Representing Pronouns in Logical Form: Computational Constraints and Linguistic Evidence

Mary P. Harper*
Department of Computer Science
Brown University
Box 1910, Providence, RI 02912

Abstract

In this paper, we discuss the representation of pronouns in logical form for the purpose of handling verb phrase ellipsis. In particular, we discuss two factors which influence the representation of pronouns in a computational model. The first is computational, the other linguistic. Both factors must be attended to in order to construct a good representation for pronouns in logical form. We review past attempts to represent pronouns in logical form for the purpose of handling verb phrase ellipsis, and show how these approaches do not meet the computational constraints outlined in this paper. We also show that they do not handle a rather simple example of verb phrase ellipsis. We develop a representation for pronouns in logical form which both meets the computational criteria outlined in this paper and handles the verb phrase ellipsis example.

1 Introduction

In this paper, we discuss the representation of pronouns in logical form. In particular, we discuss two factors which influence the representation of pronouns in a computational model. The first factor is computational, the other linguistic. Both factors must be attended to in order to devise a good representation for pronouns in logical form.

In the remainder of this paper, we examine the effect of these two factors on the representation of pronouns in the domain of verb phrase ellipsis. In Section two, we define three computational constraints which affect the way logical form is used. We then briefly discuss linguistic evidence which affects the representation of pronouns. In Section three, we demonstrate how past approaches to verb phrase ellipsis have failed to represent pronouns in logical form in a way consistent with our computational constraints. In Section four, we discuss our representation of pronouns in logical form. Finally in Section five, we show how our pronoun representation models the behavior of pronouns in verb phrase ellipsis better than past approaches.

*This work has benefited from discussions with Eugene Charniak, and was supported in part by the National Science Foundation under grants IST 8416034 and IST 8515005 and by the Office of Naval Research under grant N00014-79-C-0529.

2 Factors in a Computational Model

2.1 Computational Constraints

Logical form is an intermediate level of representation between phrase markers (which facilitate syntax and parsing), and internal representations (which facilitate inference). Logical form has been quite popular within the Artificial Intelligence community [Webber, 1978; Schubert and Pelletier, 1984; Allen, 1987] because it solves a serious computational problem. More semantic information can be gathered from a sentence than can be specified in a phrase marker, but not enough is available to give the internal representation of the sentence. In particular, when logical form is derived from a sentence, each noun phrase is assigned a logical role (e.g. agent, patient, etc.) and the verb maps to the predicate. In contrast, quantifier scoping and the antecedents of pronouns cannot be specified using only sentence-level information. Logical form provides the needed intermediate level between phrase markers and internal representation. It allows us to represent a sentence before determining how other sentences affect its meaning. With further processing, logical form could be modified into a single unambiguous interpretation of the sentence.

We propose three constraints on the use of logical form in a computational model of language:

1. Logical form should compactly represent ambiguity.
2. Logical form should be initially computable from syntax and local (sentence-level) semantics. In particular, logical form should not be dependent on pragmatics, which requires inference and hence internal representation.
3. Further processing of logical form should only disambiguate or further specify logical form. Logical form has a meaning. Any further processing must respect that meaning.

These constraints express how logical form should be used in a computational model. The first constraint expresses space concerns. The other two concern the plausibility of computing logical form, and incrementally updating it in a meaningful way. Next, we present linguistic evidence indicating how pronouns behave in the domain of verb phrase ellipsis.

2.2 Linguistic Evidence

If we want to model a certain linguistic phenomenon, additional constraints on logical form become necessary. These constraints should facilitate a model's capability to capture all and only the possible meanings of a sentence. In this paper we investigate the representation of pronouns in order to handle verb phrase ellipsis. Verb phrase ellipsis is the deletion of a verb phrase in a sentence. The second sentence in Example 1 is a sentence with verb phrase ellipsis (also called an *elided sentence*).

