The Interpretation of Noun Phrases Connected With *And*

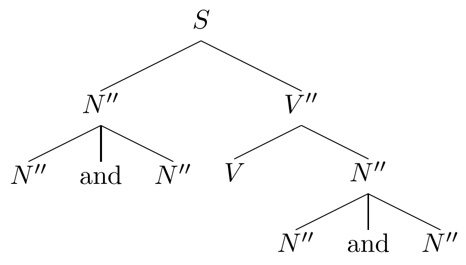
Jeff Kaufman

April 29, 2006

One aspect of English that we have not yet considered is how the conjunction *and* should be interpreted when it conjoins noun phrases. Consider the following examples:

1. In subject position:
   (a) John and Mary ate.
   (b) John and Mary met.
   (c) John and Mary kissed.
   (d) John and Mary bumped.
   (e) John and Mary collided.
   (f) John and Mary voted.
   (g) John and Mary attend Swarthmore.
2. In object position:
   (a) John ate pizza and cabbages.
   (b) John met Mary and Bob.
   (c) John bumped Mary and Bob.
3. In both:
   (a) John and Mary ate pizza and cabbages.
   (b) John and Mary met Bob and Sue.
   (c) John and Mary bumped Bob and Sue.
   (d) John and Mary attend Swarthmore and Haverford.

Let's take the generic syntactic structure to be as follows:



We had defined the extensions of N''s such as *John* and *Mary* to be entities, but that would be a stretch to use here. In *1f*, for example, we could evaluate by seeing if the set of things that have voted contains an entity *JohnAndMary* which represents *John* and *Mary* considered as a group. This would be incredibly wasteful, as then the set would need to contain every possible combination of voters, taking us from a set of cardinality N to one of cardinality $N!$.

A much nicer evaluation is to say that *1f* is true when it is true that *John voted* and that *Mary voted*. This sort of evaluation would also work with *ate* and *attend*. When we try to apply it to *met* or *bumped*, however, we run into problems. Applying it to *1b* we might think that *1b* would be true when it is true that *John met* and that *Mary met* but the real interpretation is closer to that *1b* is true when *John met Mary* and *Mary met John* are true.

What to do

We seem to have two choices:

1. Make $[\![$John and Mary$]\!]$ a set and give that set as an argument to the verb.
2. Make $[\![$and$]\!]$ be a function that gives its first two arguments to the function that is its third argument in the appropriate way.

Basically, we can put the duty of interpretation onto either the verb or the conjunction.

1 Verb

First, we'll need an extension for *John and Mary*. For the time being, let's take the following:

$$[\![\ N_1'' \text{ and } N_2''\ ]\!] = \{N_1'', N_2''\}$$

Evaluation will then be something like:

$$[\![\text{John and Mary ate}]\!] = ate'(\{john', mary'\})$$

Under this approach any verb that can be used with an *and*-connected argument needs to know how to parse the set appropriately. We also are restricted in that we can't have a verb like ate' sometimes be given entities, as in *John ate*, and sometimes sets. This requires us to consider all noun phrases as sets of entities.

Note that as *John* and *Mary* are noun phrases, the extension of *John and Mary* is not a set of entities but a set of sets of entities. That is,

$$[\![John\ and\ Mary]\!] = \{[\![John]\!], [\![Mary]\!]\} = \{\{john'\}, \{mary'\}\}$$

We would then evaluate a sentence of this form as:

$$\begin{aligned}
[\![John\ and\ Mary\ ate\ pizza\ and\ cabbages]\!] &= ([\![ate\ pizza\ and\ cabbages]\!])([\![John\ and\ Mary]\!]) \\
&= ((\lambda x \lambda y.\ ate'(y,x))(\{[\![Pizza]\!], [\![Cabbages]\!]\}))(\{[\![John]\!], [\![Mary]\!]\}) \\
&= ((\lambda x \lambda y.\ ate'(y,x))(\{\{pizza'\}, \{cabbages'\}\}))(\{\{john'\}, \{mary'\}\}) \\
&= (\lambda y.\ ate'(y, \{\{pizza'\}, \{cabbages'\}\}))(\{\{john'\}, \{mary'\}\}) \\
&= ate'(\{\{john'\}, \{mary'\}\}, \{\{pizza'\}, \{cabbages'\}\})
\end{aligned} \tag{1}$$

This is all well and good unless we want the other interpretation of this sentence, that *John ate pizza* $\wedge$ *Mary ate cabbages*. There's no way for ate' to put things together in the right way, as it is only being given sets and sets are by their nature unordered. We need something where ate' can match up things properly, something like a list. So lets redefine the extension of and:

$$[\![\ N_1'' \text{ and } N_2''\ ]\!] = \langle N_1'', N_2''\rangle$$

Now we can have each verb decide on its own how to parse its arguments. Some verbs will now have ugly extensions, but that's the domain of lexical semantics.

2 Conjunction

2.1 Examination

Instead of leaving the brunt of the analysis to the lexical semantics, we can go right ahead and put it all in the extension of *and*. This gives us much simpler extensions for verbs while pulling the meaning more out into the open. We can expect to practically do this because there aren't really that many ways that verbs process their arguments. Let's look over the examples presented earlier.

In *1a*, *1f*, and *1g*, we have a condensed form. That is, *John and Mary voted* is true exactly when *John voted* ∧ *Mary voted*. Similarly, *2a*, *2b*, and *2c* all expand the same way; there is no difference in extension between *John bumped Mary and Bob* and *John bumped Mary* ∧ *John bumped Bob*.

