5

Transitive verbs: Resolving an apparent syntax/semantics mismatch

This chapter is somewhat didactic—for a reason. The immediate goal is to discover the semantic type of transitive verbs, which could be done in a few pages. But we will instead go down a lengthy and winding path (which will turn out to be fruitless). This involves setting up a somewhat naive "Straw Man," whose hypothesis requires giving up Direct Compositionality and then leads to greater and greater complications. After Straw Man has taken enough twists and turns, we abandon that path, and turn to a simple Direct Compositional solution. In one sense the lengths to which Straw Man goes is a bit silly, for in reality most work in modern semantics does not take the Straw Man position. The ultimate solution arrived at in this chapter is fairly standard (even in non-Direct Compositional theories).[1] But this is precisely what makes this domain an important one. For here the Direct Compositional solution turns out to be

[1] An exception to this is work done under the rubric of *Neo-Davidsonian event semantics*; see, e.g., Parsons (1990).

reasonably obvious with some fairly simple tools that allow the semantics to be set up in such a way as to respect the syntax. Perhaps, then, this basic strategy extends to other domains; perhaps slightly more subtle tools can be pressed into service in other cases that appear to challenge Direct Compositionality.

5.1. First pass: Straw Man's solution

Consider sentences with transitive verbs like those in (1):

(1) a. Juliet loves Romeo.
b. The pig chased the wolf.
c. Lee kissed Sandy.

(Once again t eventive verbs like *chase* and *kiss* are put into the past tense to avoid the habitual reading that emerges with present-tense morphology.) What kind of meaning should be assigned to *kiss*, which is often thought of as a relation between individuals? Straw Man consults an elementary logic book, in which sentences like (1c) are translated into a formula of first-order predicate logic and are represented as in (2):[2]

(2) K(l,s)

Recall that an intransitive verb has as its value a (world/time-dependent) function of type <e,t>. If asked how to construe (2) as a way to name a model-theoretic object, a reasonable answer is to take *l* and *s* in the formula above to denote individuals, and to take [[K]] to be a function whose domain is the set of all ordered pairs of individuals and whose co-domain is the set of truth values. In other words, (l,s) picks out an ordered pair of two individuals, and [[K]] is a function mapping such a pair to true or false. In set terms, [[K]] characterizes a *set of ordered pairs of individuals*.

Straw Man applies this to the semantic composition of English, and concludes that English [[kiss]] is like [[K]]: a function from the set of ordered pairs of individuals to t. (Recall that the set of all ordered pairs whose first member is in set A and second is in set B is represented as A x B.) Thus the

[2] While we will show that this is not a good way to represent the semantic composition of English, the point here is *not* to criticize the use of such formulas in a logic textbook: most work using formulae like this does not claim to be modeling the grammar of natural language. The goal of the logician is quite different (to provide a way to represent valid reasoning systems).

type of meaning for a transitive verb would be <e x e,t> (or, more accurately, <s,<e x e,t>>). Incidentally, one need not have taken a logic class to come up with the hypothesis that transitive verbs take ordered pairs as their arguments. We often speak of *kiss* as denoting a (two-place) relation between two individuals. But recall that the formal notion of a (two-place) relation is a set of ordered pairs (and of course any set can be recast as the characteristic function of that set, which gives the type above).[3] It is also common parlance to say that the verb applies to a pair of a "kisser" and "kissee," or to an "agent" and a "theme," and this also leads naturally to the hypothesis that the type for transitive verbs is as above. There is, in fact, nothing wrong with this informal terminology; the central question of this chapter is whether the type above is the right way to formally model the informal intuition.

Let us call the syntactic category of a transitive verb V_2. Then the semantic type assumed above leads us to posit FR-1. (Fictional rules that will quickly be discarded will be notated as FR.) To minimize notational collision, an ordered pair whose first member is a and second is b will (non-standardly) be notated as **{**a,b**}** (with boldface curly brackets to distinguish from ordinary curly brackets which are used for sets). This notation is not ideal, but unfortunately all of the other types of bracket are used elsewhere in our formalism (the standard representation of an ordered pair is (a,b), but parentheses are also used to indicate the argument of a function). FR-1 is given extensionally here; the interested reader can supply the intensional version:

FR-1 If there is an expression α of the form <[α], NP, [[α]]>, an expression β of the form <[β], NP, [[β]]> and an expression γ of the form <[γ], V_T, [[γ]]> then there is an expression δ of the form <[α-γ-β], S, [[γ]](**{**[[α]], [[β]]**}**)>.

Since the meaning of the transitive verb combines with the meanings of its subject and object simultaneously in the semantics (i.e., by applying to an ordered pair) the syntax also directly combines the two NPs with the transitive verb.

But wait, says the syntactician. We learn from our introductory syntax texts that this cannot be the right syntax. The phrase structure rule embodied in FR-1 is S $\rightarrow$ NP V_2 NP, but we know (or at least we think) that *loves Juliet* in (1a) is a constituent and similarly for *kissed Sandy* in (1c). Moreover, we know (or at least we think) that these expressions are of exactly

[3] A two-place relation means a set of ordered pairs. There are also three-place relations which are sets of ordered triples, and more generally one can speak of *ordered n-tuples*.

the same category as expressions consisting of only single intransitive verbs—i.e., they are VPs. It is actually worth occasionally reminding oneself of the arguments that lead to this conclusion; we mention just one here. Expressions like *kissed Sandy, watered the plants*, or *fed the pig* can conjoin with each other, and can also conjoin with simple intransitive verbs:[4]

(3) a. Lee watered the plants and fed the pig.
b. Lee watered the plants and meditated.
c. Lee meditated and fed the pig.

