PLIN0020 Advanced Semantic Theory — Yasu Sudo
Plural Predication — Autumn 2018

1 Conjoined Proper Names

Proper names denote entities, or equivalently, their semantic type is *e*. What do conjoined proper names like *Alice and Bob* denote? The phrase *Alice and Bob* is made of up three components, namely, two proper names–*Alice* and *Bob*–and *and*. The proper names denote entities. What does *and* denote?

1.1 Generalised Conjunction

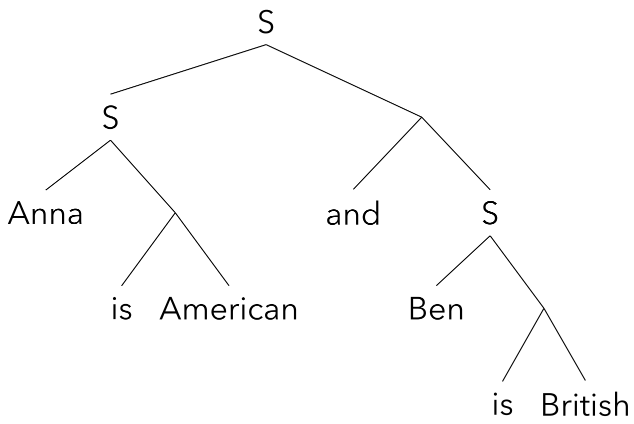
Let us start with the conjunction $\wedge$ in Propositional Logic. If you remember its truth-table, $(p \wedge q)$ is true iff both p and q are true. Here, p and q denote truth-values.

It seems that English *and* expresses something similar in sentences like the following, where *and* conjoins two (declarative) sentences.

(1) $[_{\text{S}}$ Anna is American] and $[_{\text{S}}$ Ben is British]

Let us assume the following structure.

(2)



It is easy to see that the semantic type of *and* here needs to be $\langle t, \langle t, t\rangle\rangle$. Adopting the analysis of $\wedge$ above, we arrive at the following meaning:

(3) $[\![\text{and}_{\langle t,\langle t,t\rangle\rangle}]\!]^M = [\lambda u \in D_t.[\lambda v \in D_t.\ 1 \text{ iff } u = 1 \text{ and } v = 1]]$

But *and* can also conjoin smaller constituents. For instance in (4a), VPs–which are of type $\langle e, t\rangle$–are conjoined, and in (4b), transitive verbs–which are of type $\langle e, \langle e, t\rangle\rangle$–are conjoined.

(4) a. Chris $[[_{\text{VP}}$ drank beer] and $[_{\text{VP}}$ danced]]
b. Daniel $[[_{\text{V}}$ made] and $[_{\text{V}}$ ate]] this cake all by himself.

In (4a), *and* takes two VP-denotations–which are of type $\langle e, t\rangle$–and produces another type-$\langle e, t\rangle$ function. In (4b), it takes two transitive verbs–which are of type $\langle e, \langle e, t\rangle\rangle$–and produces another type-$\langle e, \langle e, t\rangle\rangle$ function. These uses of *and* can be analyzed as follows. (Recall the shorthand: $et = \langle e, t\rangle$)

(5) a. $[\![\text{and}_{\langle et,\langle et,et\rangle\rangle}]\!]^M = [\lambda f \in D_{et}.[\lambda g \in D_{et}.\ \lambda x \in D_e.\ 1 \text{ iff } f(x) = 1 \text{ and } g(x) = 1]]$

b. $[\![\text{and}_{\langle\langle e,et\rangle,\langle\langle e,et\rangle,\langle e,et\rangle\rangle\rangle}]\!]^M$
$= [\lambda f \in D_{\langle e,et\rangle}.[\lambda g \in D_{\langle e,et\rangle}.\ \lambda x \in D_e.\lambda y \in D_e.\ 1 \text{ iff } f(x)(y) = 1 \text{ and } g(x)(y) = 1]]$

In some sense the semantic function of *and* is the same in all these cases: it takes two things of the same semantic type, and require both conjuncts to be true when all arguments are saturated (for $and_{\langle t,\langle t,t\rangle\rangle}$, there's no argument). There's actually a general way to state the meaning of *and* that encompasses all the above cases.

The general semantics for *and*—or *generalised conjunction*—makes use of the idea of 'semantic types that end in *t*', introduced at the end of the lecture notes for Week 4. The definition is repeated here:

(6) A semantic type τ *ends in t* if

a. $\tau = t$ or

b. $\tau = \langle\sigma_1, \sigma_2\rangle$ such that σ_1 is a semantic type and σ_2 a semantic type that ends in t.

Semantic types like t, $\langle e, t\rangle$, $\langle e, \langle e, t\rangle\rangle$, $\langle\langle e, t\rangle, t\rangle$, etc. end in t, while semantic types like e, $\langle et, e\rangle$, $\langle e, \langle e, e\rangle\rangle$, etc. do not (they end in e).

