The many lives of *and*

Handout 1

Patrick D. Elliott

October 10, 2022

Contents

1 Reading

- Chapter 1 of (Winter 2001) (read this before next week's class).
- Additional background: (Partee & Rooth 1983)[1]

[1] This paper will probably be extremely hard to understand, since many of the notational conventions in compositional semantics have changed dramatically since the early 1980s. I've included it here in case you're interested in the history behind flexible coordination.

2 The flexibility of coordination

In the general case, selectional restrictions delimit the range of combinatorial possibilities:

(1) [$_{DP}$ Sally] went [$_{PP}$ to the shops]
(2) *[$_{DP}$ Sally] went [$_{DP}$ the shops]
(3) *[$_{PP}$ to Sally] went [$_{PP}$ to the shops]

The combinatorial restrictions we observe associated with a predicate such as *go*, are closely related to its lexical semantics and how sentences compose.

A truly remarkable property of natural language coordination (*and/or*; *und/oder*, etc.) is its syntactic flexibility.

(4) [$_{TP}$ Louise sneezed] and/or [$_{TP}$ Josie laughed].
(5) Louise [$_{VP}$ sneezed] and/or [$_{VP}$ barked].
(6) Louise is [$_{DegP}$ very badly behaved] and/or [$_{DegP}$ badly trained].
(7) Josie skipped down the street [$_{AdjP}$ happily] and/or [$_{AdjP}$ carelessly].
(8) [$_{DP}$ Every linguist] and/or [$_{DP}$ most philosophers] love the lambda calculus.

One of the main questions we'll be investigating in this class:

- What interpretive property of coordination *explains* its flexibility?

Ultimately, we'd like our semantic theory of coordination to account for the validity of certain conjunctive/disjunctive inferences:

(9) Every linguist and most philosophers love the lambda calculus.
⇒ *Every linguist loves the lambda calculus* **and** *most philosophers love the lambda calculus*
(10) Every linguist or most philosophers love the lambda calculus (I don't remember which)
⇒ *Every linguist loves the lambda calculus* **or** *most philosophers love the lambda calculus*

But hang on, these inferences don't always go through with *and* - sometimes it depends on the predicate!

(11) Josie and Sarah sneezed.
⇒ *Josie sneezed* **and** *Sarah sneezed.*
(12) Josie and Sarah met at the bowling alley.
$\not\Rightarrow$ *Josie met at the bowling alley* and *Sarah met at the bowling alley.*

(13) Josie and Sarah lifted the couch.
$\not\Rightarrow$ *Josie lifted the couch* and *Sarah lifted the couch.*

The "standard picture" which addresses this state of affairs is as follows:

- *and* in fact has two distinct lives:
  - *and* can convery **logical conjunction**, in which case the kinds of conjunctive inferences in (10) go through (although we still need to say something about combinatorial flexibility).
  - *and* can allow us to create a **group**, in which case the kinds of conjunctive inferences in (10) don't go through; see (13). The most famous incarnation of this idea is (Link 1983). We'll discuss this in some detail later in the semester.

This brings us to another central question in this class:

- Just how many meanings does *and* have? Why do we use the same operator for both logical conjunction and group formation?

Logical conjunction and group formation are sufficiently different operations that it seems like there's no escape from treating *and* as ambiguous, i.e., we have 'and_1' and 'and_2'.

This is a rather strange state of affairs - using *and* for both of these purposes is not just a quirk of English - overwhelmingly, cross-linguistically the same coordinator is used both for logical conjunction and sum formation.

As we'll see later in the semester, the situation is even worse than that - we can find evidence for a third incarnation of *and* (Link 1984).

The natural question is whether all of these different usages of *and* can be unified under a single, basic, semantics. At first, this seems to be an insurmountable problem, but the resolution will ultimately be related to *flexible* composition.

Before we tackle this harder problem, we'll begin by developing an analysis of flexible boolean conjunction.

3 Boolean conjunction in the STLC

3.1 Preliminaries: sentential coordination

In propositional logic, the semantics of conjunction is given via a truth-table:

$\phi \wedge \psi$	**true**	**false**
true	**true**	**false**
false	**false**	**false**

Just what information does a truth table encode? One way of thinking about this is that is tells us, for any pair of truth values representing the denotations of the individual conjuncts (t, t'), what the denotation of the conjunctive sentence u is.

