And/or

About the complicated meaning of 'simple' coordinators

D. Dierikx

BA thesis Dutch Language & Culture
supervisor: dr. Y. Winter
co-supervisor: dr. E.G. Ruys

Utrecht University
Faculty of Humanities
august 2012

1 Contents

Introduction

While reading a paper regarding scope ambiguities(Ruys & Winter, 2010) I came across an example on page 11 that fascinated me:

(1) (Exactly) four teachers and authors smiled.

Two interpretations are mentioned right below the example ((2) and (3)) and two are mentioned in a footnote ((3) and (4)):

(2) "the sentence makes a claim about (exactly) four people, each of them a teacher and an author;
(3) [the sentence] makes a claim about four teachers and four authors.
(4) the sentence refers to four people, some of whom are teachers and the rest are authors;
(5) the constituency of the subject is *[exactly four teachers] and [authors].* "(Ruys & Winter, 2010)

Right after this example, there was a sentence presented with the word 'or' in it. And just like the 'and-example' it had different readings. But after spending just 19 lines of text on these multi interpretable sentences, the paper continued discussing other scope problems. But I wanted to know more about *and* and *or*! They are such small words, and can be used in so many occasions, but what exactly do they do? And what do they mean?

Job description of coordinators

The example above showed how complicated and ambiguous a sentence can become if you use *and*. but let's first take a look at some basic appearances of *and* and *or*. The main 'role' of these tiny words is that they can conjoin different words or word groups. Therefore they are called *coordinators*. We have already seen that coordinators can combine noun phrases (NP's) like *teachers* and *authors*. But *and* (and or) can also appear between two adjectives:

(6) Tina is [tall and thin].

And even whole sentences can be combined:

(7) [Tina is tall] and [Tina is thin].

In fact coordinators can take two (or more) words and/or word groups to combine them to form a new grammatically correct (part of a) sentence. It doesn't matter what category these parts have (noun phrase, adjective, sentence, etc), coordinators can handle almost anything. With only one restriction: the parts must be of the same category. [1] A sentence like (8) is therefore ungrammatical:

(8) *Tina is [tall_{Adj} and Mary_{NP}].

The fact that the same coordinators can combine multiple categories is called the *cross categorial behaviour of coordination*. At first sight this behaviour seems quite elegant because language users don't have to memorize different words for each different category they want to combine. They

[1] The article of (Sag, Gazdar, Wasow, & Weisler, 1985) lists and discusses examples of sentences where coordinated parts can be of different categories, such as: *Pat is either stupid or a liar*. In this paper I will focus on coordinations of elements of the same category.

simply can take an expression of category X, and expand that expression by combining it with another expression of the same category X and putting a coordinator in between:

(9) X $\rightarrow$ X coordinator X

The tricky part begins when we want to talk about the meaning of the coordination. Because X can be any category, the meaning of coordinated constructions varies with each different semantic category. Coordinators are therefore lexically ambiguous. If we want to construct a rule that tells us how (9) is interpreted, we have two options: we can come up with one rule that accepts different categories as its input, and manages to use the characteristics of each category to produce the correct denotation, or we have to make separate semantic rules for each category. The first option seems quite attractive, but can there be such an elegant theory? In other words:

> To what extent can there be one unified meaning of coordinators that respects their cross categorical behaviour? And if there is such a semantics, how can this meaning be written down in a non-ambiguous way?

I will begin this search by first looking at what elements we can use to construct our non-ambiguous semantics. This will make it easier for us to talk about meaning. While gathering our building blocks I will introduce an early attempt of writing down the meaning of coordinators: Conjunction Reduction. This is actually a quite limited approach and therefore we will gather more building blocks to construct the foundation of this paper: Boolean Semantics.

In chapter 2 and 3 I will discuss examples which show that coordinators aren't as simple as Boolean Semantics might imply and that the categories they combine still have a lot of influence on the possible meaning of the sentence. Chapter 2 will show the influence of intensional and extensional verbs on the application of Boolean Semantics. Chapter 3 will stray from Boolean Semantics and introduce the set product to denote the conjunction of two noun phrases.

1 The meaning of coordinators

The first thing we will need to do in order to write down the meaning of any language utterance, is to reach consensus on what meaning is. Many different philosophers have written even more books about the meaning of meaning, but here I will reason from the truth-conditional approach of semantics. This approach equals the meaning of a sentence with a description of the exact circumstances under which a sentence is true (Kerstens, Ruys, Trommelen, & Weerman, 1997). Now all we have to do is compose a system that translates natural language utterances to truth-conditions.

It is important that these truth-conditions are written down in a non-ambiguous system. We have already seen in the introduction that natural languages are ambiguous. Even though we are able to write down the truth-conditions of a sentence like (10) in English[2], denoting a sentence like (1), repeated as (11) below, in natural language would be very dangerous. This is because we would endanger ourselves of writing down a meaning that is also ambiguous.

(10)Mary sings.
(11)(Exactly) four teachers and authors smiled.

We thus need a system that links natural language utterances to situations in the real world under which the utterance is true. Logic can be used as such a system, so can mathematical symbols.

Before we continue building our semantics, we have already enough information to take a look at one of the earlier attempts to write down the meaning of coordinators.

1.1 Early steps: Conjunction Reduction

Conjunction Reduction came from Transformational Grammar and is a way of denoting conjunctions. This rule "maps sentential coordinations into phrasal coordinations."(Winter Y. , 2011) This means that every type of coordination, regardless of the category it combines, has the coordination of two sentences as underlying syntactic structure (called 'deep structure'). This means that (12) can be mapped as (13):

(12)Tina is tall and thin
(13)[Tina is tall] and [Tina is thin]

This (one!) semantic rule of coordination can thus be seen as a function that takes only sentences as its argument. However, the major problem of this approach is that it assumes the 'deep structure' (which is a matter of syntax) differs from the 'surface structure' (the way we use the expression: the actual utterance). But there is no syntactic motivation related to our semantic problem for those structural layers to be different. Instead of a rule that only deals with sentences, we need a system that lets us interpret smaller language units.

