x ((indef$_{99}$ x) | (and (friend (indef$_{99}$ x))
(possess (his$_{98}$ x) (indef$_{99}$ x))
(= (his$_{98}$ x) x))))
Ellipsis: George did too. (George saw a friend of George's.)
((def$_{100}$) | (name (def$_{100}$) George)), λ(x)(see x ((indef$_{99}$ x) | (and (friend (indef$_{99}$ x))
(possess (his$_{98}$ x) (indef$_{99}$ x))
(= (his$_{98}$ x) x))))

In contrast, assume that the pronoun *his* refers directly to *Fred*. This choice is reflected in the LF shown in 62.

Example 62
Fred$_i$ saw a friend of his$_i$.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x [$\exists$y: (and (friend y)
(possess (his$_{98}$ x) y)
(= (his$_{98}$ x) (def$_{97}$))) y])

We must still determine whether $\lambda(x)$ has scope over the existential. If it does, we replace the existential variables with a function of x as shown in 63, allowing us to provide the second reading of the elided sentence in 57.

Example 63
Trigger: Fred$_i$ saw a friend of his$_i$.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x ((indef$_{100}$ x) | (and (friend (indef$_{100}$ x))
(possess (his$_{98}$ x) (indef$_{100}$ x))
(= (his$_{98}$ x) (def$_{97}$)))))
Ellipsis: George did too. (George saw a different friend of Fred's.)
((def$_{100}$) | (name (def$_{100}$) George)), λ(x)(see x ((indef$_{100}$ x) | (and (friend (indef$_{100}$ x))
(possess (his$_{98}$ x) (indef$_{100}$ x))
(= (his$_{98}$ x) (def$_{97}$)))))

Finally, if $\lambda(x)$ does not have scope over the existential, then we replace the existential variable with a skolem constant, allowing us to provide the first reading of the elided sentence in 57.

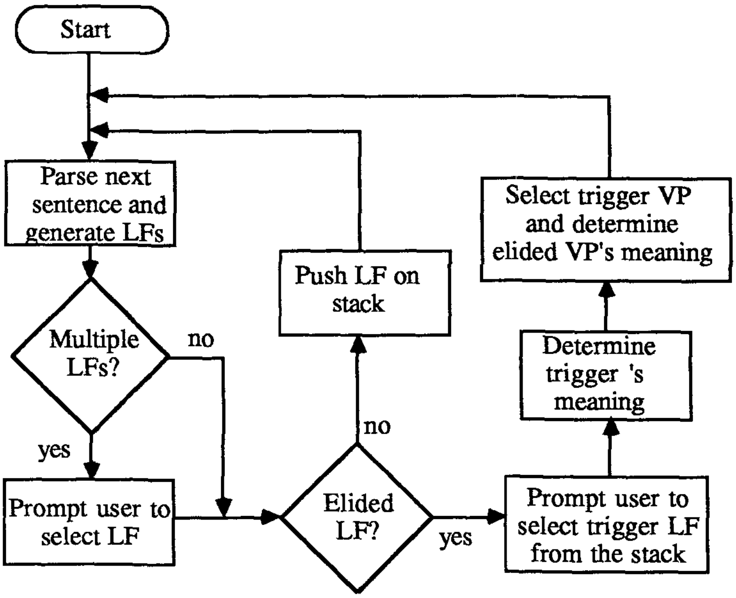

Figure 3
Flow of control for VPE program.

Example 64
Trigger: Fred$_i$ saw a friend of his$_i$.
((def$_{97}$) | (name (def$_{97}$) Fred)), λ(x)(see x ((indef$_{102}$) | (and (friend (indef$_{102}$))
(possess (his$_{98}$ x) (indef$_{102}$))
(= (his$_{98}$ x) (def$_{97}$)))))
Ellipsis: George did too. (George saw the same friend of Fred's.)
((def$_{100}$) | (name (def$_{100}$) George)), λ(x)(see x ((indef$_{102}$) | (and (friend (indef$_{102}$))
(possess (his$_{98}$ x) (indef$_{102}$))
(= (his$_{98}$ x) (def$_{97}$)))))

Hence, we are able to provide all three readings for the elided sentence in 57.

We have described an initial representation for indefinite NPs along with a way to update the representation after more information becomes available. Our approach is consistent with the three computational constraints discussed in Section 1 and models a variety of indefinite behaviors.

6. Implementation

In this section, we describe the operation of a program to generate the meanings for sentences that contain intersentential VPE. This implementation demonstrates that the LF described in this paper can be automatically generated during sentence parsing and that the intended meaning of the LF can be determined through a series of machine guided steps (see Figure 3).

The program parses the sentences in the following example, starting with the trigger sentence.

Example 65

Fred$_i$ saw (his$_i$ mother's)$_j$ picture$_k$.
George$_l$ did too.
Possible Meanings: 1. George saw Fred's mother's picture.
2. George saw George's mother's picture.

If there are multiple parses for a sentence, there will be multiple LFs with each LF corresponding to a different parse tree. In such a case, the user is prompted to select the intended parse (and hence, the intended LF). The trigger sentence in Example 65 has a single parse tree and the parser produces the single LF shown in 66.

Example 66

Fred saw his mother's picture.
((def$_1$) | (name (def$_1$) Fred),
λ(y)(see (subject y) (object ((def$_4$ y) | (and (picture (def$_4$ y))
(possess ((def$_3$ y) | (and (mother (def$_3$ y))
(possess (his$_2$ y)
(def$_3$ y))))
(def$_4$ y))))))

In this program, we label the roles of the noun phrases to keep track of logical subjects and objects in order to prevent a passive voice sentence from becoming the trigger for an active voice sentence (or vice versa). We label the subject's lambda variable as the logical subject for a sentence with active voice or as the logical object for a sentence with passive voice. We do not specify case roles like agent in our LF because their determination may require contextual information. Also, the case role of a subject need not be the same in the trigger and elided sentences, e.g., *Fred hit the window. The hammer did too.* The subject in the first sentence is probably filling the role of agent, whereas the subject of the second fills the role of instrument.

Because our program is designed to provide meanings of sentences with VPE, it examines each LF provided by the parser to see if it contains an elided VP. If it does not contain one, then additional processing of the potential trigger sentence is put off until the final meaning of the sentence is needed. Ideally, processing should be done as information becomes available; however, for the purpose of this implementation, it is conceptually simpler to refine the trigger LF only when its meaning must be determined to disambiguate an elided VP. Hence, the LF in 66 is saved on a stack of recently processed sentences.

To determine the meaning of an elided sentence (a process depicted in Figure 4), the program must first locate the trigger sentence. (The fact that the representation contains an elided VP is indicated by the dummy predicate in 67.)