In other words, a truth-table encodes a *function*.

In order to maintain a simple syntax-semantics interface, however, we're not going to treat *and* as a function from pairs of truth-values to truth-values, but rather as a *curried* function of type $T \to T \to T$.

(14) $\mathbf{and} : T \to T \to T$

This delimits the combinatorial properties of *and* such that it has to combine with each conjunct successively.

N.b., however we'll sometimes use infix notation as a notational convenience:

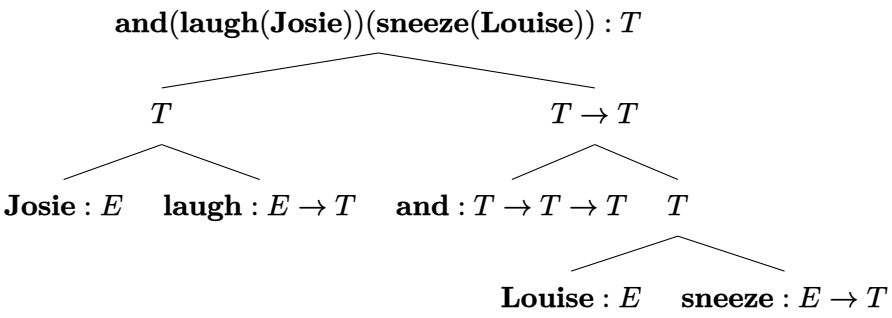
(15) $\alpha \, \mathbf{and} \, \beta := \mathbf{and}(\beta)(\alpha)$

The standard syntactic assumption (based on evidence from the binding principles; (Kayne 1994)), is that *and* composes with the latter conjunct first, followed by the initial conjunct. It should therefore denote the following function in $\mathbf{Dom}_{T \to T \to T}$.

(16) $$\begin{bmatrix} \mathbf{true} \to \begin{bmatrix} \mathbf{true} \to \mathbf{true} \\ \mathbf{false} \to \mathbf{false} \end{bmatrix} \\ \mathbf{false} \to \begin{bmatrix} \mathbf{true} \to \mathbf{false} \\ \mathbf{false} \to \mathbf{false} \end{bmatrix} \end{bmatrix}$$

This accounts neatly for sentential conjunction:

(17)



The sentence will be true just in case the denotations of both conjunctions are true, after (16).

However, the typing of *and* is rigid, it won't account for, e.g., VP-coordination or any varities of coordination we've seen beyond TP coordination (why?).

(18) Louise [sneezed] and/or [barked].

3.2 Conjunction reduction

One well-known approach to flexibility, which goes back to (Chomsky 1957), is to assume that the underlying syntactic structure of (18) actually involves sentential conjunction.

For Chomsky, this involved a special transformation rule, but a recent (re-)incarnation of conjunction reduction (Hirsch 2017) aims to derive conjunction reduction from the VP-internal subject hypothesis + ellipsis and across-the-board movement.

For example, for (18) all we need is ATB movement from SpecVP/vP:

(19) Louise$_1$ [t_1 sneezed] and [t_1 barked].

We're going to put this possibility to one side for the time being, and see how far we can get by developing a flexible compositional apparatus.

If there's interest, we might spend one class explicitly comparing flexible composition with syntactic approaches.

3.3 Flexible boolean coordination

Consider again our illustration of flexible coordination:

(20) [$_{\text{TP}}$ Louise sneezed] and/or [$_{\text{TP}}$ Josie laughed].

(21) Louise [$_{\text{VP}}$ sneezed] and/or [$_{\text{VP}}$ barked].

(22) Louise is [$_{\text{DegP}}$ very badly behaved] and/or [$_{\text{DegP}}$ badly trained].

(23) Josie skipped down the street [$_{\text{AdjP}}$ happily] and/or [$_{\text{AdjP}}$ carelessly].

(24) [$_{\text{DP}}$ Every linguist] and/or [$_{\text{DP}}$ most philosophers] love the lambda calculus.

