6. Predications

6.1. Naming and describing

6.1.0. Overview

We will now begin to study a wider variety of logical forms in which we identify components of sentences that are not also sentences.

6.1.1. A richer grammar
A variety of grammatical categories can be defined using the idea of an individual term, an expression whose function is to name an individual object.

6.1.2. Logical predicates
When the subject is removed from a sentence, a grammatical predicate is left behind; a logical predicate is what is left when any number of individual terms are removed.

6.1.3. Extensionality
The truth value of a sentence in which a predicate is applied depends only on the reference values of the terms the predicate is applied to, so the meaning of predicate is a function from reference values to truth values.

6.1.4. Identity
We will study the special logical properties of only one predicate, the one expressed by the equals sign and by certain uses of the English word is.

6.1.5. Analyzing predications
When the analysis of truth-functional structure is complete, we may go on to analyze atomic sentences as the applications of predicates to individual terms.

6.1.6. Individual terms
While individual terms are not limited to proper names, they do not include all noun phrases, only ones whose function is like that of proper names.

6.1.7. Functors
Individual terms can be formed from other individual terms by operators analogous to predicates.

6.1.8. Examples and problems
These operators enable us to continue the analysis of sentences beyond the analysis of predications by analyzing individual terms themselves.

Glen Helman 01 Aug 2013

6.1.1. A richer grammar

While there are more truth-functional connectives that we might study and more questions we might ask about those we have studied, we will now move on from truth-functional logic. The logical forms we will turn to involve ways sentences may be constructed out of expressions that are not yet sentences. Although the kinds of expressions we will identify do not correspond directly to any of the usual parts of speech, our analyses will be comparable in detail to grammatical analyses of short sentences into words.

The simplest case of this sort of analysis is related to, but not identical with, the traditional grammatical analysis into subject and predicate. You might find a grammar text of an old-fashioned sort defining *subject* and *predicate* correlatively as the part of the sentence that is being spoken of and the part that says something about it. Of course, in saying that the subject is being spoken of, there would be no intention to say that the predicate is used to say something about words. So the text might go on to say that a subject contains a word that names the "person, place, thing, or idea" (to quote one of my high school grammar texts) about which something is being said. Thus we have the situation shown in Figure 6.1.1-1.

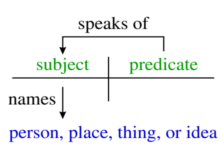

Fig. 6.1.1-1. The traditional picture of grammatical subjects and predicates.

This picture is really not adequate for either grammar or logic, but grammarians and logicians part company in the ways they refine it. Grammarians look for more satisfactory definitions of *subject* and *predicate* that still capture, at least roughly, the expressions that have been traditionally labeled in this way. Logicians, on the other hand, accept something like the definitions above and look for expressions that really have the functions they describe, whether or not these expressions would traditionally be labeled subjects and predicates.

"Subjects" and "predicates" in the logical sense provide, along with sentences and connectives, examples of two broad syntactic categories, *complete expressions* and *operators*. Sentences are examples of complete expressions and connectives are examples of operators. Like connectives, operators in general can be thought of as expressions with blanks, expressions that are incomplete in the sense that they are waiting for input. We can classify operators according to the number and kinds of inputs they are waiting for and the kind of output they yield when they receive this input. In the case of connectives, both the input and the output consists of sentences.

A "subject" in the logical sense will be a kind of complete expression, an *individual term*. This is a type of expression whose function is to refer to something; it is an expression which can be described, roughly, as naming a "person, place, thing, or idea." In 6.1.6, we will consider the full range of expressions that count as individual terms but, for now, it will be enough to have in mind two basic kinds of example—proper names (such as Socrates, Indianapolis, Hurricane Isabel, or 3) and simple definite descriptions formed from the definite article the and a common noun (such as the winner, the U.S. president, the park, the book, or the answer).

In the simplest case, a "predicate" in the logical sense—and this is what we will use the term *predicate* to speak of—is an expression that can be used to say something about the object referred to by an individual term. It is an operator whose input is an individual term and whose output is a sentence expressing what is said. Thus a logical predicate amounts to a sentence with a blank waiting to be filled by an individual term. In 6.1.2, we will move beyond this simple case to include predicates that require multiple inputs (i.e., that have several blanks to be filled). Such predicates are certainly not predicates in the grammatical sense; nonetheless a logical predicate will contain the main verb of any sentence it yields as output, so many of the simplest examples of predicates will correspond to verbs or verb phrases.

The categories of expressions we are working with now include the ones listed below (with simple examples in the style of some popular early elementary school readers from the mid-20th century):

Complete expressions

expression	*examples*
sentence	Jane ran, Spot barked, Jane ran and Spot barked
individual term	Jane, Spot

Operators

operator	*input*	*output*	*examples*
connective	sentence(s)	sentence	_ and _
predicate	individual term(s)	sentence	_ ran, _ barked

Since we now have a number of kinds of expression that might be input or output of an operator, there are many more sorts of operators that can be distinguished according to their input and output, and we will go on to consider some of them. For example, in 6.1.7, we will add a kind of operator which yields individual terms as output (for individual terms as input). The input and output of operators need not be limited to complete expressions, and in later chapters, we will add operators that take predicates as input.

Glen Helman 01 Aug 2013

6.1.2. Logical predicates

We derived the concept of an individual term from a traditional description of the grammatical subject of a sentence by focusing on the semantic idea of naming. As we will see in 6.1.6, the idea of an individual term is much narrower than the idea of a grammatical subject: not every phrase that could serve as the subject of a sentence counts as an individual term. We have seen that the opposite is true of our concept of a predicate: it includes grammatical predicates but many other expressions, too.

Like the definition of an individual term, the definition of a logical predicate is semantic: a predicate says something about the about whatever objects are named by the individual terms to which it is applied. The simplest example of this is a grammatical predicate that says something about an object named by an individual term. But consider a sentence that has not only a subject but also a direct object—Ann met Bill for example. This says something about Ann, but it also says something about Bill. From a logical point of view, we could equally well divide the sentence into the subject Ann on the one hand and the predicate met Bill on other or into the subject-plus-verb Ann met and the direct object Bill. And we will be most in the spirit of the idea that predicates are used to say something about individuals if we divide the sentence into the two individual terms Ann and Bill on the one hand and the verb met on the other. The subject and object both are names, and the verb says something about the people they name. That is why we define a *predicate* as an operator that forms a sentence when applied to one or more terms. We will speak of the application of this operator as *predication* and speak of a sentence that results as *a predication*.

We can present predicates in this sense graphically by considering sentences containing any number of blanks. For example, the predication Jane called Spot might be depicted as follows:

Individual terms: Jane Spot
Predicate: ____ called ____

The number of different terms to which a predicate may be applied is its number of *places*, so the predicate [_ called _] has 2 places while predicates, like [_ ran] and [_ barked], that are also predicates in the grammatical sense will have one place. We will discuss our notation for predicates more in 6.2.1, but we will often (as has been done here) indicate a predicate by surrounding with brackets the English sentence-with-blanks that expresses it.