1.2 Type theory: denoting categories

A sentence like *Mary sings* consists of two words: a noun and a verb, and they can be combined to form a sentence. Here we use the verb *sing* in an intransitive way, meaning that it combines with only one noun phrase, that will be the subject of the sentence, and it has no object. When we use *to*

[2] In English, sentence (10) would be true if there is a world in which there is someone named Mary and that someone is performing the act called singing.

sing this way and we want to write down some kind of 'recipe' or 'sentence forming rule' we could note something like this:

(14)noun + intransitive verb → sentence (for example: *Mary sings*)

We can also use the verb *to sing* in a transitive way, meaning that it has to appear with both a subject and an object. We can adjust rule (15) by adding a noun:

(15)noun + transitive verb + noun → sentence (for example: *Mary sings a song*)

In linguistics we usually don't look at the whole utterance at once, but start to look at parts. We combine two of those parts to form a new constituent and combine that new constituent with another part of the sentence. In (15) we combined three things at once, but when we stick to the convention of only combining two things at the same time, rule (15) can be rewritten into (16)(a) + (16)(b):

(16)(a) transitive verb + noun → something (for example: *sings a song*)
(b) noun + something → sentence (for example: *Mary sings a song*)

If we take a closer look at (14) and (16)(b) we can see some similarity. It appears that what we have temporarily called *something*, may be the same thing as the intransitive verb in (14). For example, if we look at the example *Mary sings a song*, rule (16)(a) combines the transitive *sings* with *a song* and forms *sings a song*. This constituent can be seen as the intransitive verb *to sing a song* and we can apply rule (14) to form the whole sentence.

Instead of writing down rules for combining constituents and using names like noun, transitive verb, intransitive verb, 'something', etc., we can use the more elegant type theory[3] to talk about what kind of categories we're dealing with and how to combine them.

For this paper we can suffice with just two basic semantic types to analyze natural language, all other types can be deduced from those two. Type *e* refers to entities, words and expressions that point to individuals and things in the world. *Mary*, from our earlier examples, is an entity. The type *t* refers to truth values, things that can be true (then it will be assigned the value 1) or false (0). A sentence is an example of a truth value. All other types can be built by using the following definition:

Defenition 1[4]
T, the set of types, is the smallest set such that:

(i) $e, t \in \mathbf{T}$
(ii) if $a, b \in \mathbf{T}$, then $<a, b> \in \mathbf{T}$

This definition tells us that an expression is either an entity, a truth value or something of type <*a, b*>. An expression of type <*a, b*> is called a functor and can be applied to an expression of type *a*, called the argument, to form a combined expression of type *b*. When we think of our high school mathematics lessons, we can consider the expression type <*a, b*> a function. Later on we will see that we will write down the denotation of *Mary sings* as **sings**'(**m**), which conveniently resembles the

[3] This introduction to type theory is heavily based on chapter 4.2.2 Syntax from (Gamut, 1982).
[4] (Gamut, 1982), p. 79

mathematical function notation f(x) where f is the functor and x is the argument. Thus *sings* is of type <e,t> which takes something of type <e> to form an expression of type t.

In mathematical functions, the set of things a function can take as its argument is called its domain. In <*a, b*> we can therefore think of the set of elements of type *a* as the domain of <*a, b*>. In other words, “a domain is a part of the model that collects together objects with the same structure” and “the common structure of the objects in a given domain is described by a label that we call a type.”[5] When we refer to a domain of a specific type *a*, we use the notation D_a with *a* being any type. The elementary domains are D_e and D_t.

Now let’s return to our sample sentence. The expression $\text{Mary}_e\ \text{sings}_{<e,t>}$ is correctly formed by applying *sings* of type <e,t> to an expression of type e, i.e. *Mary*, to form a sentence which has indeed type t. Transitive verbs take two entities and therefore have type <e, <e, t≫. The transitive verb *sings*, can first combine with the entity *a song*, which gives us *sings a song* of type <e, t>, which can subsequently combine with *Mary* to form an expression of type t. When we compared (14) with (16)(b), we intuitively called ‘something’ from (16) an intransitive verb. Using the type-forming rules we have just introduced, we can confirm both expressions are of the same type.

1.2.1 Complications

Note that type theory provides us a simple and effective way to refer to and deal with different categories. What typed words or phrases don´t tell us, is which adjacent expressions can be combined. In *John loves Mary*, the type <e, <e, t≫ of *loves* does not say with which of the two entities it has to be combined first. Loves can combine with Mary first to form *loves Mary* of type <e,t> or it can combine with John to form *John loves*, which has the same type. For our purpose, type theory is effective enough because in this paper we want to show how meaning is formed based on the syntactic structure. Therefore we won’t be dealing with questions regarding the underlying syntactic structure of a phrase and assume that the syntax is known.

This brings us to another complicating issue regarding constructing a semantics for coordinated constructions. We will assume that when we know the meaning of different parts (say words or even smaller parts like morphemes), we will be able to construct the meaning of the whole construction. But in natural language, meaning doesn´t just depend on the denotations of separate words, but meaning also depends on the position of parts in a sentence. When we look at a sentence like *John loves Mary*, we have just seen that type theory doesn’t say anything about who will be the *lover* and who is the *loved* one. And the ambiguity in (1): *(Exactly) four teachers and authors smiled.*, primarily has syntactic roots. So if we want to construct semantic rules for coordinated constructions, we can’t ignore the underlying syntactic structure.

