Language Tutorial
Polysemy, copredication and individuation
Day 2: The Implications of Polysemy for Semantics

Peter R. Sutton
Universitat Pompeu Fabra

TbiLLC 2023
18-22. September 2023
http://bit.ly/tbillc-tutorial

Universitat Pompeu Fabra Barcelona
beatriu de pinós
Co-funded by the European Union
Generalitat de Catalunya Departament de Recerca i Universitats

Topics for this tutorial

1. Polysemy and Copredication (focussing on common nouns)
   - Differentiating polysemy from other phenomena

   Question: Can we discriminate polysemy from other phenomena (e.g., lexical ambiguity, coercion, underspecification)?

2. Implications of polysemy and copredication in semantics
   - Chomsky's Argument
     - Polysemy and copredication force an abandonment of (externalist) truth-conditional semantics
   - Semantic accounts of polysemy

   Question: What are the implications of polysemy and copredication for semantic theory?

3. Polysemy, copredication, and quantification
   - Quantification and copredication over plural NPs
     - *three long misleading talks*

   Question: What roles do modifiers and quantifiers play in restricting the individuation criteria of common nouns (semantics/pragmatics interface)?

Outline for day 2

Recap on a version of Chomsky's argument

- Why is polysemy a challenge for canonical natural language semantics?

Some of the proposals for analysing polysemy and copredication

- Same type responses
- Richer type responses

Recap: Chomsky's argument

(Chomsky, 2000; Collins, 2017; Pietroski, 2003, among others)

- Nouns such as *book* are polysemous, not lexically ambiguous.
- However, some but not not all of the following uses of *book* evoke both senses:

(1) Collins 2017, p. 679

a. Bill memorised the book

b. Bill burnt the book

c. Bill memorised and (then) burnt the book

If polysemous nouns had an invariant, truth-conditional meaning, then cases of copredication like (1c) would be anomalous, contrary to fact.

- Therefore, nouns like *book* do not have an invariant, truth-conditional meaning.

Recap: Formalising the argument

An adapted version of the argument:

(a) non-ambiguous nouns (including polysemous nouns) denote functions e.g., from worlds/situations to sets of entities, and
(b) if informational entities, eventualities and physical entities etc. are of a different type, then
(c) there is no function expressible in the simply-typed λ-calculus that can characterise a set of entities that are, say physical and/or informational/eventualities

Recap: Responding to the argument

(a) non-ambiguous nouns (including polysemous nouns) denote functions e.g., from worlds/situations to sets of entities, and

(b) if informational entities, eventualities and physical entities etc. are of a different type, then

(c) there is no function expressible in the simply-typed λ-calculus that can characterise a set of entities that are, say physical and/or informational

Options:

- deny (a) – We'll mostly set this aside
- deny (b) – the *same type response* – we'll look at two strategies
- Shrug regarding (c) and use a richer type theory – we'll look at a few options.

Recap: The problem for traditional simply-typed semantics

Standard assumption:

- At least some of the types for propositions/informational content, eventualities, physical entities etc. are different and discrete

Data:

- Polysemous nouns have senses denoting, e.g., propositions, eventualities and physical entities (e.g., *statement*, *evidence*)

Problem:

- Example: *lunch*
  - Assumption: eventualities and physical stuff (food) are of different types (in disjoint domains)
- $\lambda w_{:s}.\lambda x_{:\sigma}.LUNCH_w(x) \quad : \langle s, \langle \sigma, t \rangle \rangle$
- What type is σ?
  - Can't be v or e (this would exclude some readings of *lunch*)
  - Can't be a functional type (wrong truth conditions)

The same type solution in outline

Standard assumption:

- At least some of the types for propositions/informational content, eventualities, physical entities etc. are different and discrete

Solution:

- Deny the standard assumption

Example:

- *lunch* denotes eventualities and physical entities (i.e., lunching events and food)
- Let's assume that both lunching events and food are in the domain of type σ
- $[\![\text{lunch}]\!] = \lambda w.\lambda x_{:\sigma}.LUNCH(x) \quad : \langle s, \langle \sigma, t \rangle \rangle$

Proponents of the same type solution

Informally

- Suggestions made in the philosophical literature (Liebesman and Magidor, 2017, 2019)
  - A plausible way of caching this out in terms of a mono-typed semantics (Liefke, 2014; Liefke and Werning, 2018)

Formally

- Proposal made within frame semantics à la Düsseldorf (Kallmeyer and Osswald, 2017; Babonnaud et al., 2016)
  - Arguably a form of mono-typed semantics
    - Plus *sorts* (to replace types)
    - Plus added structure (i.e., *frames*)
- Mereology (Gotham, 2014, 2017, 2021)
  - Polysemous nouns denote mereological sums of entities of different ontological sorts
  - Skipped today, returned to tomorrow when we look at quantification

Liebesman & Magidor's proposal

Idea 1: Indications of a single type view

"accounting for copredication requires no revisionary semantics or metaphysics, and that copredication is perfectly compatible with standard referential semantics . . . we'll argue that book has a single sense and it designates both informational and physical books" (Liebesman and Magidor, 2017, p.132)

Idea 2: Property inheritance

"Informational books are distinct from physical books, but there are many properties that both can instantiate." (Liebesman and Magidor, 2017, p.137)

Property inheritance

(2) Three interesting books are on the shelf.