If transitive verbs were introduced in the syntax in the flat way given in FR-1, we would need three additional syntactic rules for these:

(4) a. S $\rightarrow$ NP V_T NP and V_T NP
b. S $\rightarrow$ NP V_T NP and V_I
c. S $\rightarrow$ NP V_I and V_T NP

But this would still not be enough for these conjunctions can be iterated:

(5) a. Lee fed the pig and chased the wolf and meditated.
b. Lee fed the pig and meditated and watered the plants and studied.

Additional rules would thus be needed, but since one can iterate these indefinitely, no amount of new rules will be enough. Positing a category VP solves all of this: simply add a syntactic rule of the form in (6), and assume the two VP rules in (7):

(6) VP $\rightarrow$ VP and VP

(7) a. VP $\rightarrow$ V_I
b. VP $\rightarrow$ V_T NP

Note that the remarks above concerning *and* extend also to *or* as in (8), and so we also add the rule in (9):

(8) a. Lee milked the cow or fed the pig.
b. Lee milked the cow or meditated.

(9) VP $\rightarrow$ VP or VP

[4] There are actually many versions of generative grammar—including earliest Transformational Grammar—in which it is assumed that all verbs must have a subject (at least at some level) in the syntax. This would invalidate the argument below. For *fed the pig* in (3a) for example would at some level have a deleted or silent subject and be a full sentence; and all conjunction would actually involve S conjunction. We will not consider these here, but see the discussion in section 5.3.

Ultimately, there are many refinements one could make to (6) and (9), but these formulations are fine for now as we turn back to the semantics.

All is well as far as the syntax is concerned, but since the goal is a theory of syntax *and* semantics it is striking to notice that the rules above give only the syntactic side. What would be the semantics that goes along with (7b)? Well, if we continue with the Straw Man hypothesis (i.e., [[kiss]] is a function that takes as its argument some ordered pair) there can be none. The reason is simple: V_2 is not of the right type to combine with just the one individual denoted by the object NP. And since there is no semantics for (7b), it also does not make sense to ask about the semantics associated with (6) and (9); there is none.

5.2. Abandoning Direct Compositionality

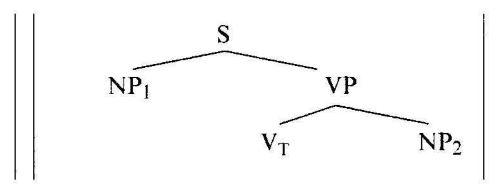
Straw Man is stubborn and does not wish to give up on the idea that [[kiss]] is a function whose domain is e x e. But he has learned his syntax well and is happy enough with the syntactic rules above. Thus he decides to abandon the hypothesis of Direct Compositionality. He will leave the syntax of (7) alone, and adopt a different model of grammar from the one we have been assuming so far. Before developing this, a word about the role of trees. Probably a bit surprisingly for a linguist, there has so far not been a single tree drawn in this book. This is because we have simply stated the rules, and the rules so far make no reference to trees. Consider a complex sentence like, for example, *Mitka howled and Porky grunted.* (Porky made his debut in semantics in Lewis 1970.) One can of course draw a tree for this, but the tree is just a representation of the steps in a proof of the well-formedness of the sentence. Suppose that the rules combining expressions do nothing more than put such expressions next to each other: this is known as *concatenation* (this assumption will be subject to greater scrutiny in 5.5). Put differently, the syntactic portion of the rules are just those that could be written as a context-free phrase structure grammar (see Chapter 3, n. 4).[5] Under this

[5] It is well known that ultimately a theory of grammar will need to include rules beyond just context-free phrase structure rules (which only concatenate expressions). While the jury may remain out for the case of English, considerable debate about this point during the 1980s (especially within the theory of Generalized Phrase Structure Grammar) showed that additional devices beyond context-free phrase structure grammars will be needed for other languages. But just how much "extra" is needed remains open. See section 5.5 for discussion of these points.

view, the grammar never needs to refer to trees as it never refers to the internal structure of some expression. A tree is always just a way to represent the proof of the well-formedness of some expression, and given the Direct Compositional hypothesis, it is also a representation of the steps in the semantic composition. Under Direct Compositionality, then, combined with a rather impoverished view of the set of syntactic operations, a tree is a convenient representation for the linguist but it is not something that the grammar itself "sees."

But of course much work in syntax and semantics assumes that the grammar does have access to the trees, and the view of semantics to which we are about to turn also makes this assumption. Thus in order for Straw Man to both adopt (7b) in the syntax and at the same time maintain that [[kiss]] is a function of type <exe,t> (or the intensional counterpart), the syntax and semantics must be pulled apart. The syntax not only proves strings well-formed but keeps track of the steps used so that in the end it is a proof of the well-formedness *of a tree*. Moreover, it "feeds" into the semantics as a separate system: a system which provides interpretations for trees. Hence we adopt the syntactic rules in (7), and add a semantic rule interpreting a tree as in FR-2. (The notation here is inspired by the notation developed in Heim and Kratzer 1998, although they do not use it for this case as they adopt the same solution for transitive verbs as is developed later in this chapter.)