To summarize: if we want to write down the meaning of coordinators in an unambiguous way, we have to formulate unambiguous truth-conditions. By using type theory, we now have a convenient notation to write down the different linguistic categories. In the next paragraph we will take a look at a system that will help us combine different categories: set theory.

[5] (Winter Y. , 2010)

1.3 Set theory: combining categories

Let's take a look at sentence (17):

(17)Tina is tall

When do we consider (17) to be true? Well, there has to be someone called Tina, and that person has to be tall. But who is Tina? Tina Turner? My cat? Or some Tina that can be found on the wiki disambiguation page? And what does it mean to be tall? Anyone who is over 1.80 meter? Over 2.00 meter? This all depends on what is agreed. Therefore it is custom to define a model with everything that can be debatable. Entities will simply be denoted by a lower case letter, so *Tina* becomes **t** (denoting the entity, and has therefore type e, not t!) and *Mary* becomes **m**. If you want to visualise entities, you can think of them as points in a confined space:

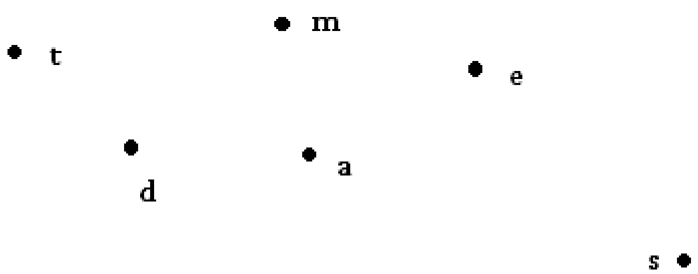


Figure 1 Entities in our model

But what about a property like being tall? Or a verb like singing? How can we visualise that? We solve this problem quite easily by saying which entities have that property. So when we say that someone is tall, we are referring to the collection of all the entities that are in that set of tall entities in our model. Being tall is simply a matter of being in that set of tall entities or not.

Our model of the world looks very simple: our world contains just entities that are or aren't part of sets. In Figure 2 below we see a world with entities (the dots) which can be named. There are entities that are part of the set of tall things, entities that are part of the set of thin things an entities that don't belong to any set.

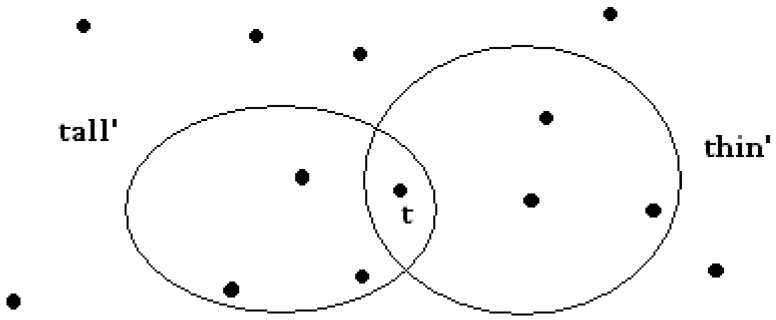


Figure 2 Intersection

Instead of drawing pictures of entities as dots with circles around them, there is a more convenient way of writing down sets. We can write the names of elements of a set down by placing them in between curly brackets. For example the set that contains a, b, and c is written as {a, b, c}. When we want to refer to that set, we can give it a name: A = {a, b, c} or **sing´** = {m, t}. Tabel 1 below shows our model.

E	{m, j, t}
sing'	{m, t}
write'	{j}
dance'	{m, t}
sleep'	{}
tall'	{t}
thin'	{t}

Tabel 1 Our model

To point to specific elements of a set, we can use the *element of* symbol: a ϵ A means that the element a is an element of the set that is called A. Earlier we represented *Mary* as **m**, so when we want to model the natural language sentence *Mary sings* in terms of set theory, all we have to do is to allocate a name to the set of singers (say **sing'**) and we can model the expression *Mary sings* by saying that m is an element of the set of singers: **m** ϵ **sing'**.

Sets can contain any number of elements, and the order in which they are written down is irrelevant. A set with no elements is called the *empty set* and has its own symbol: ∅ (which is the same as writing down {}).

Suppose we have a model with three entities: Mary, John and Tina (E = {m, j, t}) and the set of singers (**sing'** = {m, t}). We can also say that the set of singers is a subset of the set of all entities in our model: all elements that are in S are also in E. The notation is: **sing'** ⊆ E, which is the same as writing down {m,t} ⊆ {j,m,t}.

Another aspect of sets I want to point out is called the *difference* or the *complement*. This is a way of saying that something is *not* part of the set. The complement is noted with a line above the set name: $\overline{S}$ is the set of all entities that do not sing.

So far we have seen only sets of entities. But sets can contain anything, including other sets (empty or not). We can think of a set containing sets containing sets etc: {{a, b, c}, {{a}, {b}}, {}}. A special kind of set of sets is called the power set. This set contains all possible subsets of a set. Take for example the set of truth-values: {0, 1}. Now the power set of this set is: ℘{0, 1} = {{}, {0}, {1}, {0, 1}}. Power sets are always sets containing sets, and they consist of the empty set and the set itself and all possible combinations of elements in between.

1.4 The Boolean approach to coordination: similarities between categories

Now that we have introduced some elementary aspects of semantics, we can finally return to our problem of coordinating expressions and form a model for coordination (and negation) that does not harm the syntactic structure and at the same time deals with different semantic categories. The approach I want to introduce is called Boolean Semantics.

We have seen that different categories have different domains. But Boolean semantics acknowledges that these different domains “share a particular mathematical structure, known as a *Boolean Algebra*.”[6] So by finding similarities between different categories, it is possible to think of rules to semantically interpret coordinations, that hold for different domains. But what is this similarity, the so called Boolean structure?