In the example above, the two-place predicate is a transitive verb and the second individual term functions as its direct object in the resulting sentence. The individual terms that serve as input to predicates may also appear as indirect objects or as the objects of prepositional phrases that modify a verb—as in the following examples:

Individual terms: Jane Spot the ball
Predicate: ____ threw ____ ____

Individual terms: the ball the window the fishbowl
Predicate: ____ went through ____ into ____

Other examples of many-place predicates are provided by sentences containing comparative constructions or relative terms. Even conjoined subjects can indicate a many-place predicate when and is used to indicate the terms of a relation rather than to state a conjunction:

Individual terms: Jane Sally
Predicate: ____ is older than ____

Individual terms: 2 5
Predicate: _ < _

Individual terms: Jane Sally
Predicate: ____ is a sister of ____

Individual terms: Jane Sally
Predicate: ____ and ____ are sisters

Although you will rarely run into predicates with more than three or four places, it is not hard make up examples of predicates with arbitrarily large numbers of places. For example, imagine the predicate you would get by analyzing a sentence that begins Sam travelled from New York to Los Angeles via Newark, Easton, Bethlehem, ... and goes on to state the full itinerary of a trans-continental bus trip.

The places of a many-place predicate come in a particular order. For example, the sentences Jane is older than Sally and Sally is older than Jane are certainly not equivalent, so it matters which of Jane and Sally is in the first place and which in the second when we identify them as the inputs of the predicate [_ is older than _]. Even when the result of reordering individual terms is equivalent to the original sentence, we will count the places as having a definite order and treat any reordering of the terms filling them as a different sentence. So Dick is the same age as Jane and Jane is the same age as Dick will count as different sentences even though [_ is the same age as _] is symmetric in the sense that

$$\sigma \text{ is the same age as } \tau \simeq \tau \text{ is the same age as } \sigma$$

for any terms σ and τ.

Glen Helman 01 Aug 2013

6.1.3. Extensionality

The only restriction on an analysis of a sentence into a predicate and individual terms is that the contribution of an individual term to the truth value of a sentence must lie only in what we will call its *reference value*. That is, what matters for the truth value is only what a term names if it names something; and, if it names nothing, that is all that matters. The particular way it refers to what it refers to, or the way in which it fails to refer if reference fails, do not matter. Both truth values and reference values are extensions in the sense discussed in 2.18, so the predicates we will consider are like truth-functional connectives in being *extensional operators*: the extension of their output depends only on the extensions of their inputs.

In the specific case of predicates, this requirement is sometimes spoken of as a requirement of *referential transparency*. When it is satisfied, we can look through individual terms and pay attention only to their reference values when judging whether a sentence is true or false; in other cases, we might need to pay attention to the terms themselves or to the ways in which they refer to their values in order to judge the truth value. For example, in deciding the truth of The U. S. president is over 40, all that matters about the individual term the U. S. president is who it refers to. On the other hand, the sentence For the past two centuries, the U. S. president has been over 35 is true while the sentence For the past two centuries, Barack Obama has been over 35 is false—even when the terms the U. S. president and Barack Obama refer to the same person. So, in this second case, we must pay attention to differences between terms that have the same reference value. When this is so the occurrences of these terms are said to be *referentially opaque*; that is, we cannot look through them to their reference values. The restriction on the analysis of sentences into predicates and individual terms is then that we can identify an expression as an individual term filling a place of a predicate only when that occurrence of the expression is referentially transparent. Occurrences that are referentially opaque must remain part of the predicate because more than just their reference values are needed for determining the truth value of a predication.

Hints of idea of a predicate as an incomplete expression can be found in the Middle Ages, but it was first developed explicitly by Gottlob Frege in the late 19th century. Frege applied the idea of an incomplete expression not only to predicates but also to mathematical expressions for functions. Indeed, Frege spoke of predicates as signs for a kind of function, a function whose value is not a number but rather a truth value. That is, just as a function like + takes numbers as input and issues a number as output, a predicate is a sign for a function that takes the possible references of individual terms as input and issues a truth value as output by saying something true or false about the input.

We will speak of the truth-valued function associated with a 1-place predicate as a *property* and speak of the function associated with a predicate of two or more places as a *relation*. Thus a predicate is a sign for a property or relation in the way a truth-functional connective is a sign for a truth function.

Just as a truth-functional connective can be given a truth table, the extensionality of predicates means that a table can capture the way the truth values of the their output sentences depend on the reference values of their input. For example, consider the predicate __ divides __ (evenly). Just as there can be addition or multiplication tables displaying the output of arithmetic functions for a limited range of input, we can give a table indicating some of the output of the relation expressed by this predicate. For the first half dozen positive integers, we would have the table shown below. Here the input for the first place of the predicate is shown by the row labels at the left and the input for the second place by the column labels at the top. The first row of the table then shows that 1 divides all six integers evenly, the second row shows that 2 divides only 2, 4, and 6 evenly, and the final column shows that each of 1, 2, 3, and 6 divides 6 evenly.

_ divides _	1	2	3	4	5	6
1	T	T	T	T	T	T
2	F	T	F	T	F	T
3	F	F	T	F	F	T
4	F	F	F	T	F	F
5	F	F	F	F	T	F
6	F	F	F	F	F	T

Of course, this table does not give a complete account of the meaning of the predicate; and, for many predicates, no finite table could. But such tables like this will still be of interest to us because we will consider cases where there are a limited number of reference values and, in such cases, tables can give full accounts of predicates.

Further questions arise when we recognize the fact that some terms do not refer. Such terms still have a reference value but one of special sort that we will describe as a *nil value*. A term which has such a reference value will be said to be an *undefined term*. In general, we will treat undefined terms just as we treat other terms, but they require special consideration for a couple of reasons. First, this is one of the places where the issue of semantic presuppositions arises; and, after implicatures, the non-deductive inferences associated with semantic presuppositions are the ones that are most difficult to distinguish from deductive inference. The second reason is related: we will eventually consider the logical properties of definite descriptions and, as was noted in 1.3.7, it is not universally agreed which inferences concerning them are deductive and which derive from semantic presupposition.

Although it is far from universally agreed, we will assume that it is built into the idea of extensionality that all terms that fail to refer should have the same reference value. That is the basis for assigning them not merely a special sort of value—i.e., a nil value, but assigning to all of them the same reference value, which we will refer to as *the Nil*. We assume that sentences have truth values in all possible worlds, even when they contain terms, like definite descriptions, that do not refer to anything under certain circumstances. This means that we must assume that predicates yield a truth value as output even when the Nil is part of their input; that is, we assume that predicates are *total*. The truth value that is issued as output when the input includes the Nil is usually not settled by the ordinary meaning of an English predicate. Indeed, when non-referring terms are understood to have a nil reference value, this case is like the case of a category mistake (again see 1.3.7 for this idea); the Nil is just not the sort of thing of which most predicates are naturally true or false. As in other cases where truth values are not determined solely by sentences and possible worlds, we will assume that they are somehow stipulated for predications of the Nil but we will avoid considering relations among sentences that depend on the way these values are stipulated.