Example 1

```
Trigger Sentence: Fred_i loves his_i wife.
Elided Sentence:  George_j does too.
Meanings: 1. George loves Fred's wife.
          2. George loves George's wife.
```

An elided sentence has little meaning independent of the first sentence (often called a *trigger sentence*). In this example, the index on *Fred* and *his* indicates that they are coreferential. Given the fact that *his* refers to the subject of the trigger sentence, and the elided sentence depends on that sentence for its meaning, the meaning of the elided sentence is ambiguous. It can either mean that *George loves the same person as Fred*, or that *George loves his own wife*. The representation of pronouns in verb phrase ellipsis is interesting because of this ambiguity (also called *the sloppy identity problem* [Ross, 1967; Ross, 1969]).

This example demonstrates that the representation of pronouns is crucial for handling verb phrase ellipsis. Notice that though the elided sentence is ambiguous, it cannot mean that *George loves some other person's wife (other than Fred's or George's)*. In this way, the trigger sentence limits the meanings the elided sentence can have. This provides evidence that the interpretations of an elided sentence should be derived from the representation of its trigger sentence. Hence, the representation of a pronoun in a trigger sentence must express the ambiguous way a pronoun refers to a syntactic subject.

Lambda abstraction of syntactic subjects in logical form has been used to provide two ways for a pronoun to refer to a syntactic subject [Sag, 1976; Williams, 1977; Webber, 1978] in a trigger sentence. A pronoun that refers to a syntactic subject can act in two different ways, either as a lambda variable or as something which depends on the subject's type of noun phrase. This is needed to account for the ambiguity in Example 1. A pronoun which refers to a noun phrase can act in different ways. For instance, if a pronoun refers intrasententially to a quantified noun phrase, then the pronoun should behave like the variable associated with the quantified term. For example:

Example 2

```
Fred_i showed every girl_j her_j picture.
```

If a pronoun refers to a noun phrase in a different sentence or to some non-linguistic entity, then that pronoun behaves like a *discourse entity*[1]. For example:

Example 3

```
Trigger Sentence: Fred_i saw her_j picture.
Elided Sentence:  George_l did too.
Meaning: George saw the same girl's picture.
```

We need some way to represent the range of behaviors a pronoun exhibits. We can either use a single representation which is consistent with that variety of behaviors or develop several different representations for pronouns. We use the first approach, though past models of verb phrase ellipsis take the second. A single representation for pronouns is better because the representation for a pronoun can be determined before its antecedent is known.

Because we are concerned with building a computational model for verb phrase ellipsis, we want to develop a representation which captures the meaning of a pronoun in that domain and meets our computational goals. We claim that they are not incompatible, though in the next section, we show how past models do not conform to our computational constraints.

3 Past Approaches

Sag [1976], Webber [1978], and Williams [1977] use logical form to handle verb phrase ellipsis. We briefly summarize their models, and discuss how each approach represents pronouns in a way inconsistent with our computational guidelines. Because these models are descriptive, failure to meet our constraints does not lessen the impact of their work. However, because we are interested in a computational model, we can only borrow from the linguistic insights they offer.

All of the past models of verb phrase ellipsis define similar ways to map sentences into logical form. Each model requires the syntactic subject of a sentence to be lambda abstracted. The semantic role of every noun phrase is indicated by the position of its representation in logical form (predicate first, agent second, etc.). They represent a universal noun phrase as a universally quantified variable whose quantifier is placed outside the proposition containing it but inside the scope of the lambda operator. Sag [1976] represents indefinite noun phrases as existentially quantified variables. Webber [1978] represents all non-subject indefinites in the same way, but indefinites in subject position are represented as discourse entities. The order of quantifier placement in logical form is not used to indicate the final quantifier scoping. Quantifiers can be moved and ordered to specify quantifier scoping once it can be determined. All of the models represent a definite noun phrase as a string, with the exception of Webber who represents it either as a function if it is possessive (e.g. *Fred's wife* is represented as *(wife-of Fred)*) or as a string otherwise. The following shows the logical form (consistent with all the models) for a sentence:

Example 4

```
Fred told every girl every story.
Fred, λ(x)(∀y: (girl y) ∀z: (story z)
            (tell x z y))
```

Turning now to the representation of pronouns in logical form, we examine the approaches used by Sag [1976] and Webber [1978][2]. Sag [1976] represents a pronoun as a string with an index indicating the noun phrase to which it refers. He defines an interpretation rule to generate all possible representations of a trigger sentence when a pronoun is co-indexed with a syntactic subject. This rule

[1]A discourse entity is like a rigid designator. It can denote a group or an individual in discourse. For more on this, see [Webber, 1978].