Let’s take a look at our model and the entities E that ‘live’ in our model: E = {m, j, t}. We have seen in Tabel 1 that these entities can do different things in our model: sing, write, dance, sleep, be tall and be thin. These activities are all intransitive verbs and are of the same domain: <e, t>.

By using set theory we can model sentences like *Tina is tall* and *Tina is thin* by saying that the element representing *Tina* must be in the set of tall entities (**tall’**, which is in itself a subset of the set containing all elements in our model, E). The same holds for *Tina is thin* (with **thin’** being all the elements in the set that can be considered thin):

(18)t $\in$ **tall’**
(19)t $\in$ **thin’**

But how can we ‘combine’ (18) and (19) so that we can say *Tina is tall and thin*? What we want to say is that the entity Tina is both in set **tall’** and in set **thin’**. The set operator we use to denote just that is called *intersection* and is used in (20):

(20)t $\in$ (**tall’** $\cap$ **thin’**)

Recall that in Figure 2 we saw a visual representation of the intersection of set **tall’** and **thin’** and element **t** was in the overlap between the two circles.

But what if we change the coordinator and want to denote a disjunctive construction: *Tina is tall or thin*? What does a sentence like that mean when we look at it from a set perspective? Being tall *or* thin means that the element is part of one set (containing tall elements) or part of the other set (containing thin elements), or can be part of both sets. The set operator that means just that is called *union*, so the denotation of *Tina is tall or thin* becomes:

(21)t $\in$ (**tall’** $\cup$ **thin’**)

Let’s point out again that (20) and (21) describe the situations under which the sentences *Tina is tall and thin* and *Tina is tall or thin* are true; that is when they are part of the intersection/union in our model. A sentence like *Mary writes* (m $\in$ **write’**) is false in our model because the entity **m** is not part of the set of writers.

In Tabel 1 above we have seen a few subsets of the domain of entities (D_e), represented by intransitive verbs and adjectives . But what about all possible subsets of E? That would be the power set of E:

$\wp(E) = \wp(\{j, m, t\}) = \{ \emptyset, \{j\}, \{m\}, \{t\}, \{j, m\}, \{j, t\}, \{m, t\}, \{j, m, t\} \}$

[6] This quote and most of this paragraph is based on (Winter Y. , 2011). I have also used (Gazdar, 1980).

This power set represents all sets in the domain of entities that can appear in our model, and therefore $\wp(E)$ is part of our world. And together with the connectives *subset*, *union*, *intersection* and *complement* from set theory we introduced earlier, we now have tools (called a Boolean algebra[7]) to write down the meaning of a coordinated expression. But what about different domains? We still haven't shown the similarity between different domains.

In our example model we have used the set of entities. But we know there are lots of other domains. A special kind of Boolean domain is that of truth-values: {0, 1}. These values can also be represented by a power set of a set that contains only one element, for example x. The power set of a set with one element consists of the set containing the element (representing 1), and the empty set (representing 0):

$\wp(\{x\}) = \{\emptyset, \{x\}\}$

(22)(a) Mary sings and writes
(b) m $\in$ (**sings'** $\cap$ **writes'**)
(23)(a) Mary sings and John writes
(b) * (m, j) $\in$ (**sings'** $\cap$ **writes'**)
(c) * (m $\in$ **sings'**) $\cap$ (j $\in$ **writes'**)

Let's take a look at sentence (22) and (23). (22) is easy to denote with the standard intersection of two sets because one element must be part of both sets. But what about sentence (23)? Here two entities must be part of two different sets in order to be true. But how can we intersect them? We can't say that elements **m** and **j** are subsets of the intersection of the sets of singers and writers. Nor can't we look up element **m** in set **sings'** and element **j** in set **writes'** first to intersect those sets afterwards because this then we would have {**m**} $\cap$ {**j**}, which results in the empty set. The answer to this problem lies in the use of truth-values, just introduced above.

To denote a sentence like (23) we don't look at the intersection/union/complement of the sets of entities, but we combine sets that represent truth values. *Mary sings* denotes as a set containing x (= true), because the element **m** is indeed an element of the set of **sing'**. The same holds for *John writes*. The intersection of both sets of truth values also contains x, therefore the whole expression (23) is true. In (24), *Mary sings* is true and denotes x but *John dances* is false and therefore denotes as the empty set, which results in an intersection of {} and $\{x\}$, which contains no element, i.e. the whole expression denotes to false.

(24)(a) Mary sings and John dances
(b) **sing'**(**m**) $\cap$ **dance'**(**j**)

The advantage of looking at expressions denoting truth-values is that it allows us to use the binary operators union and intersection when we combine sets containing different entities as we have seen in (23). In the entity-approach it is impossible to ask about this intersection of two sets, where one set must contain the element **m** (but doesn't have to have **j**), and the other must contain the

[7] Barbara Partee explains an algebra in (Partee B.) as: "a set together with a collection of operations on this set. For example, the set of natural numbers and operations of addition and multiplication forms an algebra."[7] In this paper, we use sets as a model for our Boolean algebra, and the tools we introduced earlier: $\langle \subseteq, \emptyset, \cup, \cap, ^{-} \rangle$

element **j** (but **m** is not necessary). Such an intersection is impossible. But when we denote these expressions as truth values, we simply intersect sets that can only contain x or are empty. In fact, it appears that we can approach all domains as being a Boolean algebra, consisting of $\langle \subseteq, \wp(E), \cup, \cap, ^{-} \rangle$[8]. Therefore the following generalization is adopted by Boolean semantics:

> **"The boolean hypothesis**: All domains for denotations of expressions in natural language are boolean algebras. Coordination and negation cross-categorially denote the corresponding Boolean operators in each domain." (Winter Y. , 2011)

The D_t domain (with E = $\{x\}$) is the minimal non-trivial Boolean domain and from this point, we can use the type-theoretical information of categories to "deduce the boolean structure of many other domains. […] In general, all domains of functions with a boolean range are naturally defined as boolean algebras. We formally characterize the boolean types as follows.