6.1.4. Identity

We used special notation for all the connectives that figured in our analyses of logical form, and they all had logical properties that we studied. However, only one predicate will count as *logical vocabulary* in this sense. Other predicates and all unanalyzed individual terms will be, like unanalyzed component sentences, part of the *non-logical vocabulary*, and they will be assigned meanings only when we specify an interpretation of this vocabulary.

The one predicate that is part of our logical vocabulary will be referred to as *identity*. It is illustrated in the following sentences:

Barack Obama is the U.S. president

The winner was Funny Cide

n = 3

The morning star and the evening star are the same thing.

Sentences like these are *equations*. Equations are thus a special kind of predication.

In our symbolic notation, we will follow the third example and use the sign = to mark identity. As English notation, we will use the word is. We will represent unanalyzed individual terms by lower case letters, so we can analyze the sentences above as follows:

Barack Obama is the U.S. president
Barack Obama = the U.S. president

o = p
o is p

o: Barack Obama; p: the U.S. president

The winner was Funny Cide
the winner = Funny Cide

w = f
w is f

f: Funny Cide; w: the winner

n = 3

n = t
n is t

n: n; t: 3

The morning star and the evening star are the same thing
the morning star = the evening star

m = e
m is e

m: the morning star; e: the evening star

Once in symbolic form, these equations are very simple. The greater complexity found in most interesting mathematical equations is due to the complexity of the individual terms they contain. To exhibit that complexity in our analyses, we will need to analyze individual terms, something we will begin to do in 6.1.7.

6.1.5. Analyzing predications

Apart from the special case of equations, our symbolic notation for predications will identify the predicate first followed by a list of the individual terms that are its input. This is a departure from English word order in most cases, but we can present analyses in this way even before we introduce symbols. The example below presents the analysis of a predication into a predicate and individual terms as a series of steps.

	Bill introduced himself to Ann
Identify (referentially transparent) occurrences of individual terms within the sentence, making sure they are all independent by replacing pronouns by their antecedents	Bill introduced Bill to Ann
Pull the terms out of the sentence	Bill Bill Ann ___ introduced ___ to ___
Preserve the order of the terms, and form a predicate from the remainder of the sentence	Bill Bill Ann [_ introduced _ to _] ___ ___ ___
Put the terms into the places of the predicate	[_ introduced _ to _] Bill Bill Ann

Underlining will often be used, as it is here, to mark the places of predicates when they are filled by English expressions. In examples and answers to exercises, we will move directly from the second of these steps to the last, so the process can be thought of as one of removing terms, placing them (in order and with any repetitions) after the sentence they are removed from, and enclosing sentence-with-blanks in brackets.

In general, an application of an *n*-place predicate θ to a series of *n* individual terms $\tau_1, \ldots, \tau_n$ takes the form

$$\theta\tau_1\ldots\tau_n$$

and our English notation is this:

θ fits (series) $\tau_1, \ldots,$ ən τ_n

The use of the verb fit here is somewhat artificial. It provides a short verb that enables $\theta\tau_1\ldots\tau_n$ to be read as a sentence, and it is not too hard to understand it as saying that θ is true of $\tau_1, \ldots, \tau_n$. Another artificial aspect of this notation is the unemphasized form ən of and, which is designed to distinguish the use of and here to join the terms of a relation from its use as a truth-functional connective. The role of the term series, which will rarely be needed, is discussed in 6.1.7. We will use the general notation $\theta\tau_1\ldots\tau_n$ when we wish to speak of all predications, so we will take it to apply to equations, too, even though the predicate = is written between the two terms to which it is applied.

In our fully symbolic analyses, unanalyzed non-logical predicates will be abbreviated by capital letters. This fits with the use of capital letters for unanalyzed sentences since predicates have sentences as their output. When we add non-logical operators that yield individual terms as output, they will be abbreviated by lower case letters just as unanalyzed individual terms are.

As was done in the display above, we will use the Greek letters θ, π, μ, and ρ to refer to stand for any predicates, so they may stand for single letters and for =. The may also stand for complex predicates whose internal structure has been analyzed, something we will go on to consider in 6.2.1. We will also go on to consider compound terms, and we will use the Greek letters τ, σ, and υ to stand for any terms, simple or compound.

If we complete the analysis of Bill introduced himself to Ann, carrying it into fully symbolic form and restating it in English notation, we would get the following:

Bill introduced himself to Ann
Bill introduced Bill to Ann
[_ introduced _ to _] Bill Bill Ann
Tbba
T fits b, b, ən a
T: [_ introduced _ to _]; a: Ann; b: Bill

Notice that the bracketed English sentence-with-blanks does not appear in the final analysis, but it does appear in the key.

When sentences contain truth-functional structure, that structure should be analyzed first; an analysis into predicates and individual terms should begin only when no further analysis by connectives is possible. Here is an example:

If either Ann or Bill was at the meeting, then Carol has seen the report and will call you about it
Either Ann or Bill was at the meeting → Carol has seen the report and will call you about it
(Ann was at the meeting ∨ Bill was at the meeting)
→ (Carol has seen the report ∧ Carol will call you about the report)
([_ was at _] Ann the meeting ∨ [_ was at _] Bill the meeting)
→ ([_ has seen _] Carol the report
∧ [_ will call _ about _] Carol you the report)

(Aam ∨ Abm) → (Scr ∧ Lcor)
if either A fits a ən m or A fits b ən m
then both S fits c ən r and L fits c, o, ən r

A: [_ was at _]; L: [_ will call _ about _]; S: [_ has seen _]; a: Ann; b: Bill; c: Carol; m: the meeting; o: you; r: the report

When analyzing atomic sentences into predicates and terms, be sure to watch for repetitions of predicates from one atomic sentence to another—such as that of [_ was at _] in this example. Such repetitions are an important part of the logical structure of the sentence.

Since the notation for identity is different from that used for non-logical predicates, you need to watch for atomic sentences that count as equations. These will usually, but not always, be marked by some form of the verb to be but, of course, forms of to be have other uses, too. Consider the following example:

If Tom was told of the nomination, then if he was the winner he wasn't surprised
Tom was told of the nomination → if Tom was the winner he wasn't surprised
Tom was told of the nomination
→ (Tom was the winner → Tom wasn't surprised)
Tom was told of the nomination
→ (Tom was the winner → ¬ Tom was surprised)
[_ was told of _] Tom the nomination
→ (Tom = the winner → ¬ [_ was surprised] Tom)

Ltn → (t = r → ¬ St)
if L fits t ən n then if t is r then not S fits t

L: [_ was told of _]; S: [_ was surprised]; t: Tom; n: the nomination

It is fairly safe to assume that a form of to be joining two individual terms indicates an equation, but it is wise to always think about what is being said: an equation is a sentence that says its component individual terms have the same reference value. A use of to be joining noun phrases will indicate an equation only when these noun phrases are individual terms; the conditions under which that is so are discussed in the next subsection. Finally, notice that no identity predicate should appear in the key to the analysis. That is because it is part of the logical vocabulary; as such, it is like the connectives, which also do not appear in keys.