The meaning of *and* can be stated for any semantic type σ that ends in t as follows:

(7) For any model M,

$$[\![\text{and}]\!]^M = \lambda x \in D_\sigma.\lambda y \in D_\sigma. \begin{cases} 1 \text{ iff } x = 1 \text{ and } y = 1 & \text{if } \sigma = t \\ \lambda z \in D_{\tau_1}.\ [\![\text{and}]\!]^M(x(z))(y(z)) & \text{if } \sigma = \langle\tau_1, \tau_2\rangle \end{cases}$$

This meaning is *recursive* in the sense that $[\![\text{and}]\!]^M$ is mentioned in the definition of itself. However, it's not circular, since in all cases, the computation of meaning ends in the first clause without causing an infinite regress. To see this, let's consider a subcase where $\sigma = \langle e, \langle e, t\rangle\rangle$.

Then, both x and y will be of type $\langle e, \langle e, t\rangle\rangle$, i.e. the type of transitive predicates. Since $\sigma \neq t$, we have to use the second clause of the definition, and τ_1 will be e, because $\sigma = \langle e, \tau_2\rangle$ (where $\tau_2 = \langle e, t\rangle$). This means z will be an entity.

Now comes the recursion. We apply $[\![\text{and}]\!]^M$ to $x(z)$, and then to $y(z)$, as stated in the second clause of (7). Notice that the argument of $[\![\text{and}]\!]^M$, $x(z)$, will be a function of type $\langle e, t\rangle$, given the semantic types of x and z, and similarly $y(z)$ will be a function of type $\langle e, t\rangle$. So this time, $[\![\text{and}]\!]^M$ will conjoin two functions of type $\langle e, t\rangle$, i.e. $\sigma = \langle e, t\rangle$. Since $\sigma \neq t$, we are again in the second clause, but with a simpler type. Since $\sigma = \langle e, t\rangle$, z will be an entity as before, but $x(z)$ and $y(z)$ will now be truth-values. In the next cycle, then, $[\![\text{and}]\!]^M$ will take two truth-values, so $\sigma = t$, and we are in the first clause, and there will be no more recursion.

This whole computation can be illustrated in the following derivation. In each line, $[\![\text{and}]\!]^M$

is expanded according to (7), with renaming of variables to avoid confusion.

$$
\begin{aligned}
[\![\text{and}]\!]^M &= [\lambda f \in D_{\langle e,et\rangle}.\lambda g \in D_{\langle e,et\rangle}.\lambda z \in D_e.\ [\![\text{and}]\!]^M(x(z))(y(z))] \\
&= [\lambda f \in D_{\langle e,et\rangle}.\lambda g \in D_{\langle e,et\rangle}.\lambda z \in D_e. \\
&\qquad [\lambda P \in D_{\langle e,t\rangle}.\lambda Q \in D_{\langle e,t\rangle}.\ \lambda w \in D_e.\ [\![\text{and}]\!]^M(P(w))(Q(w))](x(z))(y(z))] \\
&= [\lambda f \in D_{\langle e,et\rangle}.\lambda g \in D_{\langle e,et\rangle}.\lambda z \in D_e. \\
&\qquad [\lambda P \in D_{\langle e,t\rangle}.\lambda Q \in D_{\langle e,t\rangle}.\ \lambda w \in D_e. \\
&\qquad\qquad [\lambda u \in D_t.\lambda v \in D_t.\ 1 \text{ iff } u = 1 \text{ and } v = 1](P(w))(Q(w))](x(z))(y(z))]
\end{aligned}
$$

Now in the final line, there's no more occurrence of $[\![\text{and}]\!]^M$ and λ-conversion will yield the same function as in (5b).

1.2 Proper Names as Generalised Quantifiers

Now coming back to *Alice and Bob*, the conjuncts here are proper names, and by assumption proper names are of type e, so generalised conjunction cannot apply to them. Fortunately, we can reanalyse proper names as generalised quantifiers, which are functions of type $\langle et, t\rangle$ and whose semantic type ends in t, so generalised conjunction will be able to apply to them.

We analysed the denotations of proper names as entities, because they are used to refer to specific entities. Interestingly, there is a certain relation between entities and generalised quantifiers of a specific kind. Let us take *Alice* as an example to illustrate it.

According to the type-e analysis, the proper name *Alice* denotes a specific entity/person. Let's call that person a. In a simple sentence of the form *Alice VP*, the VP denotes a function of type $\langle e, t\rangle$, which takes a and returns a truth-value. In the generalised quantifier analysis, $[\![\text{Alice}]\!]^M$ will be a generalised quantifier of type $\langle et, t\rangle$, so it will take the VP-denotation. What does it do to this function of type $\langle e, t\rangle$? It checks which truth-value it returns when it applies to a. That is, the generalized quantifier denotation of *Alice* looks like (8). $[\![\text{Alice}_e]\!]^M$ is its type-e denotation.