[2]The representation described by Williams [1977] is so similar to Sag's that we do not discuss it here.

specifies that a pronoun co-indexed with a subject could *optionally* be replaced by the lambda variable associated with the subject. To use this rule, Sag assumes that indices (indicating coreference between nouns) are assigned to all noun phrases in the trigger sentence, including non-referential noun phrases like *everyone*. Sag claims that an indexed pronoun is referential, unless co-indexed with a quantified noun phrase. He defines another rule which *obligatorily* applies to a pronoun indexed with a quantified noun phrase to replace the pronoun string with a quantified variable. Sag would represent the trigger sentence in Example 1 initially by co-indexing the pronoun with its antecedent (see Example 5a). With his optional rule, he derives the second representation by replacing the pronoun string with the subject's lambda variable (see 5b).

Example 5
a. Fred$_i$, λ(x)(loves x his$_i$ wife)
   ; Fred loves Fred's wife
b. Fred$_i$, λ(x)(loves x x's wife)
   ; Fred loves his own wife

Each of these representations sanctions one interpretation of *George does too.*

Sag's representation of pronouns in logical form does not obey the three computational constraints suggested in Section 2.1. Because he does not provide a representation for a pronoun before its antecedent is known, his pronoun representation violates constraint two. Sag's optional rule is required to handle the ambiguity found in Example 1. However, the replacement of a co-indexed pronoun string with a variable violates constraint three. This augmentation is not compatible with the initial representation of a pronoun as a string. Additionally, Sag's optional rule generates 2^n distinct representations for a trigger sentence containing n pronouns that refer to the syntactic subject of a sentence. Generating this many representations of a trigger sentence violates constraint one. A more compact way to represent this ambiguity is needed.

Webber [1976] represents a pronoun initially as a string. The pronoun string is replaced by a pronoun trace equated with something. What the pronoun trace is equated with depends on the pronoun's antecedent. If the pronoun refers to a noun phrase represented as a quantified variable, the pronoun trace is equated with that variable. If the pronoun refers to a definite noun phrase, the pronoun trace is equated with a discourse entity. Webber defines an optional rule to derive an additional representation if a pronoun refers to the subject of a sentence. This rule replaces the pronoun trace with the lambda variable associated with the subject. The following shows Webber's representations for the trigger sentence in Example 1:

Example 6
a. Fred, λ(r)(love r wife-of(his))
b. Fred, λ(r)(love r wife-of(Pro = Fred$_{22}$))
c. Fred, λ(r)(love r wife-of(r))

The initial representation of *his* is shown in 6a. Once pronoun resolution occurs, the pronoun string is replaced the pronominal trace representation (shown in 6b). Finally, the bound variable interpretation (shown in 6c) is derived from the pronominal trace representation. Webber uses the representations indicated in 6b and 6c to derive the two possible interpretations of the elided sentence (obtained by applying the new subject to the two lambda functions).

Example 7
a. George, λ(r)(love r wife-of(Pro = Fred$_{22}$))
   ; George loves Fred's wife
b. George, λ(r)(love r wife-of(r))
   ; George loves George's wife

Webber's use of logical form suffers from many of the same problems that Sag's approach does. Replacement of a pronoun string by a trace equated with a variable violates constraint three. Replacement of a pronoun trace equated with a discourse entity by a lambda variable does too. Each augmentation is incompatible with the previous representation of a pronoun. Webber's model also fails to compactly represent the ambiguity in a trigger sentence caused by the reference of a pronoun to a syntactic subject, thus violating constraint one.