> **Definition 1 (boolean types)**:
> Type t is boolean, and any type $\alpha\beta$ is boolean iff β is boolean."(Winter Y. , 2011)

This means that all types that 'end with t' are Boolean.

This is the similarity between different domains we have been looking for. In the following chapters we will use this theoretical background to look at some examples of coordination in English and find out if our Boolean approach lets us adequately coordinate different categories.

[8] Note that the domain of truth-values is a special kind of Boolean domain in which E = a set containing one element, to represent 0 (the empty set) and 1 (the set with that one element. Examples of this one element are x and {}.

2 Problem 1: Difference between intensional and extensional verb phrases

In 1982, Barbara Partee and Mats Rooth wrote an article called *Generalized Conjunction and type ambiguity*. In this paper, they also wanted to answer the question “whether we can give a single meaning for <u>and</u> and a single meaning for <u>or</u> that covers their uses across the full range of categories.” ((Partee & Rooth, Generalized conjunction and type ambiguity, 1983) p. 1) When outlining Montagues type theory, they soon discovered a problem between different kinds of verbs.

Now have Boolean Semantics we can easily denote sentences like *Mary sings and John writes* by intersecting two sets containing sets of sets rather than sets with different entities (14 c):

(25)(a) Mary sings and John writes
(b) **sings’**(**m**) ∩ **writes’**(**j**)
(c) **sings’**(**m**) = x, **writes’**(**j**) = x, so **sings’**(**m**) ∩ **writes’**(**j**) = x

When we look at transitive verbs in terms of types, verbs such as *sing* and *write* have type ≪<e,t>,t>, <e,t>>. In Richard Montagues famous paper *The proper treatment of quantification in ordinary English* (Montague, 1973) all transitive verbs have this type. Barbara Partee and Mats Rooth however, showed in their paper that this generalization doesn’t give us correct semantic predictions for all kinds of verbs.

2.1 One fish or two fish?

Let’s take a look at a sentence where two transitive verbs are conjoined, like (26), schematically written down in (27) and can be paraphrased as (28):

(26)John caught and ate a fish
(27)NP TV_1 and TV_2 T (for example: *John caught and ate a fish*)
(28)NP TV_1 T and NP TV_2 T (for example: *John caught a fish and ate a fish*)

NP is a noun phrase, TV_1 and TV_2 are different transitive verbs, and T is a term, the object of both TV_1 and TV_2. Now we want to give a denotation of (27), using our Boolean algebra assuming that both TV’s are of type ≪<e,t>,t>,<e,t>>. Partee and Rooth note this type as <type(T), type(IV)>[9], which is convenient because now we can see more clearly that when a transitive verb, like *catch*, is combined with a term, like *a fish*, we get the ‘intransitive verb’ *to catch a fish*$_{<e,t>}$. Recall that in our Boolean algebra we can depict something of type <<e,t>,t> as being a set of sets (with entities therein, as shown in Figure 3), and is also known as a higher order set. This results in the following description (in natural language!) of sentence (26): Both parts of the coordination (the verb *caught* as well as the verb *eat*) take as argument a set with sets of entities therein (they have type <<e,t>,t>). Within that set of sets there is a set of all things that are caught and all things that are eaten. In a picture:

[9] Be careful not to mix up capital T with small t. Capital T stands for term and small t is a truth-value. IV stands for intransitive verb.

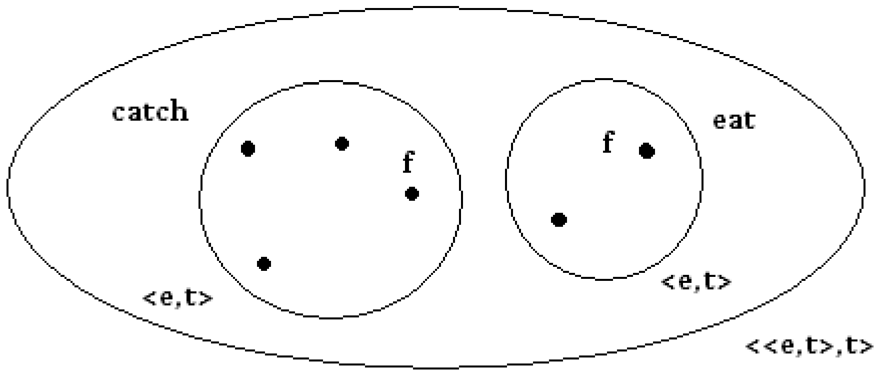


Figure 3 <e,t>,t

Because the verbs take type <<e,t>,t> type of things, it is very likely, when we look at Figure 3, that we are denoting different fish. But if we take a closer look at the meaning of (26), most people would agree that John caught and ate just one fish. This is because *catch* and *eat* are extensional verbs. Extensional verbs don't leave much room for different interpretations. This is different with intensional verbs like *want* and *need*. F. Moltmann describes intensionality as "nonspecificity (not substitutivity, existential commitment)" (Moltmann, 2008). Intensional verbs are sensitive to different interpretations. This becomes clearer when we consider the following situation.

Imagine John works for a company and he has way to much work to do. To reduce his work load, he considers hiring a secretary. But the tasks he has in mind for this secretary require other skills so John would actually be more helped if he hired an intern. When we take a look at sentence (29) we see that there is a discrepancy between what John wants (a secretary) and what he needs (an intern).

(29)John wants and needs a secretary

The thing John wants and needs doesn't have to be the same thing. There can be a set of things that John wants (including the entity *secretary*) and there can be another set with things John needs (with something that can be called secretary, but is actually an intern). And this is where the Montague type <type(T), type(IV)> is in place because the transitive verbs take sets of sets as their argument, not entities.