Example 67

George did too.
((def$_5$) | (name (def$_5$) George)), λ(w)(dummy$_{33}$ (subject w))

To accomplish this, the program has the user select the LF for the trigger sentence from the LFs stored on the stack (assume the user selects the LF in 66). Since the meaning of the trigger sentence is underspecified, the meaning of the elided sentence is also ambiguous. Thus, the program must next determine the trigger sentence's meaning. To do this, the program must find antecedents for all of the pronoun functions. Once the pronouns have antecedents, it handles all definites, and then all quantified NPs. This sequence occurs unless the antecedent for a pronoun is an NP in the sentence

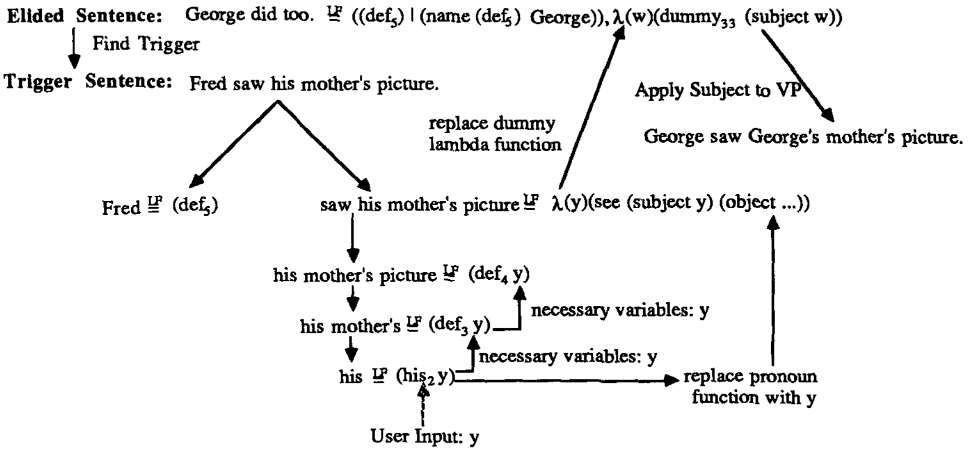

Figure 4
Evaluation of the elided sentence for Example 65.

that is not immediately compatible with the pronoun, in which case, the program attempts to refine the NP's meaning to determine whether it is formally consistent with the pronoun. In our example, the program begins disambiguating the trigger sentence by searching for antecedents of the pronoun function *(his$_2$ y)*. The user is asked to determine whether the antecedent occurs in another sentence. Assuming it doesn't, the pronoun compatibility routine prompts the user to select an antecedent NP from the sentence, using information about the type and location of the pronoun to eliminate impossibilities.[14] If a possible antecedent is a definite or indefinite, even if it is not formally consistent with a pronoun function, it may still become compatible with the pronoun once its meaning is determined.

To refine the meaning of definite noun phrases with embedded NPs, the program must first determine the meanings of embedded pronouns, definites, and indefinites, and then determine whether any embedded quantifiers have scope over the definite. For example, to determine the meaning of *(def$_3$ y)*, the meaning of *(his$_2$ y)* must be determined. If the user selects *Fred* as the antecedent, then the program also prompts the user to select either the lambda variable *y* or *(def$_1$)*. If the user picks the lambda variable (as shown in Figure 4) then the variable replaces the pronoun function in LF. Given this choice, *y* is a necessary argument for *(def$_3$ y)*, so the definite cannot be anaphoric and cannot be further simplified. Since *y* is a necessary argument for *(def$_4$ y)*, there is no modification of that function.

The program continues processing the meanings of the noun phrases in the sentence until the meanings of all constituent NPs are determined (or until an error occurs). Assume, in our example, that the program produces the following refined LF for the trigger sentence in Example 68.

---

[14] If the pronoun is reflexive, its antecedent must occur in the same clause and cannot be embedded in an NP. On the other hand, if the pronoun is not reflexive, its antecedent cannot be in the same clause unless the pronoun or its antecedent is embedded in another noun phrase. Finally, potential antecedents must be formally consistent with the pronoun function.

Example 68
Fred$_i$ saw his$_i$ mother's picture.
((def$_1$) | (name (def$_1$) Fred),
λ(y)(see (subject y) (object ((def$_4$ y) | (and (picture (def$_4$ y))
(possess ((def$_3$ y) | (and (mother (def$_3$ y))
(possess y (def$_3$ y))))
(def$_4$ y))))))

In the event that the trigger sentence contains more than one VP, the program prompts the user to select the trigger VP. Since there is only one VP in 68, the program checks it to ensure that it contains no free variables,[15] is compatible with the voice of the elided VP, and that a trigger sentence with more than one pronoun whose antecedent is the same syntactic subject obeys the multiple pronoun constraint discussed in Harper (1990). Notice that the VP in 68 contains no free variables and is compatible in voice to the elided VP in 67. Hence, the program replaces the elided VP with the trigger VP as shown in 69, and the meaning of the elided sentence is determined.

Example 69
Sentence: George did too.
Meaning: George saw George's mother's picture.
((def$_7$) | (name (def$_7$) George)),
λ(y)(see (subject y) (object ((def$_4$ y) | (and (picture (def$_4$ y))
(possess ((def$_3$ y) | (and (mother (def$_3$ y))
(possess y (def$_3$ y))))
(def$_4$ y))))))

7. Related Work

Other researchers have developed an intermediate representation for a sentence from syntactic information (Pollack and Pereira 1988; Alshawi and van Eijck 1989). These approaches agree that in order to determine the meaning of a sentence, it is useful to build a partial meaning that is augmented once contextual information becomes available. These approaches, however, use a different scheme for indicating the final meaning of a sentence and do not handle VPE.

In the rest of this section, we review past representations of pronouns, definite NPs, and indefinite NPs. We emphasize VPE research because it considers not only the representation of sentences in general, but also the representation of trigger sentences. Each approach is examined in the light of its modeling capability and our computational constraints.

7.1 Verb Phrase Ellipsis and Models of Pronouns

Pronouns are often classified as either bound variable or referential pronouns (Sag 1976; Webber 1978; Reinhart 1983; Partee and Bach 1981), but the adequacy of this dichotomy is questionable (as discussed in Section 3.1). Models of VPE must consider not only pronouns in normal sentences but also pronouns in trigger sentences, and must account for the ambiguity that arises when a pronoun's antecedent is the syntactic subject of a trigger sentence. Sag (1976) and Webber (1978) handle this ambiguity by introducing a rule to replace a pronoun whose antecedent is known to be the syntactic

---

[15] This program only deals with intersentential VPE. Hence, all variables in the trigger VP must be bound in the VP, otherwise the elided sentence cannot receive a meaning. If we augment our approach to handle antecedent-contained ellipsis, we would have to allow variables bound by an operator outside of the VP but inside the meaning of the sentence.

subject of a trigger sentence with the lambda variable corresponding to that subject. They also assume that a pronoun whose antecedent is a nonsubject definite is necessarily referential. But a problem arises when a pronoun's antecedent is a nonsubject, nonreferential definite, as in Example 70.

Example 70
Trigger Sentence: Fred$_i$ showed (his$_i$ mother)$_j$ her$_j$ dog.
Elided Sentence: George$_l$ did too.
Possible Readings:
1. George showed Fred's mother Fred's mother's dog.
2. George showed George's mother George's mother's dog.
Impossible Readings:
3. *George showed George's mother Fred's mother's dog.
4. *George showed Fred's mother George's mother's dog.