3.3.1 Boolean types

Let's consider the *types* of the conjuncts:[2]

- **sneezed**(**Louise**) : T
- **sneezed** : $E \to T$
- **badlyBehaved** : $E \to T$
- **happily** : $(E \to T) \to E \to T$
- **everyLinguist** : $(E \to T) \to T$

Note that every type *ends in* T. What this tells us is that, once the function has all of its arguments saturated, it will return a sentential value.

This is exactly what (Partee & Rooth 1983) exploit in order to account for the flexibility of coordination.

First, we'll state a formal algorithm for determining the types of expressions which can be conjoined - following (Winter 2001) we'll call these the **boolean types**.[3]

Definition 3.1. Boolean types: T is a boolean type.

- If τ is a boolean type, then $\sigma \to \tau$ is a boolean type.
- Nothing else is a boolean type.

As an exercise, show whether or not $E \to (E \to E) \to T$ is a boolean type. Go step by step.

3.3.2 Generalized conjunction: $\sqcap$

The key intuition behind our account of flexible conjunction will be that some expressions have a *polymorphic* type-signature; instead of a fixed type, we have some *type variables* which must be resolved before composition can proceed.

The STLC doesn't really have the logical resources to capture this intuition directly.[4] One way of encoding this idea is that we have an algorithm *generating* constants, depending on the resolution of a given type.

[2] I'm taking a shortcut here by treating some complex expressions as constants. "Every linguist" should of course be decomposed as **every**(**linguist**), but its internal structure isn't relevant for the discussion here. One of the virtues of explicitly using an analytical tool such as the STLC is that these idealizations become transparent.

[3] (Partee & Rooth 1983) call these *conjoinable types*; it's exactly the same notion.

[4] This is because the STLC doesn't allow quantification over types; an extension of the STLC which allows (universal) quantification over types is called `SystemF`.

Following (Winter 2001), we'll write generalized conjunction as $\sqcap$. $\sqcap$ specifies a *family of logical constants* of type $\tau \to \tau \to \tau$, where τ is a boolean type.

We'll now state an algorithm for recursively generating $\sqcap$s.

Definition 3.2. Generalized boolean conjunction: Generalized conjunction specifies a family of expressions $\sqcap : \tau \to \tau \to \tau$, where τ is a boolean type. We'll write each expression as $\sqcap_\tau$

$$\sqcap_\tau := \begin{cases} \textbf{and} & \tau = T \\ \lambda p_{\sigma \to \rho} \,.\, \lambda q_{\sigma \to \rho} \,.\, \lambda x_\sigma \,.\; \sqcap_{\rho \to \rho \to \rho} \, (p(x))(p(x)) & \tau = \sigma \to \rho \end{cases}$$

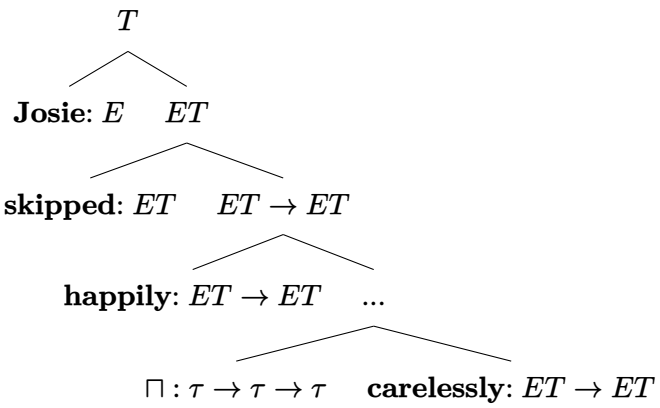
- $\sqcap_T = \textbf{and}$
- $\sqcap_{ET} = \lambda p_{ET} \,.\, \lambda q_{ET} \,.\, \lambda x \,.\, \textbf{and}(p(x))(q(x))$
- $\sqcap_{ET,T} = \lambda Q_{ET \to T} \,.\, \lambda \underline{Q}_{ET \to T} \,.\, \lambda p \,.\, \textbf{and}(Q(p))(\underline{Q}(q))$

Let's work through a complicated example together:

(25) Josie skipped [happily and carelessly]

Let's assume the following Logical Form:

(26)



(27) $\sqcap(\textbf{carelessly})(\textbf{happily})(\textbf{skipped})(\textbf{Josie})$

- What's the type of $\sqcap$ here? Resolve the definition.
- Reduce (27) (remember application associates to the left).