Glen Helman 01 Aug 2013

6.1.6. Individual terms

The chief examples of individual terms are proper names, for the central function of a proper name is to refer to the bearer of the name. But a proper name is not the only sort of expression that refers to an individual; a phrase like the first U. S. president serves as well as the name George Washington. In general, descriptive phrases coupled with the definite article the at least purport to refer of individuals. These phrases are the *definite descriptions* discussed briefly in 1.3.7, and we have been counting them as individual terms. Still other examples of individual terms can be found in nouns and noun phrases modified by possessives—for example, Mt. Vernon's most famous owner. Indeed, expressions of this sort can generally be paraphrased by definite descriptions (such as the most famous owner of Mt. Vernon). A final group of examples are demonstrative pronouns this and that and other pronouns whose references are determined by the context of use—such as I, you, and certain uses of third person pronouns. On the other hand, while *anaphoric pronouns*—i.e., pronouns that have other noun phrases as their antecedents—count grammatically as individual terms, they do not have independent reference values and will be treated differently in our analyses. We will look at their role more closely in 6.2.3; for now, it is enough to note that they raise issues for the analysis of predications that are analogous to the issues they raise for the analysis of truth-function compounds.

There is no traditional grammatical category or part of speech that includes individual terms but no other expressions. In particular, the class of nouns and noun phrases is too broad because it includes simple common nouns, such as president, as well as *quantifier phrases*—such as no president, every president, or a president. And neither common nouns nor quantifier phrases make the kind of reference that is required for an individual term.

Even before we look further at the reasons why this is so, we can distinguish individual terms from other nouns and noun phrases by thinking of them as answers to a which question. If you are asked Which person, place, thing, or idea are you referring to? and you reply with any of the individual terms, you have answered the question directly. On the other hand, a common noun by itself is ungrammatical as an answer, and a quantifier phrase does not provide a direct answer. While a president, no president, and every president are grammatical replies to the question Which person are you referring to?, the first two provide only an incomplete or evasive answers, and the third indicates that the question cannot be answered as asked.

The following table collects the examples we have just seen on both sides of the line between individual terms and other noun phrases:

Individual terms	*Not individual terms*
proper names George Washington	common nouns president
definite descriptions the first U. S. president	quantifier phrases no president, every president, a president
noun phrases with possessive modifiers Mt. Vernon's most famous owner	
non-anaphoric pronouns this, you	
anaphoric pronouns he, she, it	

Perhaps the most that can be done in general by way of defining the idea of an *individual term* is to give the following rough semantic description: an individual term is

an expression that refers (or purports to refer)
to a single object in a definite way

At any rate, this formula can be elaborated to explain the reasons for rejecting the noun phrases at the right of the table above.

The formula is intended as a somewhat more precise statement of the idea that an individual term "names a person, place, thing or idea." It uses object in place of the list person, place, thing, or idea partly for compactness and partly because that list is incomplete. Indeed it would be hard to ever list all the kinds of things that might be referred to by individual terms. If the term *object* and other terms like *entity*, *individual*, and *thing* are used in a broad abstract sense, they can apply to anything that an individual term might refer to. In particular, in this sort of usage, these terms apply to people. The main force of the formula above then lies in the ideas of *referring to a single thing* and *referring in a definite way*.

The requirement that reference be to a single thing rules out most of noun phrases on the right of the table above. First of all, if a common noun by itself can be said to refer at all, it refers not to a single thing but to a class, such as the class of all presidents. Now this class can be thought of as a single thing and can be referred to by the definite description just used—i.e., the class of all presidents—but the common noun president "refers" to this class in a different way. Common nouns are sometimes labeled *general terms* and distinguished from *singular terms*, an alternative label for individual terms. The function of a general term is to indicate a general kind (e.g., dogs) from which individual things may be picked out rather than to pick out a single thing of that kind (e.g., Spot), as an individual term does. Thus the individual term the first U. S. president picks out an individual within the class indicated by the common noun president; and the class of all presidents picks out an individual within the class indicated by the common noun class. That is, a general term indicates a range of objects from which a particular object might be chosen while an individual term picks out a particular object. Although there is much that might be said about the role of general terms in deductive reasoning, we will never identify them as separate components in our analyses of logical form, and the word term without qualification will be used as an abbreviated alternative to individual term.

The remaining noun phrases at the right of the table are like individual terms in making use of a common noun's indication of a class of objects. However, they do not do this to pick out a single member of the class but instead to help make claims about the class as a whole. The claims to which they contribute say something about the number of members of a class that have or lack a certain property, and that is the reason for describing them as "quantifier" phrases.

It's probably clear that the phrases every president and no president, even though they are grammatically singular, do not serve the function of picking out a single object. But that may be less clear in the case of a president. Sentences containing quantifier phrases like a president and some president share with those containing definite descriptions, such as the president, the feature that they can be true because of a fact about a single object. For example, The first U. S. president wore false teeth and A president wore false teeth can be said to both be true because of a fact about Washington. The difference between the two sorts of expression can be seen by considering what might make such sentences false. If Washington had not worn false teeth, The first U. S. president wore false teeth would be false but A president wore false teeth might still be true. That's because the second could be true because of facts about many different presidents (in many different countries), so its truth is not tied to facts about any one of them. If the expression a president is thought of as referring at all, its reference is an indefinite one. That is one reason for adding the qualification definite to the formula for individual terms given above, but this qualification also serves as a reminder that the presence of a definite article is a mark of an individual term while an indefinite article indicates a quantifier phrase.

Glen Helman 01 Aug 2013

6.1.7. Functors

Truth-functional connectives express truth-valued functions of truth values, and predicates express truth-valued functions of reference values. A third sort of function not only takes reference values as input but also issues them as output. We will refer to this sort of function as a *reference function* or, in contexts where we do not need a more general concept, simply as a *function*. We will refer to expressions that are signs for these functions as *functors* and refer to the operation of applying a functor simply as a *functor application*. The result of a functional application will count as a *compound term*.

Functors are incomplete expressions that stand to individual terms as connectives stand to sentences, so we can extend the table of operators in 6.1.1 as follows:

operator	*input*	*output*
connective	sentence(s)	sentence
predicate	individual term(s)	sentence
functor	individual term(s)	individual term

We will add further incomplete expressions to this list in later chapters when we consider operators that take predicates as input.