(8) For any model M,
$$[\![\text{Alice}_{\langle et,t\rangle}]\!]^M = \left[\lambda f \in D_{\langle e,t\rangle}.\ 1 \text{ iff } f([\![\text{Alice}_e]\!]^M) = 1\right]$$

As you can verify *Alice*$_e$ *VP* and *Alice*$_{\langle et,t\rangle}$ *VP* will always have the same truth-value. It's also instructive to see this generalised quantifier as a set. Since it is a function of type $\langle et, t\rangle$, it characterises a set, namely, the set of functions of type $\langle e, t\rangle$ that returns 1 when applied to $[\![\text{Alice}_e]\!]^M$, the entity the proper name *Alice* refers to in M. One can see this as the set of properties that this entity has. For instance, of $[\![\text{smokes}]\!]^M$ is in this set, it means this person smokes. If $[\![\text{likes dogs}]\!]^M$ is not in this set, it means this person does not like dog.

To put it differently, we can regard entities—which are of type e—and the set of properties they have—which is a set of type-$\langle e, t\rangle$ functions as the 'same thing'. Recall the discussion on characteristic functions. There, we discussed the one-to-one correspondence between functions of type $\langle \sigma, t\rangle$ and sets, which allows us to see them as representing the 'same thing'. The correspondence between e and $\langle et, t\rangle$ allows us to treat them as the 'same thing' as well. More specifically, any entity a can be 'lifted' to a generalised quantifier using the following recipe:

(9) $a \mapsto [\lambda f \in D_{\langle e,t\rangle}.\ 1 \text{ iff } f(a) = 1]$

The other direction is more complicated, but generally, a generalised quantifier Q created by (9) can be mapped back to a by the following operation:

(10) $Q \mapsto$ the entity a such that for each f such that $Q(f) = 1$, $f(a) = 1$

It is easier to understand this in terms of sets. set(Q) is a set of type $\langle e, t\rangle$ functions, all of which map a to 1, and it's the unique set that contains all such functions.

$$\text{set}(Q) = \{\, f \mid f \in D_{\langle e,t\rangle} \text{ and } f(a) = 1 \,\}$$

In fact, there is only one entity that these functions all map to 1, namely a, because if there were another such entity, say b, then a and b must have the same set of properties. However, there are properties like 'identical to a', which two entities cannot have simultaneously. In other words, the domain of functions of type $\langle e, t\rangle$ is rich enough to distinguish any two distinct entities x and y—there must be at least one function $f \in D_{\langle e,t\rangle}$ such that $f(x) \neq f(y)$. This must be the case, because $D_{\langle e,t\rangle}$ contains all functions from entities to truth-values.

It should be noted that the 'lowering' operation in (10) cannot apply to all generalised quantifiers. That is, there are generalised quantifiers that don't correspond to entities. For instance, take *somebody*. In terms of sets, it will denote:

$$\text{set}([\![\text{somebody}]\!]^M) = \{\, f \mid f \in D_{\langle e,t\rangle} \text{ and } f(x) = 1 \text{ for some entity } x \in D_e \,\}$$

In the general case there is no unique entity that all these functions map to 1. That is to say, there are more generalised quantifiers than there are entities. And the lowering operation in (10) is only defined for those generalised quantifiers Q that have a unique entity x such that for each $f \in \text{set}(Q)$, $f(x) = 1$ and in that case Q is mapped to that unique x.

1.3 Conjoined Generalised Quantifiers

Now we reanalysed proper names as generalised quantifiers of type $\langle et, t\rangle$, and generalised conjunction can apply to two generalised quantifiers, as in (11). In this case, $\sigma = \langle et, t\rangle$, so z will be of type $\langle e, t\rangle$.

(11)
$$\begin{aligned}
&\left[\!\!\left[\ [\ \text{Alice}_{\langle et,t\rangle}\ [\ \text{and}\ \ \text{Bob}_{\langle et,t\rangle}\ ]\ ]\ \right]\!\!\right]^M = [\![\text{and}]\!]^M\left([\![\text{Bob}_{\langle et,t\rangle}]\!]^M\right)\left([\![\text{Alice}_{\langle et,t\rangle}]\!]^M\right)\\
&= \lambda z \in D_{\langle e,t\rangle}.\ [\![\text{and}]\!]^M\left([\![\text{Bob}_{\langle et,t\rangle}]\!]^M(z)\right)\left([\![\text{Alice}_{\langle et,t\rangle}]\!]^M(z)\right)\\
&= \lambda z \in D_{\langle e,t\rangle}.\ 1 \text{ iff } [\![\text{Bob}_{\langle et,t\rangle}]\!]^M(z) = 1 \text{ and } [\![\text{Alice}_{\langle et,t\rangle}]\!]^M(z) = 1\\
&= \lambda z \in D_{\langle e,t\rangle}.\ 1 \text{ iff } z\left([\![\text{Bob}_e]\!]^M\right) = 1 \text{ and } z\left([\![\text{Alice}_e]\!]^M\right) = 1
\end{aligned}$$

This is a kind of meaning that can apply to a VP-denotation. Essentially, it will say that the VP-denotation holds both of Bob and Alice. This seems like a good analyses for sentences like (12).