4 Our Representation

In this section, we develop a representation for pronouns in logical form. This representation is used in our model of verb phrase ellipsis. In our model, we lambda abstract syntactic subjects to handle verb phrase ellipsis[3]. The logical roles of all noun phrases in a sentence are identified by position in logical form (if this presents a problem, we could always use slot-filler notation for indicating the logical roles of the arguments). Following [Webber, 1978], we represent universal noun phrases as universally quantified (and restricted) terms. For brevity, we assume that indefinite noun phrases are represented as existentially quantified (and restricted) variables. We also ignore the representation of definite noun phrases in this paper, with the exception of proper nouns and possessive noun phrases[4]. Possessive noun phrases are represented as functions of the possessive noun. Proper nouns are represented as discourse entities. Quantifier scoping is handled in the same way as in the other models. Like Webber, we derive the interpretation of an elided sentence from the representation of the trigger sentence.

To be consistent with constraint two, we must develop a representation of pronouns in logical form which can be generated before their antecedents are known. To obey constraint three, we must initially represent a pronoun in a way which will be consistent with all the ways a pronoun can act. Because of the range of behaviors pronouns can adopt (shown in Section 2.2), we represent them as functions in logical form. Because the behavior of a pronoun is specified by the type of noun phrase it refers to, we claim that pronouns are like chameleons. Depending on the noun phrases to which they refer and the location of those noun phrases, the pronoun function should be equated with various values. If a pronoun refers intrasententially to a universal or indefinite noun phrase, then the pronoun function

[3] We also lambda abstract noun phrases embedded in a syntactic subject [Harper, 1987].

[4] These assumptions allow us to concentrate on the representation of pronouns. Actually in [Harper, 1987], we represent definites as functions.

is set equal to its variable. If a pronoun refers intrasententially to a syntactic subject, then the function is equated with either a lambda variable or something else depending on the type of noun phrase. If a pronoun refers to a noun phrase in a different sentence or to some non-linguistic entity, then its function is equated with a discourse entity. Next, we describe the representation of pronouns as functions.

Each pronoun function must have a unique name (supplied by adding a unique number to the pronoun). Before a pronoun function can be completely specified, however, we need to determine what its arguments should be. We could represent a pronoun as a function on all of the lambda variables and all of the quantified variables in a specific logical form. However, pronouns cannot refer to all noun phrases that occur in a sentence. Additionally, if a pronoun is a function of a lower lambda variable, that variable will not be bound in the function. Therefore, we limit the argument list of a pronoun by formalizing what noun phrases it could refer to. In a way similar to [Bach and Partee, 1980], we specify when a pronoun can refer to a noun phrase based on the location of that noun phrase's representation in logical form[5]. This information is encoded in the argument list of a pronoun function. We claim that a pronoun function is a function of all lambda variables (associated with subjects) which have scope over it in logical form, and any non-subject quantified variables which are placed in slots at the same or a higher level in logical form (i.e. not more deeply embedded in lambda-functions). We illustrate the initial representation of pronouns with some examples.

Example 8

```
Fred_i loves himself_i.
Fred_22, λ(x)(love x (himself_1 x))
```

The sentence in Example 8 contains no universal or indefinite noun phrases, so the pronoun function representing *himself* is simply a function of the lambda variable x.

Example 9

```
Fred_i persuaded every woman_j that she_j should
go.
Fred_22, λ(x)(∀y: (woman y)
              (persuade x y
                 [(she_1 x y), λ(z)(go z)]))
```

The sentence in Example 9 contains a universal noun phrase. Since the variable y is placed at the same level in logical form as the pronoun function representing *she*, that universally quantified variable must be included in the argument list, in addition to the lambda variable x.

Example 10

```
Fred_i believes he_i must speak to every woman_j.
Fred_22, λ(x)(believe x
                 [(he_1 x), λ(z)(∀y: (woman y)
                                  (speak z  y))])
```

Though the sentence in Example 10 contains a universal noun phrase, *he* is represented as a function of only the lambda variable x. The pronoun *he* cannot refer to the variable y (too deeply embedded in lambda-functions).