FR-2



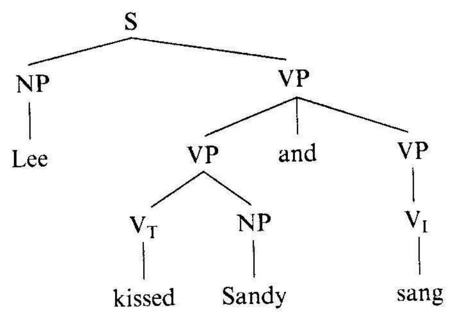
$= [[V_T]] (\{ [[NP_1]], [[NP_2]] \})$

(This is given extensionally; it is easy enough to fold in intensionality.) If this move is made, the entire fragment would be revised; each syntactic/semantic rule pair developed up to this point will be stripped of its semantic part. However, in all of the cases we have discussed so far the necessary revisions are straightforward; the revisions are left to the reader in the following exercise.

5.1. Take the rules in the fragment so far (this means only those rules labeled TR). For each one, assume that they give just the syntactic and phonological part (they could be stated as phrase structure rules or they could be stated in the notation used here). Assume further that for each such rule there is a separate semantic rule whose input is a tree. Then state each of these semantic rules, using FR-2 as a model for the statement of the semantic rules. You should note that FR-2 differs from the other rules you will be constructing in that all of those rules have as their input a *local tree*—that is, only a mother node and one or more daughter nodes—while FR-2 (by necessity) refers to a bigger chunk of tree—mother, daughter, and granddaughters.

The Direct Compositional strategy vs "syntax feeds semantics" strategy is discussed in Chapter 7, but for now we can note that the syntax feeds semantics strategy for this particular case comes at a cost. In the case of a sentence like *Lee kissed Sandy*, the fact that *kissed Sandy* has no meaning was remedied by the adoption of FR-2. But then, how does the semantics work for the case of conjoined VPs (i.e., cases where VPs are connected by *and* or *or*)? Recall that Straw Man posits the syntactic conjunction rule in (7). Now consider the syntactic structure in (10):

(10)

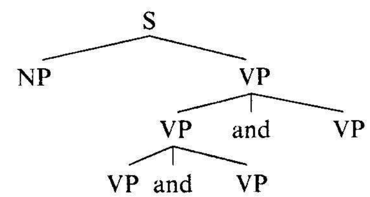


Do the rules in Straw Man's grammar provide an interpretation for this sentence? The answer is no: there is no rule to interpret either the topmost VP here nor the entire S. FR-2 is not relevant; it interprets only Ss of the form [$_S$ NP [$_{VP}$ V_T NP]], and none of the rules that you will have formulated in your answer to Exercise 5.1 will help either. So there would need to be an additional rule to interpret this tree.

5.2. Formulate the relevant rule (remembering to keep in mind the types that Straw Man is assuming).

But of course even adding the rule asked for in Exercise 5.2 does not complete the task. For the additional rules for (10) will do nothing for cases like those in (5), which have (as one possibility) the shape in (11):

(11)

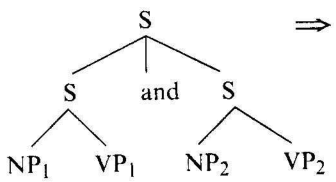


Due to the recursive nature of the *and*-rule, Straw Man could never finish stating the semantic rules. This is completely parallel to the reasoning as to why the syntax needs a VP constituent: the same reasoning leads to the conclusion that we need to find a meaning for each VP, which will be done in section 5.4.

5.3. Hidden S conjunction?

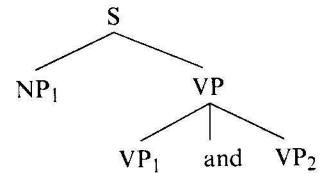
But first, Straw Man makes one final attempt to save his hypothesis. And this is to posit that the semantics actually does not interpret structures with VP conjunction. Rather, at the level at which these are interpreted, these are actually the conjunction of two sentences. This is actually not a straw man; this view has been taken in one form or another in many theories of grammar, and has been implemented in various ways. For example, in early Transformational Grammar it was assumed that there were no phrase structure rules like that in (6) and (9). Sentences like (3) were actually at an underlying level (deep structure) instances of S conjunction, with a transformational rule called Conjunction Reduction which mapped trees of the form in (12) to those in (13), provided that NP_1 was identical to NP_2 (see, e.g., Jacobs and Rosenbaum 1968):

(12)



⇒

(13)



(The rule can be generalized to extend to the case of *or*.) Additionally, it was assumed that the input to the semantic interpretation was the deep structure, and so the semantics would be assigning a meaning to structures of the sort shown in (12) and not to (13). Since we already have semantics for S conjunction, there would be no need to worry about interpretation for the case of VP conjunction. Hence for a sentence like *Romeo kissed Lee and watered the plants* we don't need to worry about giving a meaning for *kissed Lee.* The semantics would separately interpret the two sentences *Romeo kissed Lee* and *Romeo watered the plants* (along with the rule interpreting the conjunction of the two). FR-2 will do the job. (One might notice that under this view of conjoined sentences, the argument given earlier for a VP constituent has been undermined, but we will not pursue the consequences of this.) A variant on this strategy is to view the picture in reverse. That is, assume that the phrase structure rules do give something like (13) directly, but this is mapped to (12) (as its "Logical Form"). The semantics operates so as to compositionally interpret that level. Again FR-2 (combined with the other rules suitably reformulated in tree terms, as in Exercise 5.1) will now be adequate.