Given the indices on the NPs, the elided sentence has meanings 1 and 2, shown in 70. Sag's and Webber's models correctly allow meaning 1 and incorrectly allow meaning 3 at the expense of meaning 2 (because Fred's mother can be the only meaning for *her* given their approach).

Reinhart (1983) also indicates that pronouns are either bound variables or referential, providing a syntactic rule for determining when a pronoun can be bound by its antecedent: a pronoun can be bound by an NP if and only if it c-commands the pronoun. Reinhart does not represent pronouns or definites as quantified terms, yet she claims that when a pronoun's antecedent is a definite NP or a pronoun that c-commands the pronoun, then the pronoun is bound by a lambda operator abstracting the antecedent. At first glance, the idea of binding a pronoun with the lambda operator of its antecedent (given that the NP c-commands the pronoun) seems promising; it can be used to handle Example 70. However, in English, a nonreferential definite can be a pronoun's antecedent even if it does not c-command the pronoun, as in *Every man$_i$ gave the psychiatrist who cares for (his$_i$ mother)$_j$ her$_j$ diary.* Reinhart can only provide the pronoun *her* with a referential meaning, which is inappropriate in this case. Also, Reinhart assumes that the lambda variable is the only nonreferential representation for a pronoun whose antecedent is a definite NP in the same sentence. If this is correct, then there should only be one meaning for a pronoun whose antecedent is a nonreferential definite subject. However, consider Example 71 (which was inspired by an example in Sells, Zaenen, and Zec [1989]).

Example 71
Every man$_i$ believes that (his$_i$ wife)$_j$ can defend herself$_j$ better than he$_i$ can.
Possible Meanings:
1. Every man believes that his wife can defend herself better than he can defend himself.
2. Every man believes that his wife can defend herself better than he can defend her.

Reinhart's approach can only provide the first meaning of the elided sentence in 71. This example suggests that pronouns can refer to definite subjects in two nonreferential ways; lambda abstraction accounts for only one of them.

Partee and Bach (1981) attempt to dispense with LF in translating from syntax to final interpretation, building on Montague's (1970) general theory (with a few modifications to get around the strict compositionality of that approach). All of the possible representations for ambiguous sentences are simultaneously generated, avoiding the need for an intermediate level of representation. They directly provide model-theoretic interpretations for sentences containing pronouns and elided VPs. In their approach, null or elided VPs and pronouns are initially represented as variables. Pronouns are represented as variables that are either bound by some operator or remain free within the representation of the sentence. If a pronoun variable is unbound, it is assigned some value by a context assignment function, that is, a function that maps the variable to the individual that the pronoun denotes. In other words, pronouns are either bound variables in this model or they are referential. An elided VP is represented as a free property variable, typed to receive a value corresponding to a VP already in discourse. It receives its interpretation in much the same way as an unbound pronoun variable, with the exception that its antecedent must be available in linguistic context. Once the value of the null VP is specified, the meaning of the elided sentence is determined.

Bach and Partee point out a variety of examples for which their approach fails. Because there is no mechanism for ensuring that a pronoun bound in the trigger sentence is bound by the same operator in the elided sentence, their approach provides a host of impossible interpretations for elided sentences. For example, their approach provides an impossible interpretation for the elided sentence in *No man believes that Mary loves him. But she does,* given that *no man* is the antecedent for *him* and the meaning of the null VP is *loves him.* Since the variable for *him* is unbound in the elided interpretation, it must be assigned a value (e.g., *Fred*) by the context assignment function. Partee and Bach also discuss examples in which an elided sentence receives an impossible interpretation when a free pronoun variable in the trigger VP becomes accidentally bound by a quantifier in the elided sentence. In addition to the problems pointed out by Partee and Bach, others arise if we assume that definite NPs are quantified.

The above approaches also do not adhere to all of our computational constraints, which are essential in any computer model. For example, Sag's (1976), Webber's (1978), and Reinhart's (1983) approaches do not conform with the formal consistency constraint (because they replace pronoun strings with a variable to account for bound variable meanings of a pronoun) and Partee and Bach's approach (1981) does not conform with the compactness constraint.

7.2 Past Representations of Definites and Indefinites

In this section, we examine previous representations of definite NPs. In particular, we review definite descriptions and definite quantifiers. We also examine some recent work that departs from traditional representations of definite NPs (e.g., Heim [1982]; Roberts [1987]; Kamp [1981]; Klein [1987]).

7.2.1 Definite Descriptions and Definite Quantifiers. Many researchers have attempted to represent definite NPs using definite descriptions or definite quantifiers. Russell (1971) introduced definite descriptions to capture the meaning of definite NPs like *the dog* in Example 72.

Example 72

The dog barked.: (barked (ιx)(dog x))
which means:
$\exists$x (and (Dog x) ; The dog exists.
$\forall$y ((dog y) $\leftrightarrow$ x=y) ; It is the one-and-only dog.
(barked x)) ; It barked.

The definite description, *(ιx)(dog x)*, which stands for *the object x such that the property (dog x) is true* names a unique object, and hence, is translated into the formula, *$\exists x$ (and (dog x) $\forall y$ ((dog y) $\leftrightarrow$ (= x y)))*. Notice three important features of the meaning of the sentence in Example 72: the dog described by the definite NP is assumed to exist, is assumed to be unique, and fills some role in the sentence.

Definite descriptions suffer from several problems. First, there is no role specified for the effect of context on the uniqueness statement (a problem noted by many people, including Allen [1987] and Hintikka and Kulas [1985]). For example, *the dog* in 72 is described as the-one-and-only *the dog*, regardless of context. Second, definite descriptions do not adequately model anaphoric definites (as noted by Hintikka and Kulas [1985]), which need not be unique and seem to adapt to the behavior dictated by their antecedents, as in *Every boy*$_i$ *saw (his*$_i$ *dog)*$_j$ *before the beast*$_j$ *saw him*$_i$. To cover this example, the definite description for *the beast* could be replaced by some value consistent with the representation of its antecedent, but not without violating the formal consistency constraint, or another representation for anaphoric definites could be devised, but this might violate our compactness constraint. Another difficulty involves the representation of Bach–Peters sentences, like *(The boy who wrote her*$_j$*)*$_i$ *kissed (the girl who loved him*$_i$*)*$_j$, which cannot be represented without infinite recursion (as noted by Hintikka and Kulas [1985]).

Other researchers have represented definites using the quantificational meaning of a definite description directly (e.g., Webber [1983] and Montague [1970]). While an in-place definite description simply fills an argument slot in a predicate-argument structure representing the sentence, a quantifier scopes an open sentence. For example, the sentence, *The dog barked*, could be represented as shown in 73 (along with a shorthand notation, where the quantifier $\exists!$ reads *there exists a unique*).

Example 73
The dog barked.
$\exists x$: (and (dog x) ($\forall y$ ((dog y) $\leftrightarrow$ x=y))) (barked x)
Short hand notation: $\exists! x$: (dog x) (barked x)

A quantificational representation for definites suffers from several problems. First, as one might guess, it suffers from the same uniqueness problem that in-place definite descriptions have.[16] Second, definite noun phrases do not exhibit the same type of quantifier scope ambiguity that other quantified NPs have. Compare the two sentences below.