Example 11

```
Fred_i showed his_i mother_j her_j picture.
Fred_22, λ(x)(show x (picture-of (her_1 x))
                     (mother-of (his_1 x)))
```

The sentence in Example 11 contains no non-subject universal or indefinite noun phrases, so both of the pronouns are represented as functions of the lambda variable x.

Once it is possible to decide that a pronoun refers to a certain noun phrase, its pronoun function can be equated with a variable, a discourse entity, a pronoun function, or a function representing a possessive noun phrase. In this way, pronoun functions change their behavior depending on their antecedent. This augmentation respects the initial representation of the pronoun as a function. A pronoun function can certainly be equated with a constant, any of its arguments[6], or some pronoun function whose arguments are a subset of those in the pronoun function. Equality with a possessive function is also fine if the possessive function is a function of a discourse entity, variable, or pronoun function which are all defined at the same (or a higher) level of logical form as the pronoun function.

Consider some examples. Given that *she* refers to *every woman* in Example 9, the logical form following pronoun resolution appears below.

Example 12

```
Fred_22, λ(x)(∀y: (woman y)
              (persuade x y
                [(∧ ((she_1 x y), λ(z)(go z))
                    (= (she_1 x y) y))]))
```

Notice that since the pronoun refers to a non-subject universal noun phrase, the pronoun function (she_1 x y) is equated with the universally quantified variable y. Consider how the representation in Example 8 would be augmented after pronoun resolution:

Example 13

```
Fred_22, λ(x)(∧ (love x (himself_1 x))
              (or (= (himself_1 x) x)
                  (= (himself_1 x) Fred_22)))
```

Since *himself* refers to *Fred*, the pronoun function is equated with either the lambda variable x or $Fred_{22}$. By allowing a disjunction of equality statements, we compactly represent the ambiguous way that pronouns refer to syntactic subjects. Reconsider Example 11. The logical form after pronoun resolution follows:

Example 14

```
Fred_22, λ(x)(∧ (show x (picture-of (her_1 x))
                       (mother-of (his_1 x)))
              (= (her_1 x)
                  (mother-of (his_1 x)))
              (or (= (his_1 x) x)
                  (= (his_1 x) Fred_22)))
```

Notice that since *his* refers to the subject, the pronoun function (his_1 x) is equated with $Fred_{22}$ or x. Since *her*

[5] We provide a single representation for all types of pronouns, though we could represent reflexive, non-reflexive, and possessive pronouns differently.

[6] Or any variables lambda abstracted from the argument list of a pronoun function.

refers to *his mother*, the pronoun function (her_1 x) is equated with (mother-of (his_1 x)).

We define pronouns as functions to provide a meaning for pronouns in the initial logical form representation of a sentence. The representation of a pronoun as a function requires that we determine what its arguments are. Because we specify a way to do this, the initial representation of a pronoun is computable from syntax and local semantics (satisfying constraint two on the use of logical form). Once we know which noun phrase a pronoun refers to, we modify the logical form for that sentence in a way which is consistent with the initial representation of the pronoun as a function (satisfying constraint three). Our representation of pronouns provides a compact way of representing the ambiguous way a pronoun refers to a syntactic subject. Because of this, we use logical form in a way consistent with constraint one. In addition to satisfying the constraints, our approach also handles two examples which are troublesome to the past models. These examples are introduced next.

5 A Better Model

5.1 Previous Models' Failure

In this section, we discuss two examples which are troublesome for past approaches to verb phrase ellipsis. We concentrate on how Webber's [1978] model handles them because all of the models fail for similar reasons. The first example follows:

Example 15
Every boy_i showed his_i $mother_j$ her_j clock.