But the hidden S conjunction strategy also comes at a cost. First and foremost the syntax has again been complicated, no matter which variant is chosen. Either variant posits an additional level of representation and hence needs a set of explicit rules to map one level to the other. There is also an empirical difficulty centering on sentences whose subjects contain quantifiers. Consider the pairs in (14) and (15). In each pair the first involves conjunction or disjunction of just bare VPs, and the second is the corresponding conjoined Ss with (syntactically) identical subjects:

(14) a. Some pig chased the wolf and grunted.
b. Some pig chased the wolf and some pig grunted.

(15) a. Every pig chased the wolf or grunted.
b. Every pig chased the wolf or every pig grunted.

The meaning of (14b) is weaker than (14a); in the former the chaser and grunter must be the same. In (14b) the chaser and grunter could happen to

be the same, but they need not be.[6] Hence (14a) entails (14b) but not vice versa. The reverse situation holds in the *every* case. Here (15a) is weaker—the most salient reading for (15a) is each pig did one or the other of these things, but they need not have all been in agreement as to which act to perform. (15b) has the stronger requirement that they all did the same thing. Incidentally, it is true that (15a) also (at least marginally) has a second reading which is the same as (15b); this is best brought out by the continuation in (16):

(16) Every pig chased the wolf or grunted—but I can't remember which.

Nonetheless, the most salient reading of (15a) is clearly the weaker one and is true in situations in which (15b) is not. (The second reading is accounted for in Chapter 11; its existence does not affect the point at hand here.)

Consider a theory in which the Logical Form in (12) is related to the surface form in (13) and where the requirement for this is that the two NP subjects in the two clauses in (12) must be identical. Then, of course, we get the wrong prediction. In the cases above, the (a) sentences should have the same meaning as the (b) sentences. As noted above, there is a secondary reading of (15a) which is synonymous with (15b) but this is of little consolation since the correct theory needs to also pair (15a) with its primary meaning. To be sure, there are ways to solve these problems with more complex Logical Forms and additional conditions on the rules effecting the mapping to (or from) Logical Forms.[7] But the methods involved require yet further rules. Perhaps it is time to abandon Straw Man's path.

[6] It might appear at first glance that (14b) requires the wolf-chasing pig and the grunting pig to be different pigs. But closer inspection reveals that this is just an *implicature* (see section 3.2) and not part of the truth conditions. If a speaker knew that a single pig did both, they would more likely have used (14a)—hence the conclusion on hearing (14b) that different pigs are involved. But this can't really be part of the truth conditions. For if I hear a grunting pig and have no idea who it is and at the same time have some other indirect evidence about there being a wolf-chasing pig, I can certainly say (14b) with no commitment one way or the other as to whether it is the same pig. (And if it turns out to be the same pig, I have not said anything false.)

[7] Readers familiar with the "Quantifier Raising" solution to quantified NPs (see section 14.2) might think that one automatically gets the right meanings of the (b) sentences once one combines Quantifier Raising with the rule that produces the two-sentence LF. But it is not automatic, for a principle is needed to ensure that the two processes happen in the right order. We leave it to the interested reader to verify this.

5.4. Back to the drawing board: Curry'ed functions

5.4.1. Recasting the meaning of transitive verbs

The complications above are all the result of Straw Man's insistence that the meaning of a transitive verb is a function from ordered pairs to truth values. The problem is that this makes semantic composition look very different from the syntax, for the semantics treats the composition as "flat" while the syntax does not. But there is a simple way to recast things so that the semantics respects the syntax. The technique to be used here relies on the observation that any function whose domain is a set of ordered n-tuples (in this case, ordered pairs) can be mapped to a function that takes the arguments one at a time. The observation that this can be done for any such function is due originally to Schönfinkel (1924) and later H. B. Curry (see, e.g., Curry and Feys 1958). Although Schönfinkel's work is earlier, this is generally known as *Currying* a function (although see Heim and Kratzer 1998 who do use the historically more accurate term *Schönfinkelization*).

In the case at hand, we begin with the original [[K]] (a function from e x e to t) and recast this as a function which takes one individual and returns a function of type <e,t>. This can be done without loss of information—this is just repackaging the meaning. Thus [[kiss]] is a function of type <e,<e,t>>. Note that there are actually two different ways that one can recast the initial logician's [[K]] (which maps an ordered pair {x,y} to true just in case in the world in question x does kiss y). One way is to create a new function such that for any x and y such that [[K]]({x,y}) is true, [[kissed]](x)(y) is true. In other words, the "kisser" is the first argument of the new function, and the "kissee" is the second. The other mapping does just the reverse: for any x and y such that [[K]]({x,y}) is true, then [[kissed]](y)(x) is true.

It turns out that the second strategy is the correct one, given the assumption that meanings are packaged in such a way as to reflect the syntactic composition. Our new function combines with the meaning of the *syntactic direct object* first, because this is what happens in the syntax. It combines next with the subject. This is often confusing when first encountered because of the left-to-right order in which we write things: in a sentence like *Lee kissed Sandy*, the semantic composition will be [[kissed]] ([[Sandy]]) ([[Lee]]). (The other meaning can be thought of as the meaning of *was-kissed-by*).