Example 74
a. Every man loves a woman.
    1. $\forall x$: (man x) $\exists y$: (woman y) (loves x y)
    2. $\exists y$: (woman y) $\forall x$: (man x) (loves x y)
b. Every man loves the woman.
    1. $\forall x$: (man x) $\exists! y$: (woman y) (loves x y)
    2. $\exists! y$: (woman y) $\forall x$: (man x) (loves x y)

---

[16] Dowty, Wall, and Peters (1981) suggest a nice way to fix the uniqueness problem. They eliminate the one-and-only aspect of a definite quantifier by relativizing uniqueness to a context of utterance (much as the domain of a universal NP must be relativized to a context of utterance). Though their solution improves definite quantifiers, it does not eliminate the problem with definite anaphora.

While the sentence in 74a has two different meanings and two representations, the sentence in 74b expresses one meaning, but has two representations. Definite descriptions, on the other hand, provide only a single representation for *Every man loves the woman* while providing multiple representations for sentences with definite scope ambiguity, as in *The mechanic adjusted the steering wheel in each car.* Third, as we have already discussed in Section 4.1, the strict reading of the elided sentence in Example 18 is difficult for a quantified definite representation to account for. Finally, pronoun references to definites are not constrained in the way that pronoun references to other quantified NPs are. For example, when a quantifier is embedded in a relative clause attached to a noun phrase, it cannot bind a pronoun outside of that clause, as in *Fred gave the psychiatrist who cares for every woman her diary.* In contrast, the pronoun *her* in *Every man gave the psychiatrist who cares for his mother her diary* can have *his mother* as its antecedent. One might, like Hornstein (1984), assume that definites are quantified but have different properties than universal quantifiers. However, this assumption does not correct some of the problems of definite quantifiers, such as uniqueness or their inability to model anaphoric definites.

Rather than modifying definite quantifiers, it seems appropriate to represent definites as functions and gain four immediate benefits. First, the representation of a definite NP as a uniquely named function provides a mechanism for handling anaphoric definite NPs. Second, because of this naming convention, definite functions are distinct unless an equality is asserted between them. Third, since we represent each definite as a function with a restriction and the restriction provides us with a mechanism for determining the final meaning of the definite, we are able to capture the properties of a definite description without neglecting anaphoric definites. Fourth, while definite quantifiers can violate the compactness and formal consistency constraints,[17] our approach does not.

7.2.2 Heim (1982) and Discourse Representation Theory. Another approach to modeling definites was developed by Heim (1982) and Discourse Representation Theory (DRT) researchers Kamp (1981), Roberts (1987), Klein (1987). A hallmark of these approaches is their ability to handle anaphoric definites in a reasonable way and the commitment to modeling the meaning of a series of sentences in discourse, not just individual sentences.

Heim (1982) treats definites and indefinites very similarly in her theory since both can be referred to across sentence boundaries, unlike universal NPs. To provide an interpretation for a sentence, Heim first determines the *logical form* for a sentence. Her *logical form* is essentially a parse tree with quantifier scoping information indicated, though it is not a logical representation for the meaning of the sentence. Once the *logical form* for a sentence is constructed, she provides a file change semantics for the sentence using felicity conditions to distinguish definites from indefinites. For example, the novelty-familiarity felicity condition states that indefinites should always cause a new discourse referent (or file card) to be created in the discourse model but definite NPs should not introduce a new discourse referent (or file card). Clearly, Heim's approach emphasizes the anaphoric aspect of the definite NP, which both definite descriptions and definite quantifiers fail to handle well.

Heim's model handles anaphoric definites, but she must introduce accommodation to cover nonanaphoric definites. Consider the sentence *Every man*$_i$ *loves his*$_i$ *mother.*

---

[17] Formal consistency is violated when a wide scope definite is replaced with something that cannot be described as a constant and compactness is violated when a different representation is introduced to handle anaphoric definites.

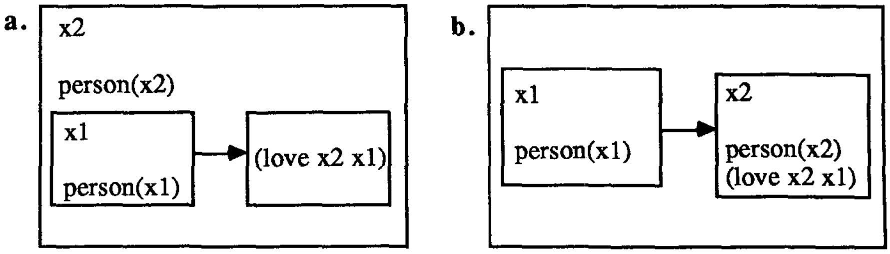

Figure 5
Discourse representations for *Someone loves everyone*.

Accommodation, a concept introduced by Lewis (1979), allows the introduction of a new file card for a definite NP if and only if that NP is related to a previous file card. In the example, *his mother* is nonanaphoric; however, given that the antecedent for *his* is *every man*, accommodation allows the introduction of a new discourse referent. However, to provide a file card for *his mother*, the model must know that *his* refers to *every man*; accommodation requires more than syntax and sentence-level information to provide the representation for some definite NPs. Additionally, Heim's model would have trouble handling any definite NPs without an accommodation link to a previous NP in the discourse model.

Heim's approach requires a considerable amount of information before a definite or indefinite NP is represented in file change semantics (including quantifier scoping information and pronoun antecedents). Hence, the process of representing sentences in file change semantics does not seem to comply with our modularity constraint.

Kamp (1981) introduces a discourse theory similar to Heim's, called Discourse Representation Theory (DRT), providing a model-theoretic interpretation for discourse models. Both theories are motivated by the fact that pronouns in one sentence can have definite and indefinite antecedents in another sentence, while universals cannot bind pronouns in other sentences. Kamp's approach has been extended by several researchers (e.g., Klein [1987]; Roberts [1987]). Since Kamp does not discuss quantifier scope ambiguities, we introduce DRT as discussed by Roberts.

In DRT, a set of construction rules converts natural language into discourse structures. To do so, however, quantifier scoping information must be specified. Consider the two DRT representations for *Someone loves everyone*. The first, shown in Figure 5a, corresponds to the reading in which *someone* has scope over *everyone*. The discourse referents for *everyone* and *someone* are *x1* and *x2*, respectively. The universal NP causes the creation of the antecedent-consequent box. Because *x2* is defined outside of *x1*'s antecedent-consequent box, it acts like a constant. The second, shown in Figure 5b, corresponds to the reading in which *everyone* has scope over *someone*. Because the discourse referent for *someone* is created in the consequent box of the universal, its denotation depends on *x1*, the discourse referent for *everyone*. Each quantifier scoping requires a different discourse representation for the meaning of the sentence.