Following pronoun resolution, the pronoun *his* can be represented as either a lambda variable or a pronoun trace equated with a universally quantified variable. However, the representation of the pronoun *her* presents a problem. Because *her* refers to *his mother*, which cannot refer to some fixed mother (or set of mothers) within this sentence, [Webber, 1978] is unable to represent the meaning of this sentence[7].

A related problem arises in an example of verb phrase ellipsis[8]. This example follows:

Example 16
Trigger Sentence: $Fred_i$ showed his_i $mother_j$ her_j dog_k
Elided Sentence: $George_l$ did too.
Meanings:
1. George showed Fred's mother Fred's mother's dog.
2. George showed George's mother George's mother's dog.
3. *George showed George's mother Fred's mother's dog.

Since the pronoun *his* refers to *Fred*, it is represented in two ways. Because the pronoun *her* refers to *his mother*, which is not a syntactic subject, the pronoun trace must be equated with the discourse entity for that noun phrase (say $mother_{22}$). Thus, there are two representations of the trigger sentence:

Example 17
a. Fred, λ(x)(show x (dog-of(Pro = $mother_{22}$))
(mother-of (x)))
b. Fred, λ(x)(show x (dog-of(Pro = $mother_{22}$))
(mother-of (Pro = $Fred_{22}$)))

Each representation of the trigger sentence in this case correctly indicates the meaning of the trigger sentence. However, the two derived interpretations of the elided sentence (indicated below) do not correspond to the expected readings.

Example 18
a. George, λ(x)(show x (dog-of(Pro = $mother_{22}$))
(mother-of (x)))
; George showed George's mother Fred's
; mother's dog.
b. George, λ(x)(show x (dog-of(Pro = $mother_{22}$))
(mother-of (Pro = $Fred_{22}$)))
; George showed Fred's mother Fred's mother's
; dog.

The representation in 18b is a reasonable interpretation for the elided sentence, but the representation in 18a is not. Moreover, one of the expected interpretations (i.e. the second reading in Example 16) cannot be derived. We claim that Webber's approach fails because not all definite noun phrases can be represented as discourse entities. If a pronoun refers intrasententially to a definite noun phrase, and that noun phrase can change who it refers to, then the pronoun must be represented in a way which captures that change.

5.2 Our Success

Unlike previous models, we have no trouble representing Example 15. Because of the way we represent pronouns and possessives, our model captures the correct meaning. Before pronoun resolution, the sentence in Example 15 is represented as follows:

Example 19
∀x: (boy x) x, λ(y)(show y
(clock-of (her_1 y))
(mother-of (his_1 y)))

Given that the pronoun *his* refers to the subject and *her* refers to *his mother*, the representation is augmented as follows:

Example 20
∀x: (boy x) x, λ(y)(∧ (show y
(clock-of(her_1 y))
(mother-of (his_1 y)))
(= (his_1 y) y)
(= (her_1 y)
(mother-of (his_1 y))))

Our model also provides reasonable interpretations for

[7] In connection with her work on verb phrase ellipsis, Webber [1978] does not represent definite noun phrases as quantified terms. It is possible that with the aid of definite quantifiers, that examples 15 and 16 could be handled.

[8] [Roberts, 1987], using a completely different approach to verb phrase ellipsis, fails to handle Example 16.

Example 16. The trigger sentence is initially represented as follows:

Example 21
```
Fred22, λ(x)(show x (dog-of (her1 x))
                    (mother-of (his1 x)))
```

Given that *his* refers to *Fred*, the pronoun function can be equated with $Fred_{22}$ or the lambda variable x. Likewise, because *her* refers to *his mother* (which is not a subject), it can be equated with the function representing it. The logical form of the trigger sentence after pronoun resolution is shown below:

Example 22
```
Fred22, λ(x)(∧ (show x (dog-of (her1 x))
                       (mother-of (his1 x)))
               (or (= (his1 x) x)
                   (= (his1 x) Fred22))
               (= (her1 x)
                   (mother-of (his1 x))))
```

The representation of the trigger sentence in 22 contains two different representations of the trigger sentence. Each of these representations allows us to derive one interpretation of the elided sentence by appending the representation of the syntactic subject of the elided sentence to the representation of the verb phrase of the trigger sentence:

Example 23
```
Reading 1:
Trigger Sentence Representation:
Fred22, λ(x)(∧ (show x (dog-of (her1 x))
                       (mother-of (his1 x)))
               (= (his1 x) Fred22)
               (= (her1 x)
                   (mother-of (his1 x))))
Elided sentence Representation:
; George showed Fred's mother Fred's mother's
; dog.
George2, λ(x)(∧ (show x (dog-of (her1 x))
                        (mother-of (his1 x)))
                (= (his1 x) Fred22)
                (= (her1 x)
                    (mother-of (his1 x))))
Reading 2:
Trigger Sentence Representation:
Fred22, λ(x)(∧ (show x (dog-of (her1 x))
                       (mother-of (his1 x)))
               (= (his1 x) x)
               (= (her1 x)
                   (mother-of (his1 x))))
Elided sentence Representation:
; George showed George's mother George's
; mother's dog.
George2, λ(x)(∧ (show x (dog-of (her1 x))
                        (mother-of (his1 x)))
                (= (his1 x) x)
                (= (her1 x)
                    (mother-of (his1 x))))
```

Because we represent a possessive noun phrase as a function, we are able to represent a pronoun's reference to a possessive by equating the pronoun function with the possessive function. This allows the correct readings of the elided sentence of Example 16 to be derived.

6 Conclusion and Future Work

We have described a representation of pronouns in logical form which obeys our computational constraints. We have discussed the computability of these representations. We have also described how the representation of a pronoun can be augmented in a way consistent with its initial meaning as a function. The representation of a pronoun as a function can be augmented by equating the function with different things depending on its antecedent. In particular, we have demonstrated that when a pronoun refers to a possessive noun phrase, its pronoun function should be equated with a function. Finally, we have a compact way to represent the ambiguous way a pronoun refers to a syntactic subject. In conclusion, representing pronouns as functions not only meets the computational constraints of Section two, but also allows us to build a better model for the linguistic evidence.

One more constraint could be added to our list of computational constraints in Section two. Since we are currently implementing a program to parse sentences into logical form, we would like to have compositional rules for generating it. We are currently exploring whether the other constraints are consistent with compositional parsing.

References

[Allen, 1987] James Allen. *Natural Language Understanding.* The Benjamin/Cummings Publishing Company, Menlo Park, CA, 1987.

[Bach and Partee, 1980] Emmon Bach and Barbara Partee. Anaphora and semantic structure. In Jody Kreiman and A. E. Ojeda, editors, *Papers from the Parasession on Pronouns and Anaphora.* Chicago Linguistic Society, Chicago IL, 1980.

[Harper, 1987] Mary Harper. A model of verb phrase ellipsis. Thesis Proposal, Brown University, 1987.

[Roberts, 1987] Craige Roberts. *Modal subordination, anaphora, and distributivity.* PhD thesis, University of Massachusetts, 1987.

[Ross, 1967] J. R. Ross. *Constraints on Variables in Syntax.* PhD thesis, MIT, 1967.

[Ross, 1969] J. R. Ross. Guess who? In R. I. Binnick, A. Davison, G. Green, and J. Morgan, editors, *Papers from the Fifth Regional Meeting of the Chicago Linguistic Society.* University of Chicago, Chicago IL, 1969.

[Sag, 1976] Ivan A. Sag. *Deletion and Logical Form.* PhD thesis, MIT, 1976.

[Schubert and Pelletier, 1984] L. K. Schubert and F. J. Pelletier. From English to logic : Context-free computation of 'conventional' logical translations. *American Journal of Computational Linguistics*, 10:165–176, 1984.

[Webber, 1978] B. L. Webber. *A Formal Approach to Discourse Anaphora.* PhD thesis, Harvard, 1978.

[Williams, 1977] Edwin S. Williams. Discourse and logical form. *Linguistic Inquiry*, 8:101–139, 1977.