The cases of *1b* and *1c* seem to need a more complicated interpretation. We can't just say that *John and Mary kissed* is true when *John kissed* ∧ *Mary kissed* because each of those components is meaningless on its own. It makes no sense to say that a single person *kissed*. We need something more like *John kissed Mary* ∧ *Mary kissed John*.

The cases of *3a*, *3b*, *3c*, and *3d* are complicated because they have at least two possible extensions. The first is that *John and Mary met Bob and Sue* is true exactly when *John met Bob* ∧ *John met Sue* ∧ *Mary met Bob* ∧ *Mary met Sue*. The other is *John met Bob* ∧ *Mary met Sue*.

The final, slightly troublesome, one is the case of *1e*. If we want to treat it like the others we'd have to say *John collided with Mary* ∧ *Mary collided with John*. What is distasteful about this is non only that we're adding words, but that *collided* seems somehow more natural in its original form while the *collided with* form sounds like a workaround to bend the verb to our will. Consider what happens if we say that *collided* – not *collided with* – is a verb that takes two arguments of equal status? Then we can just use the reflexive case of *kissed* in a slightly wasteful manner.[1]

2.2 Implementation

We will need two senses of *and* to deal with our cases properly:

Dual: $(\lambda x \lambda y \lambda Q.\ Q(x) \wedge Q(y))$

Reflexive: $(\lambda x \lambda y \lambda Q.\ Q(x,y) \wedge Q(y,x))$

We can then evaluate most of our types of sentences:

$$\begin{aligned}
[\![John\ and\ Mary\ ate\ pizza]\!] &= ([\![John\ and\ Mary]\!])([\![ate\ pizza]\!]) \\
&= (\lambda Q.\ Q(john') \wedge Q(mary'))(\lambda x.\ ate'(x, pizza')) \\
&= ate'(john', pizza') \wedge ate'(mary', pizza') \\
[\![John\ and\ Mary\ kissed]\!] &= ([\![John\ and\ Mary]\!])([\![kissed]\!]) \\
&= (\lambda Q.\ Q(john', mary') \wedge Q(mary', john'))(\lambda x \lambda y.\ kissed'(x,y)) \\
&= kissed'(john', mary') \wedge kissed'(mary', john') \\
[\![John\ and\ Mary\ collided]\!] &= ([\![John\ and\ Mary]\!])([\![collided]\!]) \\
&= (\lambda Q.\ Q(john', mary') \wedge Q(mary', john'))(\lambda x \lambda y.\ collided'(x,y)) \\
&= collided'(john', mary') \wedge collided'(mary', john') \\
&= collided'(john', mary')
\end{aligned}$$

What we can't yet properly evaluate are sentences with multiple *and*s. Observe what happens if we try:

$$\begin{aligned}
[\![John\ and\ Mary\ ate\ pizza\ and\ cabbages]\!] &= ([\![John\ and\ Mary]\!])([\![ate\ pizza\ and\ cabbages]\!]) \\
&= (\lambda Q.\ Q(john') \wedge Q(mary')) \\
&\quad ((\lambda P.\ P(pizza') \wedge P(cabbages'))(\lambda x \lambda y.\ ate'(y,x))) \\
&= (\lambda Q.\ Q(john') \wedge Q(mary'))(\lambda y.\ ate'(y, pizza') \wedge ate'(y, cabbages'))
\end{aligned}$$

[1]This wasteful manner takes advantage of the fact that $collided'(a,b)$ would have the same extension as $collided'(b,a)$. It is shown in the computation of $[\![John\ and\ Mary\ collided]\!]$ below.

Now we have a problem. We can't apply our function because the types don't match properly. We need some way to deal with the boolean *and* in function application. We could say either of the following:

$$(\lambda Q.\, Q(a) \wedge Q(b))(\lambda x.\, C(x) \wedge D(x)) = C(a) \wedge D(a) \wedge C(b) \wedge D(b)$$

$$(\lambda Q.\, Q(a) \wedge Q(b))(\lambda x.\, C(x) \wedge D(x)) = C(a) \wedge D(b)$$

This confusion is actually quite useful, as we noted previously that sentences of this form have two interpretations. The ambiguity we noted earlier can be ascribed to the choice between the two different modes of combination.

As we have ambiguity rising from this choice we might also expect it over the choice of which sense of *and* to use? First note that because the reflexive *and* only works in subject position and when there are no objects. Verbs used with this form of *and*, however, are always transitive. So we have a rule:

> Whenever there is a transitive verb with no objects but with an *and*-connected noun phrase for the subject, the reflexive *and* is used, while otherwise the dual *and* is used.

3 Contrastature

Both of these interpretations could account for the use of *and*, and neither presents anything that seems testably false, so the best criteria for judgment are probably that the second approach is more elegant and would probably be easier for a person to learn.

One last bit of interest is how to deal with the word *respectively* in the two cases. In the first case it's quite simple, as respectively is an adverb affecting the main verb, and it can just transform that verb to one that always matches arguments as they come. In the second it has to change function application over the boolean *and* so that only the respective interpretation is allowed. That's a bit of a strange thing for an adverb to be doing, unless we think of it as a comment to remove ambiguity. It's analogous to me adding *habitually* to "I'm happy" to distinguish between the two ambiguous "generally happy" and "currently happy" interpretations.