But we still haven't denoted the intensional sentence (26). How do we say that the fish is the same fish in both conjuncts? Visually, we want to say this:

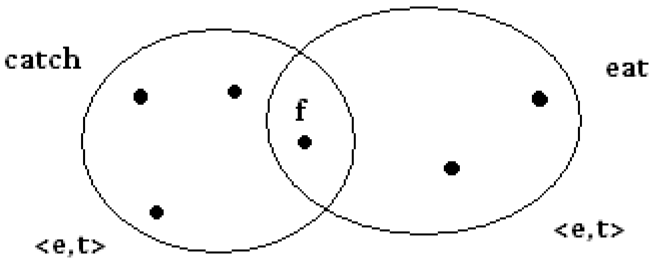


Figure 4 e,t

The obvious solution is to appoint extensional verbs the more simple type <e, <e,t>>. Notice that when we use <e, <e,t>>, the argument types aren't Boolean (because they aren't truth values), but the conjunction uses the Boolean operator intersection and therefore the whole type is Boolean. When we need to denote a conjunction or disjunction with different arguments, like (25)(a): *Mary sings and John writes*, the type <<e,t>,t> is convenient (and necessary!) because we already point at different entities.

2.2 Combining intensional and extensional verbs

The last question we can ask ourselves is: what to do when we conjoin an extensional and an intensional verb? Take for example sentence (37):

(30)John wanted and caught a fish

In natural language (37) can mean that John has been fishing and he had a particular fish in mind he wanted to catch. He did catch a fish, but it wasn't the fish he had in mind (different species, too small, etc.). So in this sentence, with both an extensional and an intensional verb, the intensionality of *wanted* leads to an intensional meaning of the whole sentence. In terms of semantics, both verbs should have the higher type ≪<e,t>,t>,<e,t>>. This means that we need 'a "redundancy rule" to insure that each "low-type" verb has predictable homonyms of higher type'(Partee & Rooth, Generalized conjunction and type ambiguity, 1983). The idea of Partee and Rooth to enter all verbs with its minimal type in the lexicon, is different from Montagues view. Montague wanted to denote all intransitive verbs with the higher <type(T), type(IV)> "and then provided meaning postulates to guarantee that the extensional verbs would behave semantically as if they were of the simpler type <e, type(IV)>." (Partee & Rooth, Generalized conjunction and type ambiguity, 1983).

So to conclude this chapter, intensional and extensional transitive verbs have showed us that we don't always need the higher type ≪<e,t>,t>,<e,t>> to denote transitive verbs. In fact there are situations in which this type gives the wrong denotation of a sentence. When we conjoin two intensional verbs, we do want the sentence to point to one and the same object and the simpler type <e,<e,t>>.

3 Problem 2: Conjunction within the determiner phrase

Up until now our conjunctions and disjunctions have occurred in rather isolated environments. One of the easier examples was the combination of two adjectives (*tall and thin*). The meaning of those two words didn't have any influence on the coordinated meaning, other than giving the subject two properties instead of one. Chapter 2 showed us an interesting case where the nature of at least one of the conjunct influenced the denotation of the whole coordination: *to catch and eat a fish*. In this chapter I want to go even further and show cases where the combination of the conjuncts can change the denotation of the whole sentence: turning a singular noun phrase into a plural one.

The contents of this chapter are entirely based on the article written by Caroline Heycock and Roberto Zamparelli: *Friends and colleagues: plurality, coordination, and the structure of DP* (2005).

3.1 Two kinds of conjunction: intersective and plurality-forming 'and'

Up until now we have only seen examples of so called *intersective and*. Sentence (6), here repeated as (31), is an example of *intersective and*:

(31)Tina is [tall and thin].

The denotation of (31) consists of the intersection of the set containing tall entities and the set containing thin entities. There is however a second type of conjunction: *plurality forming 'and'*. The easiest example is (32):

(32)[Tina and Mary] danced.

The coordination of two nouns in subject position makes the whole noun phrase plural and influences other parts of the sentence: for example, the verb has to change number. The type of conjunction combines the property that both *Tina* and *Mary* possess: *dancing*. Therefore this type of *plurality forming 'and'* is called *distributive conjunction*.

The more difficult type of plurality forming conjunction is shown in (33):

(33)[Tina and Mary] met.

The reason (33) is tricky to denote is because *meeting* is not a property that both *Tina* and *Mary* separately possess. To use the word *meet*, you have to have two entities that meet (you can't meet on your own!). The predicate *meet* is therefore non-distributive (and (33) is called *non-distributive conjunction*). Predicates like *meet* and *be together* can be called *collective predicates*. [10] In this chapter I will only focus on distributive conjunction.

[10] For those who like to see the different relations of all the introduced terms, here is a diagram (with examples), containing the terms discussed above:

- intersective 'and' (Tina is tall and thin)
- plurality-forming 'and'
  - distributive conjunction (Tina and Mary danced)
    - joint reading (This man and woman are in love)
    - split reading (My friend and colleague is writing a paper)
    - (mixed reading) (friend and colleague)
  - non-distributive conjunction (Tina and Mary met)

3.1.1 Distributive conjunction: joint and split readings

Now let's go back to the distributive conjunctions. It appears that within this type of conjunction, there can be another distinction. Take a look at the following sentences (sentence (5) and (6) from (Heycock & Zamparelli, 2005)):

(34)a. [$_{DP}$ My [$_{NP}$ friend and colleague]] **is** writing a paper.
  b. [$_{DP}$ That [$_{NP}$ liar and cheat]] **is** not to be trusted.
(35)a. [$_{DP}$ This [$_{NP}$ man and woman]] **are** in love.
  b. [$_{DP}$ This [$_{NP}$ soldier and sailor]] **are** inseparable.