To illustrate in greater detail, suppose that the universe consists of three individuals, a, b, and c. We will, moreover, be considering just one world w_1. To use English to describe the facts of the world, let it be the case that a kisses b and c, b kisses c, and c kisses a and c. Then the function [[kiss]] in w_1 is as follows:

(17)

$$a \rightarrow \left\{ \begin{array}{l} a \rightarrow 0 \\ b \rightarrow 0 \\ c \rightarrow 1 \end{array} \right\}$$

$$b \rightarrow \left\{ \begin{array}{l} a \rightarrow 1 \\ b \rightarrow 0 \\ c \rightarrow 0 \end{array} \right\}$$

$$c \rightarrow \left\{ \begin{array}{l} a \rightarrow 1 \\ b \rightarrow 1 \\ c \rightarrow 1 \end{array} \right\}$$

In order to get any mileage out of this, we also need to make explicit what was implicit in the discussion above—i.e., the syntactic and semantic combinatory rule for combining transitive verbs with their objects:

TR-6. If α is an expression of the form <[α], V_T, [[α]]> and β is an expression of the form <[β], NP, [[β]]> then there is an expression γ of the form <[α-β], VP, [[α]]([[β]])>.

Hence consider the composition of *Lee kissed Sandy*. Assume that [[Lee]] (in all worlds) is a and [[Sandy]] is b. Then *kissed Sandy* has a meaning; it is the function shown in (17) applied to the argument b, so it is that function of type <e,t> which maps a to 1 and b and c to 0. In set terms, *kissed Sandy* characterizes the set of Sandy-kissers (which in this case is the singleton set {a}). Then this VP combines with its subject by the syntactic and semantic rule TR-5′, and the resulting semantics is 1, as we would expect.

5.3. The picture above in (17) is the extension of [[kiss]] in a world that we have labeled w_1. Now take a different world w_2 in which the facts are that a kisses a, b, and c, b kisses no one, and c kisses only c. (Ignore the obvious lack of pragmatic plausibility here!)

(cont.)

First, show the extension of [[kiss]] at w_2 where this again will be shown as a Curry'ed function of the same basic structure as (17). Then show the full syntactic and semantic composition of *Sandy kissed Lee*, where at each step you will be showing the meaning of each expression as a function from worlds to some other object. (You already have [[kiss]]; it is a function mapping w_1 to the object shown above in (17) and mapping w_2 to the new object that you will have constructed.) The two NPs are listed in the lexicon and each is a constant function from worlds to individuals as given above. So your task is to show how the syntax and the semantics put this all together step by step. Do this using (12) in Chapter 4 as the model.

Does this strategy introduce undue complexity into the modeling of meanings? Not at all. We have simply taken [[kiss]] to be a function whose *co-domain* itself is a function. The general approach of the kind of model-theoretic semantics being assumed here is to have a small set of primitive model-theoretic objects (worlds, times, truth values, and individuals) and take other kinds of meanings to be constructed from these using the notion of functions.[8] There is no reason to be surprised to find that certain expressions denote functions which themselves have functions as their co-domain (or, for that matter, as their domain; see Chapter 10). In fact, such objects are already in the system. In the fully intensional semantics, the meaning of an intransitive verb, for example, is a function of type <s,<e,t>>.

Incidentally, one finds different notations for the semantic composition of a sentence like *Lee kissed Sandy*, even among researchers who agree that [[kiss]] is the Curry'ed function of type <e,<e,t>> (combining first with the individual denoted by the object). Given the world above, the composition is most accurately represented as [[kiss]](b)(a). However, one sometimes finds more reader-friendly notation when the actual step-by-step composition doesn't matter, and so one might see something like [[kiss]](a,b), which looks like it is going back to the hypothesis that [[kiss]] takes

[8] There are debates as to whether others are needed—see especially the work in event semantics (e.g., Parsons 1990) which takes events as primitives. But these will suffice for what we will be considering in this book.

an ordered pair as argument. But generally this is intended simply as a way to make things easier to read; this is called the relational notation. It's ultimately helpful to get used to going back and forth between the two notations.

*5.4.2. *Currying more generally*

The above illustrates the technique of Currying to a function that has as its domain a set of ordered pairs, but this is quite general. For any function f which takes an ordered n-tuple as its argument to give some result x, there is a corresponding function f′ which takes a single argument, and returns as value a function g from an n-1 tuple to x. This can apply recursively, until we have a function that takes its arguments one at a time.

Generalizing the remarks above, concerning the fact that there are different ways to Curry a function like [[K]], there are many ways to effect the mapping from f to the one-place function f′, depending on which position of the n-tuple is peeled off (A one-place function is one which takes just a single argument, not an ordered tuple). Call f'^{-i} the function that peels off the ith position of the n-tuple. Then there is a unique way to map f to f'^{-i} (and vice versa). Notice that the relationship between the naive meaning of *kiss* (a function taking ordered pairs of kissers and kissees) and our hypothesized actual meaning of English *kiss* is such that [[kiss]] is [[K^{-2}]] since we took the second member of the ordered pair to be the first argument of [[kiss]]. This is crucial: this was done to model the way in which the syntax works, as the syntax first combines *kiss* with its object and then with its subject (and, simply because it is traditional, we took "naive-kiss" to map a pair {a,b} to true just in case a kisses b).

***5.4.** Take a set A with two members {a,b}, a set B with two members {c,d}, a set C with two members {e,f}, and a set D with three members {1,2,3}. Consider a function f whose domain is the set of ordered triples in A x B x C and whose co-domain is D. (In other words, it takes each triple that can be composed by taking a member of A, a member of B, and a member of C and maps each such triple into some member of D.) The actual function is as follows:

(cont.)