(12) a. Alice and Bob sing well.
b. Alice and Bob are British.
c. Alice and Bob like spinach.

The generalised conjunction analysis says that these sentences should mean the same thing as (13).

(13) a. Alice sings well and Bob sings well.
b. Alice is British and Bob is British.
c. Alice likes spinach and Bob likes spinach.

However, a problem arises with the following types of examples.

(14) a. Alice and Bob met.
b. Alice and Bob live together.
c. Alice and Bob are a couple.

For example, the generalised conjunction analysis will say of (14a) that it'll be true iff $[\![\text{met}]\!]^M\left([\![\text{Bob}_e]\!]^M\right) = 1$ and $[\![\text{met}]\!]^M\left([\![\text{Alice}_e]\!]^M\right) = 1$, but what does this mean? The following sentences are all unacceptable.

(15) a. #Alice met and Bob met.
b. #Alice lives together and Bob lives together.
c. #Alice is a couple and and Bob is a couple.

So the generalised conjunction analysis works for predicates like *smoke*, *are British*, and *like spinach*, but not for predicates like *met*, *live together*, and *are a couple*.

2 Three Types of Predicates

What are the differences between the above two types of predicates? Intuitively, the former type of predicates describe properties of each entity. For example, the meaning of *like spinach* is inherently about one person. On the other hand, the latter type of predicates like *live together* are inherently about groups of people. It doesn't make sense to ask about one person whether this person lives together. The former group of predicates are called *distributive predicates* and the latter group of predicates *collective predicates*.

We can define distributive predicates as those predicates that support the *distributivity inference*. (We speak of VPs here, but we can generalise this to other syntactic categories.)

(16) A VP is said to have a **distributivity inference** if the following holds:
'Alice and Bob VP' entails and is entailed by 'Alice VP and Bob VP'.

Since mutual entailment amounts to truth-conditional identity, we can say that for distributive predicates, 'Alice and Bob VP' and 'Alice VP and Bob VP' are truth-conditionally synonymous.

For example *fell asleep* has a distributivity inference, because the following two sentences are synonymous.

(17) a. Alice and Bob fell asleep.
b. Alice fell asleep and Bob fell asleep.

Collective predicates, on the other hand, do not validate the distributive inference in either direction. Generally the right-hand side is simply unacceptable, so it doesn't even make sense to talk about entailment here. For example, *look alike* does not have a distributive inference.

(18) a. John and Mary look alike.
b. John looks alike and Mary looks alike.

The generalised conjunction analysis introduced in the previous section only works for distributive predicates, because it is made to derive the distributivity inference.

It should also be noted that there are VPs that are neither distributive nor collective. Here is one example.

(19) John and Mary bought a house.

This type of predicates validate the entailment in one direction.

(20) a. 'John and Mary bought a house' does not entail 'John bought a house and Mary bought a house'.
b. 'John and Mary bought a house' is entailed by 'John bought a house and Mary bought a house'.

It is easy to see that (20b) is true. For (20a), consider a situation where John and Mary, as a couple, bought a house by splitting the cost. Then it's not true that John bought a house (he did with Mary, but not on his own), and it's not true that Mary bought a house (she did with John, but not on her own). (Of course, there are situations where both sentences are true, namely situations where John and Mary each bought a house, but this is not enough to validate an entailment.)

Predicates like *bought a house* that only validate the distributivity inference in one direction are called *mixed predicates*.

Since the analysis in the previous section always derives the distributivity inference, mixed predicates are also problematic.

To summarise the discussion so far, we have the following three types of predicates.

- **Distributive predicates** (predicates that validate the distributivity inference)
  - 'John and Mary VP' $\Leftrightarrow$ 'John VP and Mary VP'
- **Collective predicates** (generally 'John VP and Mary VP' is unacceptable)
  - 'John and Mary VP' $\nRightarrow$ *'John VP and Mary VP'
  - 'John and Mary VP' $\nLeftarrow$ *'John VP and Mary VP'
- **Mixed predicates**: (predicates that only validate the entailment in one direction)
  - 'John and Mary VP' $\nRightarrow$ 'John VP and Mary VP'
  - 'John and Mary VP' $\Leftarrow$ 'John VP and Mary VP'

3 Plural Individuals as ‘I(ndividual)-Sums’

How do we account for collective and mixed predicates? We follow the following idea: phrases like *Alice and Bob* denote plural entities and the meaning of these predicates is inherently about such entities and cannot be reduced to predication of singular entities.