For clarity I have highlighted the verbs in (34) and (35). This points out directly that the subject in (34) consists of only one person, while the subject NP consists of two coordinated nouns! On the other hand, (35) also contains coordinated subject NP's, but in these sentences the denotation of the subject consists of two people. The difference between the determiner phrases (DP's) in (34) and (35) is that of split and joint reading. A DP has a joint reading if the conjunction of two nouns denotes only one person. When the two conjuncts each denote a different entity, then the DP has a split interpretation.

In this chapter I want to talk about the solution Heycock and Zamparelli (H&Z) have proposed regarding the semantics of split and joint readings of conjunctions. It appears that we have to know more about the internal structure of determiner phrases and have to take into account two kinds of plurality: a syntactic and a semantic one.

3.2 Semantics of joint and split readings

In their chapter about conjunction, H&Z introduce the notion of set product. This construction allows the combination of elements from one set with the elements from another set to form a new set. Take for example the set of sailors: {{a), {b)) and the set of soldiers: {{c), {d)). Now the set product of *soldier and sailor* (and the denotation of the conjunction) is the set containing all possible combinations: {{a,c), {a,d), {b,c), {b,d)). Note that this denotation implies a split interpretation.

It gets interesting when one person is both a sailor and a soldier. The set of sailors now contains: {{a), {b), {c)) and the set of soldiers: {{c), {d)). The set product of *soldier and sailor* is now: {{a,c), {a,d), {b,c), {b,c), {c)). Because one possible solution of the set product is the singleton set {c), a joint reading is possible. Entity c is both a soldier and a sailor. When an expression can have both joint and split readings, we say it has a mixed reading. If there is no overlap between the two conjoined groups, only the split reading is possible in that model.

Unfortunately, language is not as straight forward as described above. Usually surrounding elements and cross linguistic variations tend to spoil the fun. With distributive conjunction, it appears that there are languages in which a split reading in a singular sentence is ungrammatical. In English (35)(b) is well formed. But its counterpart in French (and also Italian) (36) is ungrammatical:

(36)*[Ce [marin et soldat]] sont souvent ensamble
  [this [sailor and soldier]] are often together

In French and Italian only the joint reading is possible. (36) can be rewritten as *Ce marin et soldat **est** souvent ensamble* to imply the grammatical joint reading, but then how can one and the same person be ‘together’? In plural sentences both split and joint readings are possible in both French and English.

The existence of two readings for one conjunction and the fact that there are differences in languages that accept just one (only joint reading in singular sentences, such as French and Italian) or both (joint and split readings in singular sentences, as in English) interpretations, forces us to come up with a denotation system that takes these differences into account.

3.2.1 Syntactic and semantic number features

As part of their solution, H&Z propose 2 features for number: a semantic one called LATT (for ‘lattice’) and a syntactic one called PLUR (for ‘plural’). The syntactic number feature is the easiest to understand: +PLUR means that a noun phrase is syntactically plural. *Tina and Mary* is a textbook example of a syntactic plural NP. Just the noun *Tina* has the syntactic number feature -PLUR because just one noun is written down.

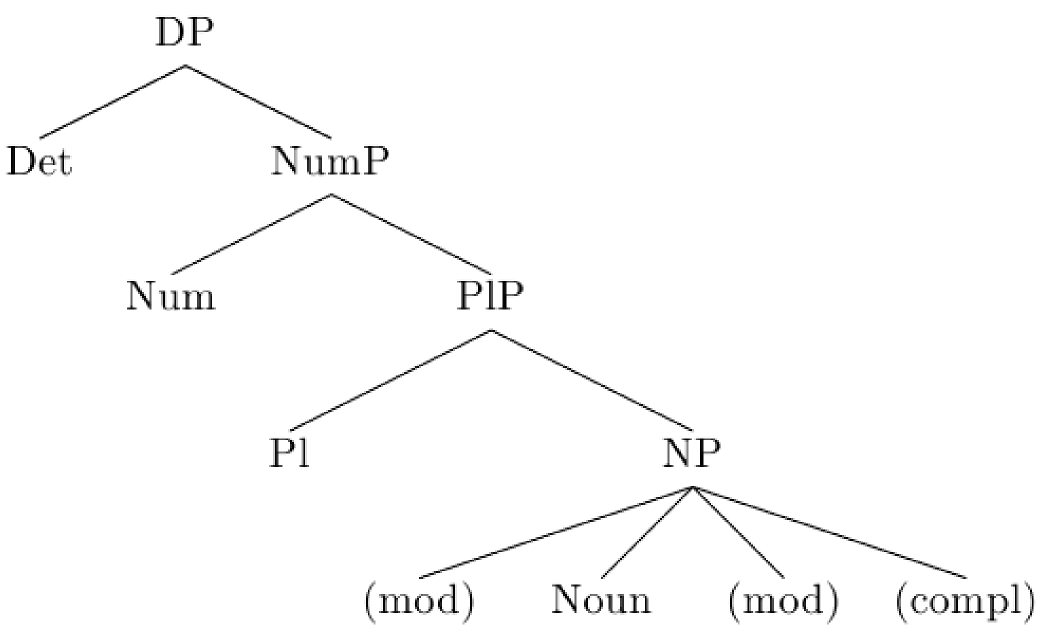
The semantic number feature takes into account the denotation of the noun phrase. +LATT means that a noun phrase is semantically plural. For example the NP *soldier and sailor* in (35)(b). denotes 2 different people (plurality!). But in sentence (34)(a), where two nouns are also conjoined, the semantic number feature is -LATT, because we are talking about just one person. This same sentence however has +PLUR because syntactically two nouns are mentioned.