Mapping into a discourse representation is a top-down process that reduces the original sentence to a structure with a discourse referent for each noun phrase, with predicates indicating restrictions on the discourse referents as well as relations between discourse referents. As we already pointed out, a universal is represented by placing its discourse referent and restriction into an antecedent box, with additional sentence information placed in the consequent box (resulting in a meaning like a universal in

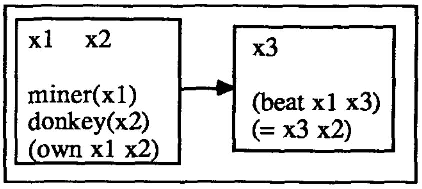

Figure 6
Discourse representation for *Every miner who owns a donkey beats it.*

predicate calculus). Indefinites and definites are represented by placing their discourse referents and restriction information in the box corresponding to the current level in the model. An accessibility relation determines when a pronoun can have a particular discourse referent as its antecedent. A pronoun's antecedent can be any discourse referent defined in the box where the pronoun is instantiated or in any box containing that box. Additionally, a pronoun in a consequent box can also refer to anything in the antecedent box (unless the antecedent is embedded in another box contained in the antecedent box).

Roberts (1987) combines DRT with c-command to distinguish two types of binding, c-command binding and discourse binding. C-command binding occurs when the best way to represent the anaphoric NP is by replacing it with the variable associated with the operator of the NP that c-commands it. On the other hand, discourse binding is needed to handle anaphoric dependencies on things that don't c-command a pronoun or anaphoric definite. For example, consider the sentence *Every miner who owns a donkey beats it.* In Robert's approach, if *a donkey* had c-commanded *it,* then no discourse referent would be created for *it;* instead, the pronoun would be represented using the discourse referent of its antecedent. However, because *a donkey* does not c-command *it,* the sentence is handled as shown in Figure 6. Notice that the pronoun is represented as a discourse referent *x3,* which is equated with the discourse referent for *a donkey* (i.e., *x2*). Pronouns that haven't already been replaced by a discourse referent must be equated with some accessible discourse referent.

Klein (1987) has augmented DRT to handle VPE by introducing a concept that is very similar to lambda abstraction. He is able to represent VPs and *abstract* the syntactic subject of the sentence. Consider how Example 3 is handled.

Example 3
Trigger Sentence: Fred$_i$ loves his$_i$ wife.
Elided Sentence: George$_j$ does too.
Possible Meanings: 1. George loves Fred's wife. (strict meaning)
2. George loves George's wife. (sloppy meaning)

The discourse representation for the two readings of the elided sentence are shown in Figure 7. Klein represents the trigger VP as a boxed structure named P. Within this box is a distinguished variable *x2* (distinguished variables are marked with brackets), which corresponds to the abstracted subject. The trigger sentence is represented as *P(x0),* which is very similar to applying the discourse referent for the subject to a lambda function named P. The discourse referent for *his* is *x4,* which can either be equated with the distinguished discourse referent (i.e., *x2*) or with something outside of the VP box. To get the sloppy reading, it is equated with the distinguished discourse

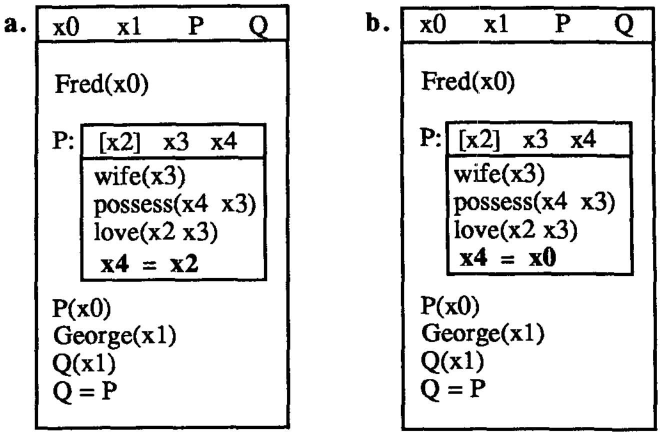

Figure 7
Discourse representations for the readings of Example 3.

referent, as shown in Figure 7a. The elided sentence is represented initially as $Q(x1)$, where $x1$ is the discourse referent for the subject of the elided sentence. The sloppy reading for the sentence is provided when Q is equated with P.

On the other hand, to derive the strict reading of the elided sentence, the discourse referent for the pronoun in the VP is equated with the discourse referent for the subject, namely $x0$ (as shown in Figure 7b). Again the meaning of the elided sentence is derived by equating Q with P, but in this case the pronoun's discourse referent is equated with the discourse referent for the subject. Hence, Klein derives the two expected readings for the elided sentence in 3.

This approach to VPE is similar to ours, except we introduce explicit differences between definite and indefinite NPs. To see why this is an issue, compare Examples 75 and 57.

Example 75
Fred$_i$ saw his$_i$ friend.
George$_j$ did too.
Possible Meanings: 1. George saw Fred's friend.
2. George saw George's friend.

Example 57
Fred$_i$ saw a friend of his$_i$.
George$_j$ did too.
Possible Meanings: 1. George saw the same friend of Fred's.
2. George saw a different friend of Fred's.
3. George saw a friend of George's.

Because definites and indefinites are represented in the same way in DRT, the sloppy readings for both of these examples are represented in precisely the same way (shown in Figure 8a), as are the strict readings (shown in Figure 8b). However, the meaning of *his friend* is quite different from the meaning of *a friend of his* in VPE. The elided sentence in 75 cannot mean *George saw a different friend of Fred's*, in contrast to the elided

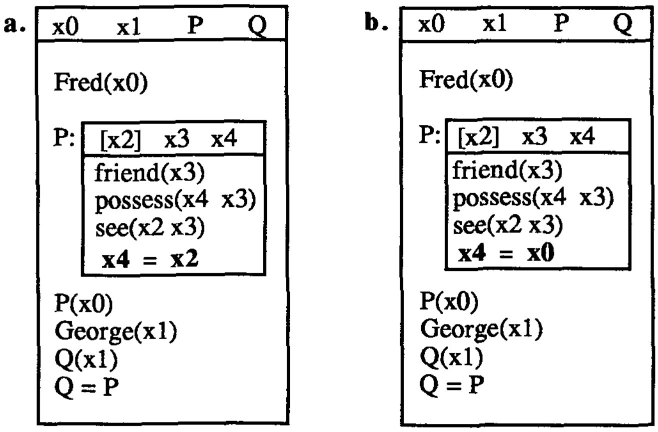

Figure 8
Discourse representations for the readings of Examples 57 and 75.

sentence in 57. One more interpretation is available for the elided sentence in 57 than for the elided sentence in 75, but in each case, only two representations are provided. In our approach, definites and indefinites are treated quite differently, not simply in how they are initially represented, but also in how they are processed to determine final meanings. It is therefore easier for us to explain the differences between examples 57 and 75. In Section 5.3, we demonstrated how our approach is able to provide the three readings for the elided sentence in 57. In contrast, our approach provides only two readings for the elided sentence in 75, as shown with a similar example (i.e., Example 3) in Section 4.4.