What are plural entities? One way to think about them is that they are entities just like normal entities like Alice, Bob, my laptop, etc. but unlike these ‘simple’ entities, they are composed of multiple entities. For example, the plural entity denoted by *Alice and Bob* will be an entity distinct from Alice and from Bob, but it has Alice and Bob as its parts.

More formally, we model plural individuals as *individual-sums*, or *i-sums* for short.

- Normal entities like Alice and Bob are from now on called *singular entities* (alt.: *atomic individuals*).
- The i-sum consisting of Alice and Bob is represented as ‘Alice$\oplus$Bob’.

Here ‘$\oplus$’ is the i-sum forming operator, and it has the following properties. For any entities (singular or plural) x, y, and z,

- $x \oplus y = y \oplus x$ (Commutativity)
- $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ (Associativity)
- $x \oplus x = x$ (Idempotence)

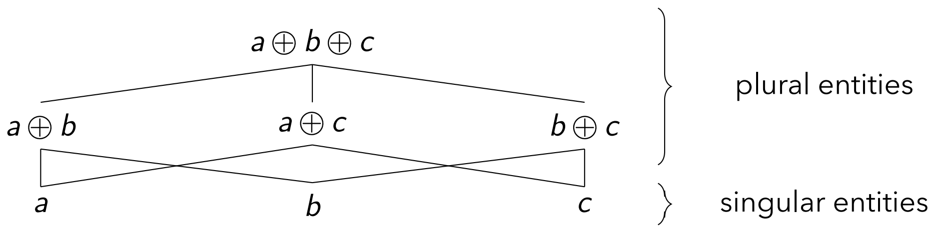
Since we have associativity, we often omit parentheses. That is, $a \oplus (b \oplus c)$ and $(a \oplus b) \oplus c$ represent the same entity.

Now we enrich the domain of entities D_e by ‘closing it with $\oplus$’. This means the following. D, the set of entities specified by the model M, is the set of singular entities. We have been assuming so far that $D_e = D$, but from now on, D_e will be assumed to contain also plural entities, in addition to singular entities. That is, for any two members x and y in D_e, we will also have $x \oplus y$ in D_e (this is what it means to ‘close’ D_e with $\oplus$, i.e. you have all possible plural entities in D_e based on D).

(21) a. Every member of D is a member of D_e (i.e. D is a subset of D_e)
b. Whenever x and y are members of D_e, $x \oplus y$ will also be a member of D_e.
c. Nothing else is a member of D_e.

For convenience, we will refer to the set of singular entities as SG, which is identical to $\mathcal{D}$, and the set of plural entities as PL. So $D_e = SG \cup PL$. Here’s a small example model with $D = \{ a, b, c \}$.

- $D_e = \{ a, b, c, \quad a \oplus b,\ a \oplus c,\ b \oplus c, \quad a \oplus b \oplus c \}$
- $SG = \{ a, b, c \}$
- $PL = \{ a \oplus b,\ a \oplus c,\ b \oplus c, \quad a \oplus b \oplus c \}$



Plural entities have other entities as *parts*. We denote the 'part-of' relation by $\sqsubseteq$. As the horizontal line in this symbol indicates, we take every entity, singular or plural, to be trivially a part of itself. But only plural entities have non-trivial parts. For example:

(22) a. $a \oplus b \oplus c \sqsubseteq a \oplus b \oplus c$
b. $a \oplus b \sqsubseteq a \oplus b \oplus c$
c. $a \oplus c \sqsubseteq a \oplus b \oplus c$
d. $a \sqsubseteq a \oplus b \oplus c$
e. $a \sqsubseteq a \oplus c$

(23) a. $a \not\sqsubseteq b \oplus c$
b. $a \oplus b \not\sqsubseteq a \oplus c$

The part-of relation is represented in the above diagram by the lines.

We assume that conjoined proper names denote plural entities. For example, for some specific model M_2:

(24) a. $[\![\text{Alice}]\!]^{M_2} = a$
b. $[\![\text{Bob}]\!]^{M_2} = b$

Then, we have:

(25) $[\![\text{Alice and Bob}]\!]^{M_2} = a \oplus b$

This use of *and* (call it $and_{\langle e,ee \rangle}$) therefore simply denotes $\oplus$ and is sometimes called 'non-Boolean conjunction'. (The function below is the same thing as '$\oplus$', but we put the λ's here to make the types of the arguments explicit)

(26) For any model M,
$[\![\text{and}_{\langle e,ee \rangle}]\!]^{M} = \lambda x \in D_e . \lambda y \in D_e .\ x \oplus y$

In this setting collective predicates can be understood as simply predicates that are only true of plural entities. For example, suppose that in M_1:

(27) a. $[\![\text{Bob}]\!]^{M_1}$ = b
b. $[\![\text{Chris}]\!]^{M_1}$ = c
c. $[\![\text{Daniel}]\!]^{M_1}$ = d