{a,c,e} → 1
{a,c,f} → 1
{a,d,e} → 3
{a,d,f} → 2
{b,c,e} → 1
{b,c,f} → 3
{b,d,e} → 2
{b,d,f} → 3

Convert this to the corresponding Curry'ed function of type <C,<B,<A,D>>>.

5.4.3. Solving Straw Man's problems

Let us consider whether this strategy solves the problems encountered by Straw Man. First, as seen above, a VP such as *kissed Sandy* now has a meaning. And, having Curry'ed the transitive verb, this VP has a meaning of exactly the same type as the meaning of a VP with only an intransitive verb. Both are world/time-dependent functions of type <e,t> (put differently, they characterize sets of individuals). Because of this, we also now have a simple way to give the semantics for conjoined VPs without having to assume that these really involve hidden **S** conjunction. For expository simplicity, we ignore the possibility of allowing the denotation of a VP to be a partial function from the domain of individuals, and so assume that each VP maps each individual into either true or false.[9] Consider the phrase structure rule in (6) which introduces VPs conjoined with *and*. (Keep in mind that this will eventually be generalized to allow coordination of all categories.) Up until now we have in our fragment only a semantics for *and* that conjoins sentences (TR-1), but the discussion in 5.1 led to the conclusion that there is VP conjunction as well. Direct Compositionality requires a semantics to go with (6), and we are now in a position to see what this semantics is. Thinking in set terms, the contribution of *and* is simply intersection. (Ultimately one would hope for single *and* for sentences, VPs, and other expressions; this is accomplished in Chapter 12.) In other words, consider a sentence like a slight variant of (3c) *Lee meditated and fed Porky*. The intuition is easiest to get in set (rather than function) terms. Since [[feed]] is of type <e,<e,t>>, it combines with [[Porky]] (some individual) to give the characteristic function of the set of individuals who fed Porky. [[meditated]] also characterizes some set. These two sets can intersect; and the result is the set of individuals in both the pig-feeding set and the meditating set. This is exactly right; when this combines with Lee, the sentence says that Lee is in this set—i.e., the set which is the intersection of the meditators and Porky-feeders. By the definition of intersection, it follows that Lee is in each set. Notice, then, that with ordinary individual-denoting subjects (like *Lee*) the truth conditions for the case of conjoined VPs are the same as the truth conditions for the case of conjoined sentences, as in *Lee fed Porky and Lee meditated.* But this follows from the semantics that intersects the two VP meanings, and so the correct truth conditions are arrived at without having to pretend that there are really two sentences.

To formalize the rule for conjoined VPs, note first that since we are not considering the case of partial functions here we can safely go back and forth from talking about sets of individuals to the corresponding characteristic function of this set and vice versa. Given a function f of type <x,t> (for any set x) we have introduced the notation f_S to indicate that subset of x mapped to 1 by f. Now take any set z which is a subset of some set x. We now introduce the notation z_F to indicate the function of type <x,t> that characterizes z (i.e., the function mapping all members of z to 1 and all other members of x to 0). Using these notational devices, we can state the VP conjunction rule (extensionally):

TR-7. If α is an expression of the form <[α], VP, [[α]]> and β is an expression of the form <[β], VP, [[β]]>, then there is an expression γ of the form <[α-and-β], VP, $([[\alpha]]_S \cap [[\beta]]_S)_F$>.

Thus the meaning of the conjoined VP is the function characterizing the intersection of the two sets characterized by the daughter VPs. The rule introducing *or* is similar, where here the relevant semantics is set union:

TR-8. If α is an expression of the form <[α], VP, [[α]]> and β is an expression of the form <[β], VP, [[β]]>, then there is an expression γ of the form <[α-or-β], VP, $([[\alpha]]_S \cup [[\beta]]_S)_F$>.

[9] As noted several times, a possible way to model presupposition failure is to use partial functions. To use an example we have seen before, we might ultimately want to treat the presupposition in a predicate *stopped smoking* by saying that it is undefined (at a world/time pair) for all individuals who never smoked. If this is correct, the rule for conjunction given below will need refinement.

5.5. Take a universe with three individuals {a,b,c}. Take a world w_1 such that [[kiss]](w_1) is as shown in (17) above. Assume further the following: [[Juliet]](w_1) = a and [[Romeo]](w_1) = c. Moreover [[sleepwalk]](w_1) is the following function:

a → 0, b → 1, c → 1.

Take the sentences in (i) and (ii):

(i) Juliet sleepwalked and kissed Romeo.
(ii) Romeo kissed Juliet or sleepwalked.

Show the full syntactic and semantic composition of this sentence extensionally with respect to w_1. In other words, show the value of each expression at w_1 which the syntax proves well-formed. (Use the model in (12) in Chapter 4 for how to display this.) For all of the VPs, show these both as sets and as the functions which characterize those sets. What is crucial is to see why the sentence doesn't need to be converted into two separate sentences in order to arrive at the correct semantics for the whole; showing out the extension of each of the well-formed VPs in set terms helps to elucidate that point.