Signs for mathematical functions provide examples of functors. The expression 7 + 5 can be analyzed as

Individual terms: 7   5
Functor: _ + _

But functors are not limited to mathematical vocabulary. Any individual term that contains one or more individual terms can be seen as the result of applying a functor to those component terms. Thus the oldest child of Ann and Bill can be analyzed as

Individual terms: Ann   Bill
Functor: the oldest child of ___ and ___

And the more complex individual term the book that Ann's father mentioned has the following analysis:

Individual term: Ann
Functors: ___'s father
the book that ________ mentioned

Possessives and prepositional phrases often give rise to functors but all that is needed to have a functor is an individual term that contains an individual term.

Our notation for functors will be analogous to that for predicates. Functors can be represented in semi-symbolic notation by individual-terms-with-blanks surrounded by brackets. Using this notation, the first two examples above could be given the analyses:

[_ + _] 7 5
[the oldest child of _ and _] Ann Bill

In the case of the third example, we will use parentheses to show grouping

[the book that _ mentioned] ([_'s father] Ann)

In fact, there is no danger of ambiguity here; but the structure is clearer with parentheses, and, in the full symbolic notation, compound terms should be enclosed in parentheses when they fill a place of a functor or predicate.

In that notation, unanalyzed functors will be represented by lower case letters and will be written before the individual terms filling their places. The general form of a compound term is this

$$\zeta\tau_1\ldots\tau_n$$

and our English notation will be

$$\zeta \text{ of (series) } \tau_1, \ldots, \text{ən } \tau_n$$

or

$$\zeta \text{ applied to (series) } \tau_1, \ldots, \text{ən } \tau_n$$

both of which are in keeping with the usual way of reading a functional application, but one or the other will work better in certain contexts. When we need a general variable for functors we will use ζ or ξ.

Using this symbolic and English notation, we can express the final analyses of the examples above as follows:

symbolic notation	*English notation*	*key*
psf	p of s ən f	p: [_ + _]; f: 5; s: 7
oab	o of a ən b	o: [the oldest child of _ and _]; a: Ann; b: Bill
b(fa)	b of f of a	b: [the book that _ mentioned]; f: [_'s father]; a: Ann

The symbolic notation for functors that is used here is designed to minimize parentheses and commas and is fairly common in work on logic, but it is different from the most common mathematical notation for functional applications. The general rule for interpreting it is this: (i) after a predicate—i.e., after a capital letter—each unparenthesized letter and each parenthetical unit occupies one place of the predicate and (ii) within a parenthetical unit the first letter is a functor and each following unparenthesized letter and each parenthetical unit occupies one place of this functor.

Here are some examples for comparison

common mathematical notation	*symbolic notation used here*	*English notation*
f(a)	fa	f of a
f(a, b)	fab	f of a ən b
f(g(a))	f(ga)	f of g of a
f(a, g(b))	fa(gb)	f of a ən g of b
f(g(a), b)	f(ga)b	f of series g of a ən b
f(g(a, b))	f(gab)	f of g of series a ən b

The last two examples above show the role of the optional term series in avoiding ambiguity. Because the letters used to represent functors and non-logical predicates do not have a fixed number of places associated with them, when a single ən follows two occurrences of of, it can be unclear where the series of terms marked by ən actually began. There are other ways of handling this ambiguity. Parentheses suffice in written notation and parentheses, like other punctuation, can be reflected in speech. For example, it is natural to mark the difference between f of (g of a) ən b and f of (g of a ən b), respectively, by varying the speed with which they are spoken in ways that might be indicated by "f of g-of-a ən b" and "f of g of a-ən-b".

In the presence of functors, the potential for undefined terms increases considerably. Even if the cat on the mat has a non-nil reference value, the cat on the refrigerator may not—to say nothing of the cat on the house of Ann's father's best friend or the cat on 6. That is, functors accept a large variety of inputs and can be expected to issue output with undefined reference for some of them. This problem can be reduced (though not eliminated) by limiting functors to input of certain sorts. That is usually done by assigning individual terms to various *types* and allowing only individual terms of certain types to serve as inputs to a given functor. For example, the functor [_ + _] might be restricted to numerical input. We will not follow this approach (which complicates the description of logical forms considerably), but it does capture a number of features, both syntactic and semantic, of a natural language like English.

6.1.8. Examples and problems

We will begin with a couple of extended but straightforward examples.

If Dan is the winner and Portugal is the place he would most like to visit, he will visit there before long
Dan is the winner and Portugal is the place he would most like to visit → Dan will visit Portugal before long
(Dan is the winner ∧ Portugal is the place Dan would most like to visit) → Dan will visit Portugal before long
(Dan is the winner ∧ Portugal is the place Dan would most like to visit) → Dan will visit Portugal before long
(Dan = the winner ∧ Portugal = the place Dan would most like to visit) → [_ will visit _ before long] Dan Portugal
(d = n ∧ p = [the place _ would most like to visit] Dan) → Vdp

(d = n ∧ p = ld) → Vdp
if both d is n and p is l of d then V fits d ən p

V: [_ will visit _ before long]; l: [the place _ would most like to visit]; d: Dan; n: the winner; p: Portugal

Al won't sign the contract Barb's lawyer made out without speaking to his lawyer
¬ Al will sign the contract Barb's lawyer made out without speaking to his lawyer
¬ (Al will sign the contract Barb's lawyer made out ∧ ¬ Al will speak to his lawyer)
¬ (Al will sign the contract Barb's lawyer made out ∧ ¬ Al will speak to Al's lawyer)
¬ ([_ will sign _] Al the contract Barb's lawyer made out ∧ ¬ [_ will speak to _] Al Al's lawyer)
¬ (S a (the contract Barb's lawyer made out) ∧ ¬ P a (Al's lawyer))
¬ (S a ([the contract _ made out] Barb's lawyer) ∧ ¬ P a ([_'s lawyer] Al))
¬ (S a (c ([_'s lawyer] Barb)) ∧ ¬ Pa(la))

¬ (Sa(c(lb)) ∧ ¬ Pa(la))
not both S fits a ən c of l of b and not P fits a ən l of a

P: [_ will speak to _]; S: [_ will sign _]; c: [the contract _ made out]; l: [_'s lawyer]; a: Al; b: Barb

When analyzing either a predication or an individual term, make sure that you remove all the largest individual terms it contains. That is, if you identify a component individual term, make sure that it is not part of a compound term that is itself a component of the sentence or term you are analyzing. To analyze Al will speak to his lawyer as [_ will speak to _'s lawyer] Al Al would be to ignore an important aspect of its structure. Of course, when applying this maxim, it is important to distinguish individual terms from other noun phrases. For example, although Dan is the winner of the contest can be analyzed initially as Dan = the winner of the contest, the grammatically similar sentence Dan is a winner of the contest should be analyzed as [_ is a winner of _] Dan the contest because a winner of the contest is not an individual term.