In Klein's (1987) approach to VPE, the discourse referents for definites and indefinites are provided only after quantifier scoping information is available. Hence, Klein's discourse representation for these NPs is similar to our refined representations for definites and indefinites (i.e., definite functions following argument simplification and indefinite functions), where a discourse referent that is created in a box introduced by a universal corresponds to a function of a universal variable. However, Klein's approach does not determine whether lambda operators (i.e., distinguished variables) have scope over discourse referents in a VP box. To handle Examples 57 and 75 in the framework of DRT, Klein must determine whether a discourse referent should be defined inside or outside of the VP box, stipulating that a discourse referent is created inside a box if and only if the operator responsible for introducing that box has scope over the NP. In contrast to our approach, the information required to make this decision must be known before building the discourse model.

8. Conclusion

We began with the idea that LF, a compact intermediate level of representation derived only from syntactic and sentence-level information, is a valuable component in any computer model generating meaning for sentences. It can be used to postpone the determination of an unambiguous meaning for a sentence until the information required to select the intended meaning becomes available. Even though the meaning of the logical form for a sentence is ambiguous, it can be used by a contextual processing module to guide the refinement of its own meaning (as well as the meaning of other sentences' LFs).

We have developed logical form representations for pronouns, definites, and indefinites and have demonstrated through a variety of detailed examples that these representations accurately model the linguistic behavior of the language (for a discussion of a wider variety of examples see Harper [1989, 1990]). We have proposed three computational constraints for using LF in a computational framework and have demonstrated that our LF conforms to these constraints. And finally, we used our representation in an implementation that was capable of processing nontrivial examples.

There are two topics that we have only touched on in this paper: the extraction of information from context and the handling of syntactic ambiguity. Though LF constrains the information sought by a contextual module, it does not completely specify a strategy for locating and processing contextual information. A single parse tree can be automatically mapped into LF. However, in syntactically ambiguous sentences (e.g., *Fred saw the bird with his binoculars*) more than one parse is possible. A solution is to allow multiple LFs for a sentence (though this would violate the compactness constraint). Another solution, which we are currently pursuing, is to associate LFs with a parse forest for the sentence (resulting in a compact representation for the parse trees and logical forms).

Acknowledgments

This paper contains results from the author's thesis in the Computer Science Department at Brown University. The paper has benefited from discussions with Eugene Charniak, Leora Morgenstern, Tom Dean, Paul Harper, Kate Sanders, and Frederic Evans. It has also benefited from the helpful suggestions of the referees. The work was supported in part by the NSF grants IST 8416034, IST 8515005, and IST 9011179 and ONR grant N00014-79-C-0529, and AFOSR grant F49620-88-c-0132.

References

Allen, James (1987). *Natural Language Understanding*. The Benjamin/Cummings Publishing Company.

Alshawi, Hiyan, and van Eijck, Jan (1989). "Logical forms in the core language engine." In *Proceedings, 27th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada, 25–32.

Bach, Emmon, and Partee, Barbara (1980). "Anaphora and semantic structure." In *Papers from the Parasession on Pronouns and Anaphora*, edited by Jody Kreiman and A. E. Ojeda, Chicago Linguistic Society, 1–40.

Charniak, Eugene, and McDermott, Drew V. (1985). *Introduction to Artificial Intelligence*. Addison-Wesley.

Creaney, Norman, and McTear, Michael F. (1990). *The Incremental Disambiguation of Quantifier Scoping Ambiguities*. Unpublished manuscript, University of Ulster, N. Ireland.

Dowty, David R.; Wall, Robert E.; and Peters, Stanley (1981). *Introduction to Montague Semantics*. D. Reidel Publishing Company.

Evans, Gareth (1980). "Pronouns." *Linguistic Inquiry*, 11, 337–362.

Geach, Peter T. (1962). *Reference and Generality*. Cornell University Press.

Harper, Mary P. (1988). "Representing pronouns in logical form: Computational constraints and linguistic evidence." In *Proceedings, 7th National Meetings of AAAI*. Saint Paul, Minnesota, 712–717.

Harper, Mary P. (1989). "The representation of noun phrases in logical form." Memorandum CS-89-46, Brown University, Providence, Rhode Island.

Harper, Mary P. (1990). *The representation of noun phrases in logical form*. Doctoral dissertation, Brown University, Providence, Rhode Island.

Heim, Irene (1982). *The semantics of definite and indefinite noun phrases*. Doctoral dissertation, University of Massachusetts, Amherst, Massachusetts.

Hintikka, Jaako (1979). "Quantifiers in logic and quantifiers in natural language." In *Game Theoretical Semantics*, edited by E. Saarinen, 81–117. D. Reidel.

Hintikka, Jaako, and Kulas, Jack (1985). *Anaphora and Definite Descriptions: Two Applications of Game-Theoretical Semantics*. D. Reidel.

Hobbs, Jerry R. (1983). "An improper

treatment of quantification in ordinary English." In *Proceedings, 21st Annual Meeting of Annual Meeting of Association for Computational Linguistics*. Cambridge, Massachusetts, 57–63.

Hornstein, Norbert (1984). *Logic as Grammar: An Approach to Meaning in Natural Language*. MIT Press, 277–322.

Kamp, Hans (1981). "A theory of truth and semantic representation." In *Formal Methods in the Study of Language, Volume 1*, edited by Jeroen Groenendijk, Theo Janssen, and Martin Stokhof. Mathematische Centrum.

Kartunnen, Lauri (1969). "Pronouns and variables." In *Proceedings, Fifth Regional Meeting of the Chicago Linguistic Society*. 108–116.

Klein, Ewan (1987). "VP ellipsis in DR theory." In *Studies in Discourse Representation and the Theory of Generalized Quantifiers*, edited by J. Groenendijk, D. de Jongh, and M. Stokhof. Foris. 161–187.

Lewis, David (1979). "Score keeping in a language game." In *Journal of Philosophical Logic*, 8(3), 339–359.

May, Robert (1985). *Logical Form: Its Structure and Deviation*. MIT Press.

Montague, Richard (1970). "Universal grammar." *Theoria*, 36, 373–398.

Morgenstern, Leora (1988). *Foundations of a logic of knowledge, action, and communication*. Doctoral dissertation, New York University, New York, New York.

Partee, Barbara, and Bach, Emmon (1981). "Quantification, pronouns, and VP anaphora." In *Formal Methods in the Study of Language, Volume 1*, edited by Jeroen Groenendijk, Theo Janssen, and Martin Stokhof. Mathematische Centrum, 445–481.

Pollack, Martha E., and Pereira, Fernando C. N. (1988). "An integrated framework for semantic and pragmatic interpretation." In *Proceedings, 26th Annual Meeting of Annual Meeting of Association for Computational Linguistics*. Buffalo, New York, 75–86.

Reinhart, Tanya (1983). *Anaphora and Semantic Interpretation*. Croom Helm.

Roberts, Craige (1987). *Modal subordination, anaphora, and distributivity*. Doctoral dissertation, University of Massachusetts, Amherst, Massachusetts.

Ross, John R. (1967). *Constraints on variables in syntax*. Doctoral dissertation, MIT, Cambridge, Massachusetts.