This account will eventually be refined in several ways. First, *and* and *or* will ultimately be listed in the lexicon (so they themselves have a meaning) and will be members of a single syntactic category. Second, there appears to be evidence that both *and* and *or* actually take their arguments one at a time such that the structure of, for example, *walks and chews gum* is actually [walks[and[chews gum]]] (see, e.g., Munn 1993). Note that this means that [[and]] is (thinking in set terms) the Curry'ed version of the intersection operator. (The interested reader can use prose to state just exactly what this function is.) Both of these desiderata are accomplished in the revision in the next chapter. And finally, as noted above, conjunctions like *and* and *or* are cross-categorial: for any two expressions [α] and [β] of the same category X, there is also an expression of the form [α-and-β]. The complex expression itself is of category X; we know this because it has the same distribution as any other expression of category X. Ultimately then a single *and* and a single *or* should be listed in the lexicon which serves to conjoin expressions of any category. A full generalization is given in Chapter 12 (and section 13.4.4.2

contains yet another way to accomplish the generalization). But for the moment we will not collapse S conjunction (TR-1) with VP conjunction (TR-7). Note, though, that while the semantics for VP conjunction and for S conjunction look at first blush to be completely different, they have something in common. Recall that the value of an S is a function characterizing a set of worlds; the fully intensional version of the *and*-rule in the case of S conjunction also intersects these sets.

5.6. In light of the remarks above, it is tempting to try to collapse the two conjunction rules as follows:

FR-3 If there is an expression α of the form <[α], X, [[α]]> and an expression β of the form <[β], X, [[β]]> then there is an expression γ of the form <[α-and-β], X, $([[\alpha]]_S \cap [[\beta]]_S)_F$>, for X ranging over S and NP>.

Aside from the fact that ultimately we want to allow in conjunction of other categories besides just VP and S, there is a "cheat" embodied in FR-3. Why is it cheating to try to collapse the case of VP conjunction and S conjunction in the way done in FR-3?

Straw Man's final problem was that his conjecture that VP conjunction was secretly S conjunction gave (without additional complexities) the wrong meanings for sentences like (14a) and (15a) where the subjects are quantified:

(14) a. Some pig chased the wolf and grunted.

(15) a. Every pig chased the wolf or grunted.

Under the two-sentence analysis of these, these should have the same meaning as the corresponding sentences where the subject is repeated, and yet they do not. But we cannot truly boast that the Curry'ed transitive verb solution solves all of Straw Man's problems until it is shown that this problem is avoided here. In fact, we are not yet in a position to do that: but this is only because the meaning of NPs with quantifiers has not yet been discussed. We postpone this until Chapter 10. It turns out that armed with a meaning for expressions like *every man*, etc., the meaning (or at least preferred meaning) of (14a) and of (15a) is exactly as predicted. (It will also be shown that the secondary meanings can be derived.) At this point we can only say "stay tuned."

*5.5. A note on three-place verbs

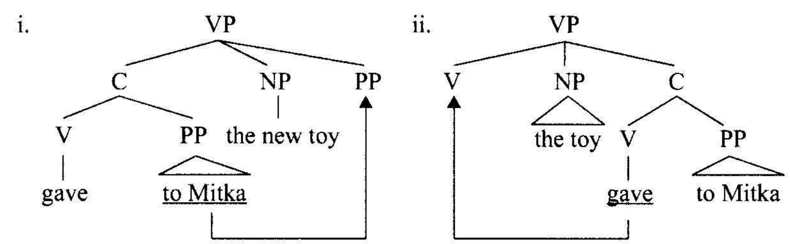
English has many verbs that occur with two constituents within the verb phrase, as in *gave the new toy to Mitka*, which consist of the sequence V-NP-PP (we will call the verb V_3 here). There are also cases of verbs which take sequences of two NPs such as another version of *give*: *gave Mitka the new toy*. (The relationship between the two versions of *give* is much discussed in the generative grammar literature; here we focus only on the version that takes an NP and PP, but see also Exercise 9.10.) And some verbs take NPs and sentential-like expressions (which we will call CP) as in *told Chicken Little that the sky is falling* and so forth.

Here we focus on the V-NP-PP verbs such as *give*. Before going further we need to say something about the meaning of *to Mitka*. Assume that *to* functions much like a case marker and contributes nothing to the meaning, so that *to Mitka* means the same as *Mitka*. Returning to the full VP *gave the new toy to Mitka*, there are four logical possibilities for how this is put together in the syntax (indeed, all four possibilities have been suggested):

1. The VP is "flat": in phrase structure rule terms there is a rule of the form: VP → V_3 NP PP (see, e.g., Gazdar, Klein, Pullum, and Sag 1985).
2. The NP and PP actually form some sort of constituent; call it Z and there are two rules:
   VP → V_3 Z and Z → NP PP (see Kayne 1984).
3. *give* and *the new toy* form a constituent which then combines with a PP (as does the simple verb *dash*). The new rules would thus be something like:
   VP → V_4 PP and V_4 → V_3 NP.
   This means that *give* and *the new toy* combine to give a complex verb of the same category as things like *dash*; *gave the new toy* then combines with the PP *to Mitka* and the result is a VP (see, e.g., Dowty 1988).
4. The last possibility is one which requires additional apparatus in the syntax, as the syntax involves more than just concatenating expressions. This would be to assume that *give* actually first combines with *to Mitka* and the resulting expression then combines with *the new toy*. Versions of this (either for this case or for related cases) are proposed in Chomsky (1957) (for a VP like *consider George a fool*), and explored in much subsequent Categorial Grammar literature (Bach 1979, 1980, Dowty 1982, and others), within GPSG by Jacobson (1987), and then again in Government and Binding theory in Larson (1988).