Also, when you locate a definite description, make sure that you have identified the whole of it. What you are most likely to miss are modifiers, usually prepositional phrases or relative clauses, that follow the main common noun of the definite description. For example, although the place might be an individual term in its own right in other cases, in the example above is it only part of the term the place Dan would most like to visit. Similarly, the contract is only the beginning of the individual term the contract Barb's lawyer made out. In both of the these cases, the rest of the definite description is a relative clause with a suppressed relative pronoun; that is, they might have been stated more fully as the place that Dan would most like to visit and the contract that Barb's lawyer made out, respectively. It might help here to think of prepositional phrases and relative clauses as modifying a common noun before the definite article is attached. That is, the phrases above have the form the (place Dan would most like to visit) and the (contract Barb's lawyer made out), so any component of these sentences containing the initial the must also contain the whole of the following parenthesized expressions.

There are some cases where a prepositional phrase or relative clause following a common noun should not be counted as part of a definite description. Some prepositional phrases can modify both nouns and verbs, and a prepositional phrase following a noun within a grammatical predicate might be understood to modify either it or the main verb. The sentence The dog chased the cat on the mat is ambiguous in this way since the mat might be understood to be either the location of the chase or the location of the cat, who might have been chased elsewhere. This sort of ambiguity can be clarified by converting the prepositional phrase into a relative clause, which can only modify a noun; if this transformation—e.g.,

The dog chased the cat that is on the mat

—preserves meaning, then the prepositional phrase is part of the definite description. On the other hand, since anaphoric pronouns cannot accept modifiers, replacing a possible noun phrase by a pronoun will produce a sentence in which a prepositional phrase unambiguously modifies the verb. This can be done by moving the noun phrase to the front of the sentence, joining it to the remaining sentence-with-a-blank by the phrase is such that, and filling the blank with an appropriate pronoun (he, she, or it). In this example, that would give us

The cat is such that the dog chased it on the mat

So, the prepositional phrase on the mat should be taken to modify cat or chased depending on whether the first or second of the displayed sentences best captures the meaning of the original. Of course, when a potentially ambiguous sentence is taken out of context, it may not be clear which of two alternatives does best capture the original meaning; in such a case, either analysis is a possible interpretation and the difference between them shows what further information is needed in order to determine what was meant.

Not all relative clauses contribute to determining reference. Those that do are *restrictive* clauses, and it is these that should be included in definite descriptions. Other relative clauses are *non-restrictive*. Non-restrictive clauses cannot use the word that and, when punctuated, are marked off by commas. Restrictive clauses are not marked off by commas in standard English punctuation and may use that (but are not limited to this relative pronoun), and they can in some cases be expressed without a relative pronoun. It is easiest to tell what sort of relative clause you are faced with when more than one of these differences is exhibited. For example, the relative clause The cat that the dog had chased was asleep or The cat the dog had chased was asleep is clearly restrictive while the one in The cat, who the dog had chased, was asleep is clearly non-restrictive. This means that the relative clause in the first is part of the definite description the cat that the dog had chased. The relative clause in the second would instead be analyzed as a separate conjunct to give the dog had chased the cat ∧ the cat was asleep as the initial step of the analysis.

Another indication of the difference between the two sorts or relative clause is that a non-restrictive clause can modify a proper name—as in Puff, who the dog had chased, was asleep. And, since neither prepositional phrases nor restrictive relative clauses can modify a proper name, putting a proper name in a blank that was left when you removed an apparent individual term can show whether you really removed the whole of the term. For example, Puff on the mat was asleep and Puff that the dog had chased was asleep are both ungrammatical.

6.1.s. Summary

1 We move beyond truth-functional logic by recognizing complete expressions other than sentences and operators other than connectives. Our additions are motivated by a traditional description of grammatical subjects and predicates. The new complete expressions are individual terms, whose function is to name. Given this idea, we can define a predicate as an operator that forms a sentence from one or more individual terms.

2 A predicate corresponds to an English sentence with blanks that might be filled by terms. These blanks are the predicate's places and the operation of filling them is predication.

3 We will maintain something analogous to truth-functionality by requiring that predicates be extensional. This means that all places of a predicate must be referentially transparent (rather than referentially opaque): when judging the truth value of a sentence formed by the predicate, we must be able see through the terms filling these places to what those terms refer to. Thus, just as a connective expresses a truth function, a predicate expresses a function that takes reference values as input and issues truth values as output. Such a function may be called a property if it has one place and a relation if it has 2 or more. In symbolic notation, it takes the form $\sigma = \tau$ and, in English notation, it takes the form σ is τ.

4 While recognizing quite a variety of non-logical vocabulary in our analyses, we recognize only one new item of logical vocabulary, the predicate identity. This is a 2-place predicate that forms an equation, which is true when its component terms have the same reference value.

5 In our symbolic notation, we use lower case letters to stand for unanalyzed individual terms, the equal sign for identity, and capital letters to stand for non-logical predicates. Non-logical predicates, both capital letters and predicate abstracts are written in front of the terms they apply to (with a predicate abstract enclosed in brackets), and = is written between the terms to which it applies. In English notation, predications other than equations are written as θ fits τ or θ fits (series) $\tau_1, \ldots,$ ən τ_n.

6 In addition to proper names, the individual terms include definite descriptions and various non-anaphoric pronouns. They do not include certain other noun phrases, quantifier phrases in particular. We will speak of the "person, place, thing, or idea" referred to by an individual term by using such words as object, entity, individual, and thing, understanding these to apply to anything that might be named. Common nouns are also not individual terms. Indeed, they may be labeled general terms to distinguish their function of indicating a class of objects from the function of individual terms, also called singular terms, which is to refer to a single individual in a definite way. The word term will often be used as shorthand for individual term.

7 A functor is an operator that takes one or more individual terms as input and yields an individual term as output. Just like other operators, it expresses a function, in this case a reference function, which yields reference values when applied to reference values. Although a reference function is a particular sort of function, so the latter term is more general, we will use it term primarily for reference functions. The operation of combining a functor with input is application, and the individual term that is the output is a compound term, for which we use the symbolic notation $\zeta\tau_1\ldots\tau_n$ and the English notation ζ of τ or ζ of (series) $\tau_1, \ldots,$ ən τ_n. (The phrase applied to is sometimes a more convenient alternative to of.) For any functor, there will almost always be some terms for which the application of the functor yields an undefined term. Although this problem can be reduced by limiting the input of functors to objects of certain types, we will not include this complication in our account of logical forms.

8 It can be difficult to recognize the individual terms that fill the places of a predicate or a functor. It is important in include in a definite description all the modifiers that are part of it. Some of these may be prepositional phrases or relative clauses which follow the common noun. In some cases, a prepositional phrase in this position might either be part of a definite description or modify a verb; but such an ambiguity cannot arise with relative clauses so a prepositional phrase can be made into a relative clause in order to test what it modifies. Relative clauses must therefore be part of the definite description when they are restrictive; on the other hand, non-restrictive clauses (the sort set off by commas) are analyzed using conjunction.