Russell, Bertram (1971). "Reference." In *Readings in the Philosophy of Language*, edited by J. F. Rosenberg and C. Travis. Prentice-Hall, 163–175.

Sag, Ivan A. (1976). *Deletion and logical form*. Doctoral dissertation, MIT, Cambridge, Massachusetts.

Schubert, Lenhart K., and Pelletier, Francis J. (1984). "From English to logic: Context-free computation of 'conventional' logical translations." *American Journal of Computational Linguistics*, 10, 165–176.

Sells, Peter; Zaenen, Annie; and Zec, Draga (1985). "Reflexivization variation: Relations between syntax, semantics, and lexical structure." In *Studies in Grammatical Theory and Discourse Structure, Volume 1*, edited by M. Iida, S. Wechsler, and D. Zec. CSLI.

Webber, Bonnie L. (1978). *A formal approach to discourse anaphora*. Doctoral dissertation, Harvard University, Cambridge, Massachusetts.

Webber, Bonnie L. (1983). "So what can we talk about now?" In *Computational Models of Discourse*, edited by M. Brady and R. Berwick, 331–371. MIT Press.

Williams, Edwin S. (1977). "Discourse and logical form." *Linguistic Inquiry*, 8, 101–139.

Appendix A: The Syntax and Semantics of Logical Form

In this appendix, we provide a syntax and semantics for our logical form (adapted from Morgenstern [1988]). We represent a sentence S using the formula τ, $\lambda(w)\phi$, where τ is a term representing the subject and $\lambda(w)\phi$ is the lambda function representing the VP.

A.1 Syntax

1. The logical constants: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$,), (, =, $\forall$, $\exists$, :, |, λ. We often use English equivalents of the logical constants (e.g., *and* for $\wedge$).
2. Nonlogical constants: These include numerical constants (e.g., 1, 2, etc.), character constants (e.g., A, b, etc.), nonnumerical, noncharacter constants (e.g., Fred34).
3. Variables: For example, x, y, z.
4. Predicate symbols: For example, run, boy, etc.
5. Function symbols: For example, def_4, indef_{66}, his_{66}.

We will also characterize terms, q-terms, f-terms, atomic formulae, well-formed formulae, and sentences.

1. Terms: A term is any expression that refers to an object. Formally, they are defined by the following rules:
   (a) If τ is a constant, τ is a term.
   (b) If τ is a variable, τ is a term.
   (c) If $\tau_1, \tau_2, \ldots, \tau_n$ are terms, and θ is an n-ary function symbol, then $(\theta\ \tau_1\tau_2\ldots\tau_n)$ is a term.
2. Q-term: We introduce the idea of a q-term to the syntax. A q-term is a quantifier, restriction (which is a well-formed formula), and variable used as a pseudo-term in a formula. The meaning of a q-term will be introduced in the section on semantics. For example, $[\forall\alpha\colon \psi\ \alpha]$ is a q-term, where α is a variable and ψ is a well-formed formula.
3. F-term: We introduce the idea of an f-term to the syntax. An f-term consists of a function and a restriction (which is a well-formed formula). The meaning of an f-term will be introduced in the section on semantics. For example, $((\theta\ \tau_1\ \tau_2\ldots\tau_n) \mid \psi)$ is an f-term, where ψ is a well-formed formula.
4. Atomic formula: If π is an n-ary predicate symbol, and $\tau_1, \tau_2, \ldots, \tau_n$ are terms, f-terms or q-terms, then $(\pi\ \tau_1\tau_2\ldots\tau_n)$ is an atomic formula.
5. Well-formed formula: well-formed formulas are defined by the following formation rules:
   (a) If ϕ is an atomic formula, then it is a well-formed formula.
   (b) If ϕ is a well-formed formula $\tau_1{=}\tau_2$ and τ_1 and τ_2 are terms, then ϕ is a well-formed formula.
   (c) If ϕ is a well-formed formula, then $\neg\phi$ is a well-formed formula.
   (d) If ϕ_1 and ϕ_2 are well-formed formulae, then (or ϕ_1 ϕ_2), (or_a ϕ_1 ϕ_2), (and ϕ_1 ϕ_2), (if ϕ_1 ϕ_2), and (iff ϕ_1 ϕ_2) are well-formed formulae.

- (e) If ϕ is a well-formed formula and α is a variable, then $\forall\alpha\ \phi$ and $\exists\alpha\ \phi$ are well-formed formulae.
- (f) If ϕ is a formula containing q-terms or f-terms, then that formula is well-formed.
- (g) If ϕ is a well-formed formula and τ is a term, q-term, or f-term, then τ, $\lambda(x)\phi$ is a well-formed formula.

6. Sentences: Sentences are well-formed formulae that do not contain free variables and are capable of being mapped to a truth value, which requires:
   - (a) the refinement of the meanings of pronoun functions and anaphoric definite functions,
   - (b) the simplification of the argument list of nonanaphoric definite functions,
   - (c) the selection of one statement in or_a forms,
   - (d) the determination of quantifier scoping,
   - (e) the replacement existential variables not in the scope of negation with indefinite functions,
   - (f) the replacement of dummy VPs.

   Note that or_a is used as a device (outside of the logical system) for updating LF with additional (though ambiguous) information or for enumerating all possible meanings for a LF.

A.2 Semantics
The model M for language L.

1. Domain D of objects in the world.
2. A mapping assigning each nonlogical constant of the language a member of the domain.
3. A mapping assigning each n-ary predicate of L a set consisting of n-tuples that can be formed out of elements of D.
4. A mapping assigning each n-ary function of L a set of n + 1 tuples formed from the elements of D.

We define the value of a constant term τ under interpretation M as follows:

1. If τ is a constant, then the value of τ under M is the element of D that M maps to τ.
2. If τ is of the form $(\theta\ \tau_1\ \tau_2 \ldots \tau_n)$ where θ is an n-ary function symbol, σ_1 is the value for τ_1, σ_2 is the value for $\tau_2, \ldots, \sigma_n$ is the value for τ_n, and $[\sigma_1, \sigma_2, \ldots, \sigma_n, \sigma]$ is an element in the set of $n+1$ tuples that M maps to θ, then σ is the value of τ.

To this we add:

1. If $(\mathrm{pro}_i\ \tau_1\ \tau_2 \ldots \tau_n)$ occurs in a formula ϕ and $\tau_1, \ldots, \tau_n$ are terms and for any τ_i that is a variable bound by $\lambda(\tau_i)$, the lambda operator must have

scope over the function. Also for any variable τ_j not bound by a lambda operator, then if there is an operator over τ_j contained as a q-term in the formula containing the pronoun function (or in a higher formula), then that operator op_j has scope over the pronoun function.