4 Generalized conjunction and DP coordination

4.1 QP coordination

Certain instances of DP coordination fall under the remit of generalized conjunction:

(28) Every linguist and most philosophers love the lambda calculus.

What's special about quantificational DPs like *every linguist*? unlike ordinary, individual-denoting arguments, the function-argument relation is reversed.

- $\mathbf{love}_{E\to E\to T}(\mathbf{LamCalc}_E)(\mathbf{Patrick}_E)$
- $\mathbf{everyLing}_{ET\to T}(\mathbf{love}_{E\to E\to T}(\mathbf{LamCalc}_E))$

N.b. it's often assumed that quantificational DPs undergo a covert movement operation (*quantifier raising*; (May 1977)). We'll talk about this more later in the semester, but for now observe that there is no motivation for such a transformation in the case of a subject quantifier such as *every linguist*.

Note, furthermore, that *every linguist* has a *boolean type*.

let's compute the expression $\sqcap_{ET\to T}$:

(29) $\sqcap_{ET\to T} := \lambda Q_{ET\to T} \,.\, \lambda Q'_{ET\to T} \,.\, \lambda P_{ET} \,.\, \mathbf{and}(Q(P))(Q'(P))$

Now we can compositionally translate (28) into the STLC.

The result should be the following:

(30) $\mathbf{and}(\mathbf{mostPhil}(\mathbf{love}(\mathbf{lamCalc})))(\mathbf{everyLing}(\mathbf{love}(\mathbf{lamCalc})))$

Note that generalized conjunction is a way of encoding something like conjunction reduction in the compositional semantic apparatus - essentially, the VP gets repeated in the argument of each DP. Just how different do you think these approaches really are?

Note that this derives the inferences:

(31) $\Rightarrow$ *Most philosophers love the lambda calculus*

(32) $\Rightarrow$ *Every linguist loves the lambda calculus*

4.2 Montague lift

Consider a simple example of DP coordination such as the following:

(33) Josie and Sarah love Bunny.

I've implied that perhaps we need a different variant of *and* for DP coordination, but now consider the following:

(34) Patrick and every philosopher love the lambda calculus.

An insight that goes back to (Montague 1973), is that even apparently individual-denoting expressions can be 'lifted' into expressions that denote quantifiers.

Recall from intro semantics that the standard treatment of expressions such as *every philosopher* is in terms of *generalized quantifier theory* (Barwise & Cooper 1981), i.e., as sets of sets of individuals.

In STLC, we reason about functions rather than sets; the way that we encode a set is via a characteristic function, therefore a set of sets of individuals is encoded via an expression of type $(E \to T) \to T$.

The intuition behind generalized quantifier theory is that a generalized quantifier must inspect the denotation of the VP in order to determine the truth of the sentence - for example, $\mathbf{everyPhil}(P)$ returns *true* iff P contains every philosopher.

In other words, **everyPhil** has the following semantics:

(35) $[\![\mathbf{everyPhil}]\!]\,(P) = \mathbf{true}$ iff $\{\, x \mid x \text{ is a philosopher} \,\} \subseteq \{\, x \mid P(x) = \mathbf{true} \,\}$

Now let's say we have a sentence such as "Patrick loves the lambda calculus".

- What property must the set derived from the VP denotation have in order for the sentence to be true?

We can reframe the meaning of **Patrick** as inspecting the VP denotation:

(36) $[\![\mathbf{Patrick}]\!]\,(P) = \mathbf{true}$ iff Patrick $\in \{\, x \mid P(x) = \mathbf{true} \,\}$

We encode this in the lambda calculus as follows (we'll write $\alpha^{\uparrow}$ for the "lifted" variant of expression α).

(37) $\mathbf{Patrick}^{\uparrow} := \lambda P_{ET} \,.\, P(\mathbf{Patrick})$

We can encode the lift operation itself in the STLC as a distinct function:

(38) $.^{\uparrow} := \lambda x \,.\, \lambda P \,.\, P(x)$ $\qquad E \to (E \to T) \to T$

Operations like *lift* are often referred to in the semantics literature as *type shifters* (Partee 1986).