6.1.x. Exercise questions

1. Analyze each of the following sentences in as much detail as possible.

a. Ann introduced Bill to Carol.

b. Ann gave the book to either Bill or Carol.

c. Ann gave the book to Bill and he gave it to Carol.

d. Tom had the package sent to Sue, but it was returned to him.

e. Georgia will see Ed if she gets to Denver before Saturday.

f. If the murderer is either the butler or the nephew, then I'm Sherlock Holmes.

g. Neither Ann nor Bill saw Tom speak to either Mike or Nancy.

h. Tom will agree if each of Ann, Bill, and Carol asks him.

i. Reagan's vice president was the 41st president.

j. Tom found a fly in his soup and he called the waiter.

k. Tom found the book everyone had talked to him about and he bought a copy of it.

l. Wabash College is located in Crawfordsville, which is the seat of Montgomery County.

m. Sue and Tom set the date of their wedding but didn't decide on its location.

2. Synthesize idiomatic English sentences that express the propositions associated with the logical forms below by the intensional interpretations that follow them.

a. Wci ∧ Scl

S: [_ is south of _]; W: [_ is west of _]; c: Crawfordsville; i: Indianapolis; l: Lafayette

b. Mab → Mba

M: [_ has met _]; a: Ann; b: Bill

c. Iacb ∧ Iadb

I: [_ introduced _ to _]; a: Alice; b: Boris; c: Clarice; d: Doris

d. Wab ∧ Kabab

K: [_ asked _ to write _ about _]; W: [_ wrote to _]; a: Alice; b: Boris

e. g = c → (f = s ∧ p = t)

c: the city; f: football; g: Green Bay; p: the Packers; s: the sport; t: the team

f. (Sab ∧ ¬ Sa(fc)) → ¬ b = fc

S: [_ has spoken to _]; f: [_'s father]; a: Ann; b: Bill; c: Carol

g. (B(fa)(mb) ∨ S(ma)(fb)) → Cab

B: [_ is a brother of _]; C: [_ and _ are cross-cousins]; S: [_ is a sister of _]; f: [_'s father]; m: [_'s mother]; a: Ann; b: Bill

h. Pab(m(sb)(sc)) ∧ Pac(m(sb)(sc))

P: [_ persuaded _ to accept _]; m: [the best compromise between _ and _]; s: [_ 's proposal]; a: Ann; b: Bill; c: Carol

Glen Helman 01 Aug 2013

6.1.xa. Exercise answers

1. a. Ann introduced Bill to Carol

[_ introduced _ to _] Ann Bill Carol

Iabc

I fits a, b, ən c

I: [_ introduced _ to _]; a: Ann; b: Bill; c: Carol

b. Ann gave the book to either Bill or Carol

Ann gave the book to Bill ∨ Ann gave the book to Carol

[_ gave _ to _] Ann the book Bill ∨ [_ gave _ to _] Ann the book Carol

Gakb ∨ Gakc

either G fits a, k, ən b or G fits a, k, ən c

G: [_ gave _ to _]; a: Ann; b: Bill; c: Carol; k: the book

c. Ann gave the book to Bill and he gave it to Carol

Ann gave the book to Bill ∧ Bill gave the book to Carol

[_ gave _ to _] Ann the book Bill ∧ [_ gave _ to _] Bill the book Carol

Gakb ∧ Gbkc

both G fits a, k, ən b and G fits b, k, ən c

G: [_ gave _ to _]; a: Ann; b: Bill; c: Carol; k: the book

d. Tom had the package sent to Sue, but it was returned to him

Tom had the package sent to Sue ∧ the package was returned to Tom

[_ had _ sent to _] Tom the package Sue ∧ [_ was returned to _] the package Tom

Htps ∧ Rpt

both H fits t, p, ən s and R fits p ən t

H: [_ had _ sent to _]; R: [_ was returned to _]; p: the package; s: Sue; t: Tom

e. Georgia will see Ed if she gets to Denver before Saturday

Georgia will see Ed ← Georgia will get to Denver before Saturday

[_ will see _] Georgia Ed ← [_ will get to _ before _] Georgia Denver Saturday

Sge ← Ggds

Ggds → Sge

if G fits g, d, ən s then S fits g ən e

G: [_ will get to _ before _]; S: [_ will see _]; d: Denver; e: Ed; g: Georgia; s: Saturday

f. If the murderer is either the butler or the nephew, then I'm Sherlock Holmes

the murderer is either the butler or the nephew → I'm Sherlock Holmes

(the murderer is the butler ∨ the murderer is the nephew) → I = Sherlock Holmes

(the murderer = the butler ∨ the murderer = the nephew) → i = s

(m = b ∨ m = n) → i = s

if either m is b or m is n then i is s

b: the butler; i: I; m: the murderer; n: the nephew; s: Sherlock Holmes

g. Neither Ann nor Bill saw Tom speak to either Mike or Nancy

¬ (Ann saw Tom speak to either Mike or Nancy ∨ Bill saw Tom speak to either Mike or Nancy)

¬ ((Ann saw Tom speak to Mike ∨ Ann saw Tom speak to Nancy) ∨ (Bill saw Tom speak to Mike ∨ Bill saw Tom speak to Nancy))

¬ (([_ saw _ speak to _] Ann Tom Mike ∨ [_ saw _ speak to _] Ann Tom Nancy) ∨ ([_ saw _ speak to _] Bill Tom Mike ∨ [_ saw _ speak to _] Bill Tom Nancy))

¬ ((Satm ∨ Satn) ∨ (Sbtm ∨ Sbtn))

not either either S fits a, t, ən m or S fits a,t, ən n or either S fits b,t, ən m or S fits b,t, ən n

S: [_ saw _ speak to _]; a: Ann; b: Bill; m: Mike; n: Nancy; t: Tom

h. Tom will agree if each of Ann, Bill, and Carol asks him

Tom will agree ← each of Ann, Bill, and Carol will ask Tom

Tom will agree ← ((Ann will ask Tom ∧ Bill will ask Tom) ∧ Carol will ask Tom)

[_ will agree] Tom ← (([_ will ask _] Ann Tom ∧ [_ will ask _] Bill Tom) ∧ [_ will ask _] Carol Tom)

$$Gt \leftarrow ((Aat \land Abt) \land Act)$$

$$((Aat \land Abt) \land Act) \rightarrow Gt$$

if both both A fits a ən t and A fits b ən t and A fits c ən t then G fits t

A: [_ will ask _]; G: [_ will agree]; a: Ann; b: Bill; c: Carol; t: Tom

The function of each here is to indicate individual requests rather than a request made by Ann, Bill, and Carol as a group; the latter idea might be analyzed using a single four-place predicate [_, _, and _ will ask _] in place of the series of two-place predications used above.

i. Reagan's vice president was the 41st president.