Suppose also that in this model, *b* and *c* look alike, but *b* and *d* do not. Then, we have:

(28) a. $[\![\text{look alike}]\!]^{M_1}([\![\text{Bob and Chris}]\!]^{M_1}) = [\![\text{look alike}]\!]^{M_1}(b \oplus c) = 1$
b. $[\![\text{look alike}]\!]^{M_1}([\![\text{Bob and Daniel}]\!]^{M_1}) = [\![\text{look alike}]\!]^{M_1}(b \oplus d) = 0$

By assumption, collective predicates are never true of singular entities. Thus, sentences like *Bob looks alike* is simply always false. One might argue that it is not only false but is also ungrammatical. Indeed, it feels a bit different from when *Bob and Daniel look alike* is false. There are several ways to capture this intuition. For example, sentences like *Bob looks alike* are not only simply false but also violate the presupposition of the predicate *look(s) alike* that the subject needs to be a plural entity. Such presuppositions on arguments are very common. For example, *is upset* presupposes that the subject is a sentient entity, and sentences like *My bike is upset* is either unacceptable or gets an anthropomorphic interpretation. Another analytical possibility here is that *Bob looks alike*, unlike *Bob and Daniel look alike*, is false in all possible models, and our semantic intuitions reflect this difference. We leave this issue open here.

What about mixed predicates? Mixed predicates are those predicates that are compatible with both singular and plural entities.

Suppose that in M_3, Alice refers to entity a, who bought a house alone, and Bob refers to entity b, who also bought a house alone, and these two are the only houses that were bought. Then, we have:

(29) a. $[\![\text{Alice bought a house}]\!]^{M_3} = [\![\text{bought a house}]\!]^{M_3}(a) = 1$
b. $[\![\text{Bob bought a house}]\!]^{M_3} = [\![\text{bought a house}]\!]^{M_3}(b) = 1$
c. $[\![\text{Alice and Bob bought a house}]\!]^{M_3} = [\![\text{bought a house}]\!]^{M_3}(a \oplus b) = 0$

Suppose now that in M_4, Alice refers to entity c and Bob refers to entity d, and c and d together bought a house, and that was the only house-buying in this model. Then:

(30) a. $[\![\text{Alice bought a house}]\!]^{M_4} = [\![\text{bought a house}]\!]^{M_4}(c) = 0$
b. $[\![\text{Bob bought a house}]\!]^{M_4} = [\![\text{bought a house}]\!]^{M_4}(d) = 0$
c. $[\![\text{Alice and Bob bought a house}]\!]^{M_4} = [\![\text{bought a house}]\!]^{M_4}(c \oplus d) = 1$

This implies that there shouldn't be no entailment between (31a) and (31b) in either direction.

(31) a. Alice bought a house and Bob bought a house.
b. Alice and Bob bought a house.

However, recall that there actually is an entailment from (31a) to (31b). This is a problem, because our semantics so far says that (31b) is only true if Alice and Bob bought a house together.

A related issue arises with distributive predicates. Given their meaning, we want to say that they can only be true of singular entities. For example, given that *Alice and Bob* denotes a plural entity, $[\![\text{fell asleep}]\!]^M([\![\text{Alice and Bob}]\!]^M)$ should always be false in any model M. But the following sentence is not always false.

(32) Alice and Bob fell asleep.

How do we account for such sentences?

One possibility is to assume that *and* is ambiguous between generalized conjunction and $\oplus$. Since the generalized conjunction does derive the distributivity inference, it will yield the correct reading with a distributive predicate, while the $\oplus$-reading of *and* would remain infelicitous. For mixed predicates, both generalized conjunction and $\oplus$ are compatible, but they yield different readings. That is with generalized conjunction, (31a) and (31b) will be synonymous, but not with $\oplus$. Notice that this correctly accounts for the one-way entailment. That is, whenever (31a) is true, the generalized conjunction reading of (31b) will be true, so in this direction, the entailment is valid. However, since (31b) is ambiguous and only one of the readings entails (31a), the entailment in this direction is not valid.

However, this ambiguity-based account has two empirical problems. The first problem is illustrated by (33).

(33) Andrew and Ben met in a pub and had beer.

This sentence contains a conjoined VP, where the first VP is a collective predicate *met in a pub* and the second VP is a distributive predicate *had beer*. Then, under either reading of *and*, there should be a problem. However, the sentence is still acceptable. Secondly, distributive predicates are generally compatible with plural expressions, even those that do not contain *and*, as shown in (34).

(34) a. They fell asleep.
b. The children fell asleep.

The plural definite subjects of these sentences presumably refer to plural entities, and as such it seems implausible to analysis to them as conjoined generalized quantifiers. The same point can be made with the one-way entailment of mixed predicates. Suppose that *they* in (35) refers to Andrew and Ben. Then, (35b) entails (35a) but not vice versa.