There are some interesting questions to ask about the ontological status of these operators. For our purposes, we'll assume that a type-shifter can always be freely inserted in course of semantic composition.

We now have a way of converting **Patrick** into something with a GQ type $\lambda P \,.\, P(\mathbf{Patrick})$ (which denotes the set of sets that contain *Patrick*).

Show the following:

(39) $a^{\uparrow}(P) \Rightarrow P(a)$ $\qquad$ for any expression $a : E$, any expression $P : E \to T$

This means that it is harmless to assume that individual-denoting subject DPs are always lifted (in fact, this is exactly what Montague assumed).

More importantly, this allows us to account for (34)! (show this in detail).

(40) $\sqcap_{ET \to T}(\mathbf{everyPhil})(\mathbf{Patrick}^{\uparrow})(\mathbf{love}(\mathbf{lamCalc}))$
$\Rightarrow$

Derived inferences:

- *Patrick loves the lambda calculus*
- *Every philosopher loves the lambda calculus*

In fact, we can now every account for conjunction of individual-denoting DPs with generalized conjunction.

(41) $\sqcap_{ET \to T}(\mathbf{Josie}^{\uparrow})(\mathbf{Sarah}^{\uparrow})(\mathbf{love}(\mathbf{Bunny}))$
$\Rightarrow \mathbf{and}(\mathbf{love}(\mathbf{Bunny})(\mathbf{Josie}))(\mathbf{love}(\mathbf{Bunny}(\mathbf{Sarah})))$

Derived inferences:

- *Josie loves Bunny*
- *Sarah loves Bunny*

Note that this still won't make the right predictions for certain kinds of predicates:

(42) Sarah and Josie met.

In fact, we can even find an incarnation of the same kind of problem with bona fide QP coordination.

(43) Josie and every linguist met.

Informally, this can mean that, for every linguist x, the group consisting of Josie and x met.

Next week we'll discuss an approach to conjunction with predicates such as *met*.

References

Barwise, Jon & Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4(2). 159–219. https://doi.org/10.1007/BF00350139 (20 September, 2021).

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton.

Hirsch, Aron. 2017. *An inflexible semantics for cross-categorial operators*. Massachussetts Institute of Technology dissertation.

Kayne, Richard S. 1994. *The antisymmetry of syntax* (Linguistic Inquiry Monographs 25). Cambridge, Mass: MIT Press. 195 pp.

Link, Godehard. 1983. The logical analysis of plurals and mass terms: A lattice-theoretic approach. In P. Portner and B. H. Partee (ed.), *Formal semantics - the essential readings*, 127–147. Blackwell.

Link, Godehard. 1984. Hydras. On the logic of relative clause constructions with multiple heads. In *Varieties of formal semantics: Proceedings of the fourth amsterdam colloquium, september 1982*, 245–257. Dordrecht Foris. https://dare.uva.nl/search?identifier=2adcbc0e-4a5d-4855-b58b-d94ac3cd3d0b.

May, Robert. 1977. *The grammar of quantification*. Massachussetts Institute of Technology dissertation.

Montague, Richard. 1973. The Proper Treatment of Quantification in Ordinary English. In K. J. J. Hintikka, J. M. E. Moravcsik & P. Suppes (eds.), *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics* (Synthese Library), 221–242. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-010-2506-5_10 (8 February, 2020).

Partee, Barbara. 1986. Noun-phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh & M. Stokhof (eds.), *Studies in discourse representation theory and the theory of generalized quantifiers*, 115–143. Dordrecht: Foris.

Partee, Barbara & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In *Meaning, use, and interpretation of language*, Reprint 2012, 361–383. Berlin, Boston: De Gruyter. https://www.degruyter.com/view/product/10320.

Winter, Yoad. 2001. *Flexibility principles in boolean semantics - the interpretation of coordination, plurality, and scope in natural language* (Current Studies in Linguistics 37). Cambridge Massachussetts: The MIT Press. 297 pp.