2. If $(\mathrm{def}_i\ \tau_1\ \tau_2 \ldots \tau_n)$ occurs in a formula ϕ and $\tau_1, \ldots, \tau_n$ are terms and for any τ_i that is a variable bound by $\lambda(\tau_i)$, then the lambda operator has scope over the function. Also for any variable τ_j not bound by a lambda operator, then if there is an operator over τ_j contained in a q-term in the same formula (or in a higher formula) as the function but outside of the function's restriction, then that operator op_j has scope over the function. On the other hand, if for any τ_j that is a variable whose operator is a universal or an existential contained in the function's restriction, the function can receive one of the following 2 meanings:
   1. $\mathrm{op}_j\tau_j\ (\mathrm{def}_i\ \tau_1\ \tau_2 \ldots \tau_j \ldots \tau_n)$
   2. $(\mathrm{def}_i\ \tau_1\tau_2 \ldots \tau_{j-1}\ \tau_{j+1} \ldots \tau_n))$

What it means for a sentence ϕ to be true under an interpretation M:

1. If ϕ is an atomic sentence (i.e., ϕ is of the form $(\pi\ \tau_1, \ldots, \tau_n)$ where σ_1 is the value of $\tau_1, \ldots$, and σ_n is the value of τ_n), then $\mathrm{M} \models \phi$ if and only if $[\sigma_1, \ldots, \sigma_n]$ is a member of the set that M assigns to π.
2. If ϕ has the form $\neg\phi_1$, where ϕ_1 is an atomic sentence of the form $(\pi\ \tau_1, \ldots, \tau_n)$ and σ_1 is the value of $\tau_1, \ldots$, and σ_n is the value of τ_n), then $\mathrm{M} \models \phi$ if and only if $[\sigma_1, \ldots, \sigma_n]$ is in the antiextension of the set that M assigns to π.
3. If ϕ has the form $\phi_1 \vee \phi_2$, $\mathrm{M} \models \phi$ if and only if $\mathrm{M} \models \phi_1$ or $\mathrm{M} \models \phi_2$ or both.
4. If ϕ has the form $\phi_1 \wedge \phi_2$, $\mathrm{M} \models \phi$ if and only if $\mathrm{M} \models \phi_1$ and $\mathrm{M} \models \phi_2$.
5. If ϕ has the form $\phi_1 \rightarrow \phi_2$, $\mathrm{M} \models \phi$ if and only if $\mathrm{M} \models \neg\phi_1$ or $\mathrm{M} \models \phi_2$ or both.
6. If ϕ has the form $\phi_1 \leftrightarrow \phi_2$, $\mathrm{M} \models \phi$ if and only if $\mathrm{M} \models \phi_1$ and $\mathrm{M} \models \phi_2$ or $\mathrm{M} \models \neg\phi_1$ and $\mathrm{M} \models \neg\phi_2$.
7. For sentences of the form $\forall\alpha\psi$ or $\exists\alpha\psi$, we use β-variants. If M and M′ are interpretations with identical domains and β is a constant, M is a β-variant of M′ if M and M′ differ only in what they assign to β.
   1) If ϕ has the form $\forall\alpha\psi$, $\mathrm{M} \models \phi$ if and only if, for all M′, if M′ is a β-variant of M, $\mathrm{M}' \models \psi(\beta/\alpha)$, where $\psi(\beta/\alpha)$ is the expression obtained by substituting β for all free occurrences α in ψ.
   2) If ϕ has the form $\exists\alpha\psi$, $\mathrm{M} \models \phi$ if and only if, for some M′, if M′ is a β-variant of M, $\mathrm{M}' \models \psi(\beta/\alpha)$, where $\psi(\beta/\alpha)$ is the expression obtained by substituting β for all free occurrences α in ψ.
8. If ϕ has the form τ, $\lambda(\mathrm{x})\psi_2$ and τ is a term, then $\mathrm{M} \models \phi$ if and only if $\mathrm{M} \models \psi'_2$ (where ψ'_2 is $\psi_2(\tau/\mathrm{x})$).
9. If ϕ has the form τ, $\lambda(\mathrm{x})\psi_2$ and τ is the q-term $[\forall\alpha\colon \psi_1\ \alpha]$, then $\mathrm{M} \models \phi$ if and only if $\mathrm{M} \models \forall\alpha$ (if $\psi_1\ \psi'_2$) (where ψ'_2 is $\psi_2(\alpha/\mathrm{x})$).

10. If ϕ has the form τ, $\lambda(x)\psi_2$ and τ is the q-term $[\exists\alpha\text{: } \psi_1\ \alpha]$, then $M \models \phi$ if and only if $M \models \exists\alpha$ (and $\psi_1\ \psi_2'$) (where ψ_2' is $\psi_2(\alpha/x)$).
11. If ϕ has the form τ, $\lambda(x)\psi_2$ and τ is the f-term $((\theta\ \tau_1\ \tau_2 \ldots \tau_n) \mid \psi_1)$, then $M \models \phi$ if and only if $M \models$ (and $\psi_1\ \psi_2'$) (where ψ_2' is $\psi_2((\theta\ \tau_1\ \tau_2 \ldots \tau_n)/x)$

Additionally:

1. If $((\theta\ \tau_1\ \tau_2 \ldots \tau_n) \mid \psi_1)$ is contained in a formula ϕ, the formula is equivalent to (and $\phi'\ \psi_1$), where ϕ' is the formula obtained by replacing the f-term with $(\theta\ \tau_1\ \tau_2 \ldots \tau_n)$ in ϕ.
2. If $(\pi\ [\text{op}_1\alpha_1\text{: } \psi] \ldots)$ is a formula, then this is equivalent to:
   op_1 (and ψ $(\pi\ \alpha_1 \ldots)$) if op_1 is $\exists$ or op_1 (if ψ $(\pi\ \alpha_1 \ldots)$) if op_1 is $\forall$.
3. If $(\pi \ldots [\text{op}_j\alpha_j\text{: } \psi_j\ \alpha_j] \ldots [\text{op}_i\alpha_i\text{: } \psi_i\ \alpha_i] \ldots)$ and α_j is free in ψ_i then $\text{op}_j\alpha_j$ must have scope over op_i.
4. If τ_1, $\lambda(\alpha_1)(\pi \ldots [\text{op}_i\alpha_i\text{: } \psi_i\ \alpha_i] \ldots)$ and op_i is an existential operator, then replace the q-term with an f-term, where the function's arguments include all of the variables corresponding to operators that have scope over it (possibly including α_1)) and replace all occurrences of α_i with the function.
5. If τ_1, $\lambda(\alpha_1)(\pi \ldots [\text{op}_i\alpha_i\text{: } \psi_i\ \alpha_i] \ldots)$ and α_1 is free in ψ_i then $\lambda(\alpha_1)$ must have scope over op_i.
6. $(\pi\ [\text{op}_1\alpha_1\text{: } \psi_1\ \alpha_1]\ [\text{op}_2\alpha_2\text{: } \psi_2\ \alpha_2] \ldots [\text{op}_n\alpha_n\text{: } \psi_n\ \alpha_n])$ is consistent with the set of meanings (using or_a to enumerate them) derived by enumerating all sentences with a legal partial order of the quantifiers given the above quantifier scoping constraints: $\text{op}_1\alpha_1, \text{op}_2\alpha_2, \ldots,$ and $\text{op}_n\alpha_n$. To map the sentence to a truth-value, one partial order must be selected.