(35) a. They bought a house.
b. Andrew bought a house and Ben bought a house.

4 Distributivity

In this section, we will introduce an idea that solves both of the problems above, namely the idea of a *distributivity operator*. It is instructive first to consider the following sentence.

(36) Alice and Bob each bought a house.

This sentence is in fact synonymous with (37).

(37) Alice bought a house and Bob bought a house.

That is to say, unlike the version of (36) without *each*, it is no longer true in a context where Alice and Bob bought a house together. In a way, the word *each* here disambiguates the sentence.

What is the meaning of *each*? A standard analysis is that it denotes a distributivity operator. The idea is that in (36), for example, the distributivity operator applies the denotation of *bought a house* separately to each atomic part of $[\![\text{Alice and Bob}]\!]^M$.

(38) For any model M, $[\![\text{each}]\!]^M = \lambda f_{\langle e,t\rangle}.\lambda x \in PL.1$ iff for each $y \in SG$ such that $y \sqsubseteq x$, $f(y) = 1$

The variable y here ranges over singular entities (i.e. members of SG) that comprise the plural entity x, and the predicate f is applied to each y. This accounts for the meaning of (36) as follows.

(39)
$$\left\llbracket [_{\mathrm{S}}\ [_{\mathrm{DP}}\ \text{Alice}\ [\ \text{and}\ \text{Bob}\ ]\ ]\ [\ \text{each}\ [_{\mathrm{VP}}\ \text{bought}\ [_{\mathrm{DP}}\ \text{a}\ \text{house}\ ]\ ]\ ]\ ] \right\rrbracket^{M}$$

$$= \left\llbracket [\ \text{each}\ [_{\mathrm{VP}}\ \text{bought}\ [_{\mathrm{DP}}\ \text{a}\ \text{house}\ ]\ ]\ ] \right\rrbracket^{M} \left(\left\llbracket [_{\mathrm{DP}}\ \text{Alice}\ [\ \text{and}\ \text{Bob}\ ]\ ] \right\rrbracket^{M} \right)$$

$$= \left\llbracket [\ \text{each}\ [_{\mathrm{VP}}\ \text{bought}\ [_{\mathrm{DP}}\ \text{a}\ \text{house}\ ]\ ]\ ] \right\rrbracket^{M} \left(\llbracket \text{Alice} \rrbracket^{M} \oplus \llbracket \text{Bob} \rrbracket^{M} \right)$$

$$= \left[\begin{array}{l} \lambda f_{\langle e,t \rangle}.\lambda x \in PL. \\ 1 \text{ iff for each } y \in SG \\ \text{such that } y \sqsubseteq x,\ f(y) = 1 \end{array} \right] \left(\left\llbracket [_{\mathrm{VP}}\ \text{bought}\ [_{\mathrm{DP}}\ \text{a}\ \text{house}\ ]\ ] \right\rrbracket^{M} \right) \left(\llbracket \text{Alice} \rrbracket^{M} \oplus \llbracket \text{Bob} \rrbracket^{M} \right)$$

$$= 1 \text{ for each } y \in SG \text{ such that } y \sqsubseteq \llbracket \text{Alice} \rrbracket^{M} \oplus \llbracket \text{Bob} \rrbracket^{M},$$
$$\left\llbracket [_{\mathrm{VP}}\ \text{bought}\ [_{\mathrm{DP}}\ \text{a}\ \text{house}\ ]\ ] \right\rrbracket^{M} (y) = 1$$

$$= 1 \text{ iff } \llbracket \text{Alice} \rrbracket^{M} \text{ bought a house in } M \text{ and } \llbracket \text{Bob} \rrbracket^{M} \text{ bought a house in } M$$

We extend this idea to (40), which does not contain *each*.

(40) Alice and Bob bought a house.

That is, we assume that a covert distributivity operator is optionally present in this sentence, making the sentence ambiguous. When it is present, the sentence has the same meaning as (36), and it is not present, it only has the 'collective construal', which is true just incase Alice and Bob bought a house together. Let us denote the covert distributivity operator by Δ. Thus, we assume (40) to be ambiguous between the following two parses with two different readings:

(41) a. Alice and Bob Δ bought a house. $\Rightarrow$ Distributive reading
b. Alice and Bob bought a house. $\Rightarrow$ Collective reading

Now we account for the entailment from (42a) to (42b).

(42) a. Alice bought a house and Bob bought a house.
b. Alice and Bob bought a house.

That is, whenever (42a) is true, there is a parsing of (42b) that is true, namely (41a). In fact, the entailment goes through in the other direction too once we fix our parsing to (41a), as (41a) is synonymous with (42a) after all. That (42b) also has a parse (41b) without Δ, however, makes it look as if the entailment is only one way.

Turning now to distributive predicates, we assume that they are only true of singular entities. Consequently, when the subject is plural, Δ must be present to make the sentence acceptable.