3.2.2 The extended DP-tree

Another part of H&Z’s solution consists of the extension of the internal DP-structure. The most simple determiner phrase structure can be drafted like below:

(37)$[_{DP}$ the$_{Det}$ $[_{NP}$ apple$_{N}$]]

H&Z use an extended structure with additional projections:



Figuur 1 DP tree with extended projections

In NP we still compose our conjunction, like [$_{NP}$ Tina and Mary]. NP is also the place for the syntactic number features ±PLUR. The extra projections of PlP and NumP are used to store additional categories within the DP.

The reason why this extended projection is useful is because it appears that there is a difference between English and French/Italian in how the NumP position is used. Take another look at the English (46) and Italian (47) sentences below:

> (38)[$_{DP}$ the [$_{NP}$ soldier and sailor]
> (39)[$_{DP}$ il$_{[-LATT]}$ [$_{NumP}$ Num$_{[-LATT]}$ [$_{NP}$ soldado e marinado]]]

The English sentence has both split and joint readings: (38) is well-formed when there is one person in our model who is both soldier and sailor, and (38) is also well-formed when there is a single soldier in our model, and another distinct sailor. In English, the DP-structure is quite simple. But in a language like Italian, more extended projections of the DP tree are used. It appears that the overt determiner *il* has the -LATT feature and delivers that feature to the Num-position. Pl aquires the ±PLUR-value from the NP via the agree operation. But no matter what value Pl gets from the NP (in this case it will be the +PLUR-value because of the conjunction), in languages like Italian Num “filters the PlP denotation: only elements in the set which contain a single atom are preserved.” ((Heycock & Zamparelli, 2005) p. 243). And as we have seen at the beginning of section 3.2, singleton sets in the denotation of conjunctions refer to joint readings. So the conjunction can head for a split interpretation (via the +PLUR-feature), the determiner blocks this reading.

It may seem a bit complicated to introduce different number features and extended projection trees, but H&Z have written a paper packed with examples and explanations as to why they have chosen this approach. Not only are they addressing the distributive conjunction problem, but they also use their tools to denote mass and count nouns. I hope this chapter has introduced their approach concerning distributive conjunctions.

4 Conclusion

Now we have seen different ways to denote coordinated constructions, let's go back to our initial question:

> To what extent can there be one unified meaning of coordinators that respects their cross categorical behaviour? And if there is such a semantics, how can this meaning be written down in a non-ambiguous way?

I think it is now safe to say that the Boolean Semantics introduced in chapter 1 is a very elegant way to deal with coordinated constructions. By denoting coordinants as elements of Boolean structures instead of entities, a broader range of denotations can be captured.

Chapters 2 and 3 however have shown that a Boolean solution isn't always sufficient. Chpater 2 has shown that appointed types can sometimes be 'too much'. In other situations, as described in chapter 3, the syntactic structure of a construction still has a lot of influence on the possible meaning of an expression, which cannot be overlooked. The use of set products helps to denote the meaning of conjunctions and fits the already adapted method of set theory. But cross linguistic examples show that set products can give too many denotations. In those situations, the syntactic extended DP structure also has influence on the constructions' meaning.

I don't think a uniform semantics that covers all possible coordinations is possible, simply because there are so many preconditions involved. If there would be such 'one semantics', it would be very clumsy and difficult to use. Boolean Semantics is a good starting point, and from there on there can be made different 'subrules' to capture more specific, more demanding situations.

5 Bibliography

Gamut, L. (1982). 4.2.2 Syntax. In L. Gamut, *Logica, taal en betekenis 1.* Utrecht: Het Spectrum.

Gamut, L. (1982). *Logica, taal en betekenis 2*. Utrecht: Het Spectrum.

Gazdar, G. (1980). A cross-categorial semantics for coordination. *Linguistics and Philosophy 3* , 407-409.

Heycock, C., & Zamparelli, R. (2005). Friends and Collegues: Plurality Coordination and the Structure of DP. *Natural Language Semantics* , 201-270.

Kerstens, J., Ruys, E., Trommelen, M., & Weerman, F. (1997). *Plato's Probleem.* Bussum: Coutinho.

Moltmann, F. (2008). *Intensional Verbs and their Intentional Objects.* Retrieved 06 25, 2012, from Homepage of Frederike Moltmann: http://semantics.univ-paris1.fr/pdf/handout%20IntensVerbs-Yale.pdf

Montague, R. (1973). The Porper Treatment of Quantification in Ordinary English. In K. M. Hintikka, *Approaches to Natural* (pp. 221-242). Dordrecht: Reidel.

Partee, B. (n.d.). *Ling 409 Lecture Notes, Lecture 20: Introduction to Algebra and Algebras.* Retrieved 2012, from http://people.umass.edu/partee/409/Lecture%2020%20Algebra%20Intro.pdf

Partee, B., & Rooth, M. (1983). Generalized conjunction and type ambiguity. *Meaning, Use and Interpretation of Language* (pp. 361-383). Berlin: de Gruyter.

Ruys, E., & Winter, Y. (2010). Scope ambiguities in formal syntax and semantics. In D. a. Gabbay, *Handbook of Philosophical Logic – Second Edition* (pp. 159–225). Amsterdam: John Benjamins.

Sag, I., Gazdar, G., Wasow, T., & Weisler, S. (1985). Coordination and how to distinguish categories. *Natural Language and Linguistic Theory 3* , 117-171.

Winter, Y. (2010). Elements of Formal Semantics draft October 2010. *to appear* , 22.

Winter, Y. (2011). Flexibility Principles in Boolean Semantics: coordination, plurality and scope in natural language. In Y. Winter, *Flexibility Principles in Boolean Semantics: coordination, plurality and scope in natural language* (p. 8). Cambridge, Massachusett: MIT Press.

Winter, Y. (2011). Lecture notes on Boolean Semantics. Utrecht.