Reagan's vice president = the 41st president

[_'s vice president] Reagan = [the _th president] 41

$$vr = pf$$

v of r is p of f

p: [the _th president]; v: [_'s vice president]; f: 41; r: Reagan

j. Tom found a fly in his soup and he called the waiter

Tom found a fly in his soup ∧ Tom called the waiter

Tom found a fly in Tom's soup ∧ Tom called the waiter

[_ found a fly in _] Tom Tom's soup ∧ [_ called _] Tom the waiter

Ft(Tom's soup) ∧ Ctr

Ft([_'s soup] Tom) ∧ Ctr

$$Ft(st) \land Ctr$$

both F fits t ən s of t and C fits t ən r

C: [_ called _]; F: [_ found a fly in _]; s: [_'s soup]; r: the waiter; t: Tom

k. Tom found the book everyone had talked to him about and he bought a copy of it

Tom found the book everyone had talked to him about ∧ Tom bought a copy of the book everyone had talked to him about

Tom found the book everyone had talked to Tom about ∧ Tom bought a copy of the book everyone had talked to Tom about

[_ found _] Tom the book everyone had talked to Tom about ∧ [_ bought a copy of _] Tom the book everyone had talked to Tom about

Ft(the book everyone had talked to Tom about) ∧ Bt(the book everyone had talked to Tom about)

Ft([the book everyone had talked to _ about] Tom) ∧ Bt([the book everyone had talked to _ about] Tom)

$$Ft(bt) \land Bt(bt)$$

both F fits t ən b of t and B fits t ən b of t

B: [_ bought a copy of _]; F: [_ found _]; b: [the book everyone had talked to _ about]; t: Tom

l. Wabash College is located in Crawfordsville, which is the seat of Montgomery County

Wabash College is located in Crawfordsville ∧ Crawfordsville is the seat of Montgomery County

[_ is located in _] Wabash College Crawfordsville ∧ Crawfordsville = the seat of Montgomery County

Lbc ∧ c = [the seat of _] Montgomery County

$$Lbc \land c = sm$$

both L fits b ən c and c is s of m

L: [_ is located in _]; s: [the seat of _]; b: Wabash; c: Crawfordsville; m: Montgomery County

m. Sue and Tom set the date of their wedding but didn't decide on its location

Sue and Tom set the date of their wedding ∧ Sue and Tom didn't decide on the location of their wedding

Sue and Tom set the date of Sue and Tom's wedding ∧ ¬ Sue and Tom decided on the location of Sue and Tom's wedding

[_ and _ set _] Sue Tom the date of Sue and Tom's wedding ∧ ¬ [_ and _ decided on _] Sue Tom the location of Sue and Tom's wedding

Sst(the date of Sue and Tom's wedding) ∧ ¬ Dst(the location of Sue and Tom's wedding)

Sst([the date of _] Sue and Tom's wedding) ∧ ¬ Dst([the location of _] Sue and Tom's wedding)

Sst(d(Sue and Tom's wedding)) ∧ ¬ Dst(l(Sue and Tom's wedding))

Sst(d([_ and _'s wedding] Sue Tom)) ∧ ¬ Dst(l([_ and _'s wedding] Sue Tom))

$$Sst(d(wst)) \land \neg Dst(l(wst))$$

both S fits s, t, ən d of (w of s ən t) and not D fits s, t, ən l of (w of s ən t)

D: [_ and _ decided on _]; S: [_ and _ set _]; d: [the date of _]; l: [the location of _]; w: [_ and _'s wedding]; s: Sue; t: Tom

2. a. [_ is west of _] Crawfordsville Indianapolis ∧ [_ is south of _] Crawfordsville Lafayette

Crawfordsville is west of Indianapolis ∧ Crawfordsville is south of Lafayette

Crawfordsville is west of Indianapolis and south of Lafayette

b. [_ has met _] Ann Bill → [_ has met _] Bill Ann

Ann has met Bill → Bill has met Ann

If Ann has met Bill then he has met her

c. [_ introduced _ to _] Alice Clarice Boris ∧ [_ introduced _ to _] Alice Doris Boris

Alice introduced Clarice to Boris ∧ Alice introduced Doris to Boris

Alice introduced Clarice and Doris to Boris

d. [_ wrote to _] Alice Boris ∧ [_ asked _ to write _ about _] Alice Boris Alice Boris

Alice wrote to Boris ∧ Alice asked Boris to write Alice about Boris

Alice wrote to Boris ∧ Alice asked Boris to write her about himself

Alice wrote to Boris and asked him to write her about himself

e. $g = c \rightarrow (f = s \land p = t)$

Green Bay = the city → (football = the sport ∧ the Packers = the team)

Green Bay is the city → (football is the sport ∧ the Packers are the team)

Green Bay is the city → football is the sport and the Packers are the team

If Green Bay is the city, then football is the sport and the Packers are the team

f. ([_ has spoken to _] Ann Bill ∧ ¬ [_ has spoken to _] Ann ([_'s father] Carol)) → ¬ Bill = [_'s father] Carol

(Ann has spoken to Bill ∧ ¬ [_ has spoken to _] Ann Carol's father) → ¬ Bill = Carol's father

(Ann has spoken to Bill ∧ ¬ Ann has spoken to Carol's father) → ¬ Bill is Carol's father

(Ann has spoken to Bill ∧ Ann hasn't spoken to Carol's father) → Bill isn't Carol's father

Ann has spoken to Bill but not to Carol's father → Bill isn't Carol's father

If Ann has spoken to Bill but not to Carol's father, then Bill isn't Carol's father

g. (B([_'s father] Ann)([_'s mother] Bill) ∨ S([_'s mother] Ann)([_'s father] Bill)) → [_ and _ are cross-cousins] Ann Bill

([_ is a brother of _] Ann's father Bill's mother ∨ [_ is a sister of _] Ann's mother Bill's father) → Ann and Bill are cross-cousins

(Ann's father is a brother of Bill's mother ∨ Ann's mother is a sister of Bill's father) → Ann and Bill are cross-cousins

Ann's father is a brother of Bill's mother or Ann's mother is a sister of Bill's father → Ann and Bill are cross-cousins

If Ann's father is a brother of Bill's mother or Ann's mother is a sister of Bill's father, then Ann and Bill are cross-cousins

h. Pab(m([_'s proposal] Bill)([_'s proposal] Carol))
∧ Pac(m([_'s proposal] Bill)([_'s proposal] Carol))
Pab([the best compromise between _ and _] Bill's proposal Carol's proposal)
∧ Pac([the best compromise between _ and _] Bill's proposal Carol's proposal)
[_ persuaded _ to accept _] Ann Bill the best compromise between Bill's proposal and Carol's proposal
∧ [_ persuaded _ to accept _] Ann Carol the best compromise between Bill's proposal and Carol's proposal
Ann persuaded Bill to accept the best compromise between his and Carol's proposals ∧ Ann persuaded Carol to accept the best compromise between Bill's proposal and hers
Ann persuaded each of Bill and Carol to accept the best compromise between their proposals

Glen Helman 01 Aug 2013