(43) a. Alice and Bob Δ fell asleep.
b. *Alice and Bob fell asleep.

Finally, Δ simply cannot apply to a collective predicate, because, by assumption, a collective predicate cannot be true of singular entities.

(44) a. Alice and Bob look a like.
b. *Alice and Bob Δ look alike.

5 Further Readings

Bennett (1974) and Hausser (1974) are the earliest analyses of plurality in the framework of Montague Grammar. These authors analyse plural entities as sets, rather than i-sums. But such theories require a lot of redundancy in the lexicon for predicates that can apply to both singular and plural entities. For this reason, Scha (1981) proposes to treat singular and plural entities on a par, i.e. they are all sets. This is easy to do, the domain of singular entities and the domain of singleton sets are obviously isomorphic.

While these authors treat plural entities as sets (other authors that do so include Landman 1989a,b, 2000, Schwarzschild 1996, Winter 2001), Link (1983) advocates a mereological approach where the 'part-of' relation is taken to be the primitive and atomic entities are not necessarily required (see Champollion & Krifka 2016 for a linguistically-oriented overview of mereology; Varzi 2015 is also an accessible survey article on this topic). Link makes use of this ontology for the semantics of mass nouns. We will discuss mass nouns in Week 8.

The crucial difference between the set approach and the i-sum approach is that in the set approach, we can talk about sets of sets of entities, sets of sets of sets of entities, sets of sets of sets of sets of entities, etc., while in the i-sum approach, the structure of plural entities is 'flat', so-to-speak. There is a lot of discussion on whether such extra structure is necessary to account for the meanings of plural nouns phrases in natural language. If you are interested, read Landman (1989a,b) and Schwarzschild (1996), among others.

However, if we are only interested in sets of entities, the set approach and the i-sum approach are essentially the same, because the domain of sets of entities and the domain of i-sums are isomorphic.

Winter (2001:Ch.5) (a shorter version appeared as Winter 2002) discusses a different classification of predicates than the three-way classification we discussed above. His idea is motivated by plural vs. singular quantificational phrases, which we will discuss next week.

The distributivity operator Δ was originally put forward by Link (1987) and Roberts (1987). Scha (1981) proposed to build in the distributivity to the lexical entry of predicates, but there are cases involving distributivity at a non-lexical level. See also Landman (2000) and Winter (2001) for discussion on this. Schwarzschild (1996) discusses cases involving 'intermediate distributivity', which can be understood as distributivity over non-singular parts.

References

Bennett, Michael. 1974. *Some Extensions of a Montague Fragment of English*: University of California Los Angeles Ph.D. dissertation.

Champollion, Lucas & Manfred Krifka. 2016. Mereology. In Maria Aloni & Paul Dekker (eds.), *The Cambridge Handbook of Formal Semantics*, 368–388. Cambridge: Cambridge University Press. doi:https://doi.org/10.1017/CBO9781139236157.014.

Hausser, Roland. 1974. *Quantification in an Extended Montague Grammar*: University of Texas at Austin Ph.D. dissertation.

Landman, Fred. 1989a. Groups, I. *Linguistics and Philosophy* 12(5). 559–605.

Landman, Fred. 1989b. Groups, II. *Linguistics and Philosophy* 12(6). 723–744.

Landman, Fred. 2000. *Events and Plurality*. Dordrecht: Kluwer.

Link, Godehard. 1983. The logical analysis of plurals and mass terms: A lattice theoretical approach. In Rainer Bäuerle, Christoph Schwarze & Arnim von Stechow (eds.), *Meaning, Use, and the Interpretation of Language*, 302–323. Berlin: Mouton de Gruyter.

Link, Godehard. 1987. Generealized quantifiers and plurals. In Peter Gardënfors (ed.), *Generalized Quantifiers: Linguistic and Logical Approaches*, Dordrecht: Kluwer.

Roberts, Craige. 1987. *Modal Subordination, Anaphora and Distributivity*: University of Massachusetts, Amherst Ph.D. dissertation.

Scha, Remko. 1981. Distributive, collective and cumulative quantification. In Jeroen A. G. Groenendijk, Theo M. V. Janssen & Martin J. B. Stokhof (eds.), *Formal Methods in the Study of Language*, 483–512. Dordrecht: Foris.

Schwarzschild, Roger. 1996. *Pluralities*. Dordrecht: Kluwer.

Varzi, Achille. 2015. Mereology. In Edward N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, Spring 2015 edn.

Winter, Yoad. 2001. *Flexibility Principles in Boolean Semantics: The Interpretation of Coordination, Plurality, and Scope in Natural Language*. Cambridge, MA: MIT Press.

Winter, Yoad. 2002. Atoms and sets: A characterization of semantic number. *Linguistic Inquiry* 33(3). 493–505. doi:10.1162/002438902760168581.