The basic idea of the proposals in 4 above is that *gave to Mitka* in the expression *give the toy to Mitka* is a *discontinuous constituent*. There are two broad classes of ways to implement this intuition. One makes use of two levels of representation in the grammar, and posits a movement rule to break up the constituent. This in turn has two possible implementations. (i) The basic structure is as in (18i) and *to Mitka* moves to the right as shown there, or (ii) the basic structure is as in (18ii) and *give* preposes (we suppress details of the actual proposals, and we arbitrarily adopt the label C for the expression that is the discontinuous constituent).

(18)



Chomsky (1957) proposed a version of (i) (although for a different set of cases), and Larson (1988) proposes a version of (ii).

A rather different way to implement the basic intuition is put forth in Bach (1979, 1980). This relies on the observation that the existence of discontinuous constituents does not inevitably lead to the conclusion that there is more than one level of representation. Rather, the syntactic rules combining two expressions might perform operations other than just putting one (or more) expressions next to each other. Perhaps there are also cases where one expression is infixed into another.[10] (Note that

[10] A common misconception is that the existence of "discontinuous constituents" requires more than one level of representation because a sentence must be represented as a tree, and trees do not admit of discontinuous constituents. If the grammar contained an infixation operation, the representation of the sentence would involve an object with crossing branches—not a tree. But the underlying assumption—that trees are sacred objects—makes little sense; a tree may or may not

morphological processes include infixation.) Bach dubbed the process in syntax as *Wrap*, and proposed that *give* first combines with *to Mitka* to form a complex transitive verb *give to Mitka*. Further, any complex transitive verb combining with its object (here, *the new toy)* does so by "wrapping" around its object. Bach's particular formulation of the operation referred to word boundaries: Wrap places one expression after the first *word* of the other. Wrap operations have been formalized in other ways as well. For example, Pollard (1984) suggests that the grammar refers not just to unanalyzed strings but *headed* strings. Each string has a distinguished head, and Wrap places the infix after the head of the other expression (or, in some languages, before the head of the other expression). If there are Wrap operations, the remarks made earlier regarding the role of structure in grammar are not quite correct: the grammar must keep track of just enough structure within an expression as to specify the infixation point. But none of the Wrap proposals requires the grammar to refer to structures as rich as full trees. We will formalize a variant of Pollard's proposal in Chapter 6.

This fourth solution is the most complex in terms of the syntax: it requires a Wrap operation (or a movement rule). Nonetheless, there seems to be some evidence for it (see section 15.4.2). To the extent that we deal in this book with three-place verbs, the Wrap solution is what we will (tentatively) adopt. (Note that this is compatible with the general architecture assumed here while the movement solution is not.) And even though the evidence for Wrap in the case of English is not entirely conclusive, there is independent evidence that grammars containing only the equivalent of context-free phrase structure rules will not be enough for other languages (see n. 5 and, e.g., Shieber 1985). Wrap operations have

be the appropriate representation as that depends entirely on what we discover is the correct rule system. (A tree is a sensible representation of how the rules work in a context-free phrase structure grammar; it is not a sensible representation in other systems.) Notice that if there are multiple levels of representation then it is also the case that a tree is not the appropriate representation for a sentence. Rather, a *sequence* of trees is. There is no particular reason to prefer a sequence of trees to some other kind of object. The important point is that *representations* represent the rule system(s); the task is to discover the kinds of rules employed by natural language.

been shown to be useful in the analysis of some of the key cases that lead to this conclusion.[11]

In any case, the main interest here is in how the semantics works. Given the hypothesis of Direct Compositionality, each of the four logically possible analyses of three-place verbs sketched earlier leads to a slightly different way to flesh out the semantics associated with the syntax. Here we consider only the possibilities labeled (3) and (4) above. While we have seen that (4) is syntactically more complex, (3) and (4) are equally straightforward in terms of the semantics. Either way, the idea is that—once again—the verb in the syntax is combining with its arguments one at a time. And since both of these are individuals, the resulting verb is of type <e,<e,<e,t>>>. The meanings would be different depending on whether we adopted (3) or (4). If (3) is correct, [[give]] as a function maps an individual which serves as the gift to a function taking the recipient to a function from givers to 1 or 0. If (4) is correct, the recipient is the first argument, the gift is the second argument, and the giver is the last argument in.

***5.7.** (i) If (1) were correct, what might be its associated semantics?
(ii) One way to give a semantics that goes along with (2) would be similar to the way you (hopefully) constructed for (i). Make this explicit.

[11] One such example is the analysis of "cross-serial dependencies" in Dutch given in Pollard (1984). The key phenomenon was discussed in Huybregts (1976). To illustrate the basic phenomenon, we use subordinate clauses (because Dutch, like most other Germanic languages, always has the main verb in second position in main clauses, which complicates the situation). Thus we find embedded sentences in Dutch like the following:

(i) ...dat Jan Piet Marie Cecilia zag helpen laten zwemmen
that Jan Piet Marie Cecilia saw help make Swim
"that Jan saw Piet help Marie make Cecilia swim"

This consists of a series of NPs and a series of verbs, and the "subject" of the nth verb in the verb chain is the nth NP in the NP chain. (This is not an isolated example but a systematic phenomenon.) While a set of context-free phrase structure rules could literally give the relevant string set it cannot give it in a way which—assuming a Direct Compositional view—will give a sensible semantics. Pollard (1984) showed that supplementing concatenation rules with Wrap can account for this.