- Properties can be inherited via association relations
- (2) do not force us to explain how we can copredicate over different sorts of entities
  - This sentence can straightforwardly be about physical books described as interesting based on an inheritance of the properties of their contents
- And vice versa: *book* can denote informational books and prima facie physical predicates can apply to these based on property inheritance

(3) Mao's red book brought about many political changes despite being small.

Basic implementation within simple type theory?

- **BasTyp** $= \{e, t, s\}$
- Functional types constructed recursively (e.g., $\langle e, t\rangle$, $\langle s, \langle e, t\rangle\rangle$, etc.)

Possible implementation

- Polysemous nouns denote properties of type $\langle s, \langle e, t\rangle\rangle$
- Physical entities, eventualities etc. in the domain of type e

Challenge

- What about informational/propositional denoting nouns e.g., *statement*?

Possible response

- Assume that informational entities are in $\mathcal{D}_e$
- Assume mapping functions from $\langle s, \langle e, t\rangle\rangle$ to e (and vice versa?)

But there may be a way to avoid positing these mapping functions

Alternative implementation within simple type theory

Assume that interpretations of DP and S are of the same type

- Proposals in e.g. Liefke 2014; Liefke and Werning 2018 (see also Partee 2007)

(4) (Liefke and Werning, 2018, p. 646)

a. [$_{\text{DP}}$ Bill] destroyed his friendship with John.

b. [$_{\text{CP}}$That Bill suspected John of courting Pat] destroyed his friendship with John.

(5) Pat remembered [[$_{\text{DP}}$ Bill] and [$_{\text{CP}}$ that he was waiting for her]]. (Liefke and Werning, 2018, p. 647)

Implementation (simplified)

(6) a. **BasTyp** $= \{\sigma\}$ (the type for $[\![[_{DP} \cdot]]\!]$ and $[\![[_{S} \cdot]]\!]$)
b. Functional types constructed recursively

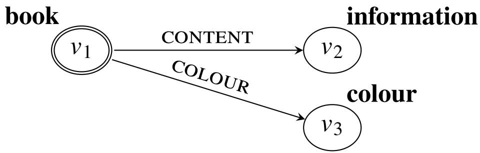
(7) a. $[\![$ $[_{DP}$ lunch $]$ $]\!]$: σ
b. $[\![$ $[_{VP}$ was delicious $]$ $]\!]$: $\langle \sigma, \sigma \rangle$
c. $[\![$ $[_{VP}$ took ages $]$ $]\!]$: $\langle \sigma, \sigma \rangle$
d. $[\![$ $[_{VP}$ was delicious but took ages $]$ $]\!]$: $\langle \sigma, \sigma \rangle$
e. $[\![$ $[_{S}$ Lunch was delicious but took ages $]$ $]\!]$: σ

Düsseldorf-style frame semantics

Babonnaud et al. 2016 and Kallmeyer and Osswald 2017

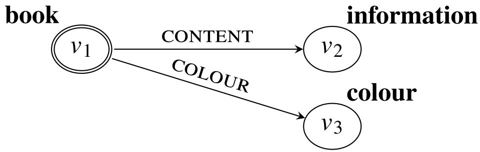
- A frame theory inspired by the work of Barsalou (1992)
- Building on e.g., Petersen 2015; Löbner 2015

(8)



v_1, v_2	Values	E.g., physical books, informational contents, red
CONTENT, COLOUR	Attributes	Functions from values to values
book, **information**, **colour**	Types	Types of values in a type hierarchy

Frames and polysemy

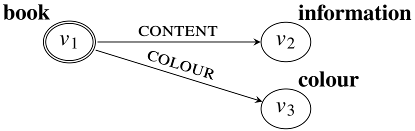
(8) 

- For *book*, the attribute CONTENT, links the physical book to the contents (as the *formal* meaning component of the Generative Lexicon).
- Modifiers that e.g. add an attributes to the v_1 node (or fill in a value for v_3) modify physical books
- Modifiers that add an attributes to the v_2 node modify physical books
- Assumes that the core meaning of *book* is 'physical book'

Polysemy or coercion?

Yesterday:

- Still an open question: Can we treat polysemy as (systematised) coercion?

(8) 

The central node is the physical book

- What about contexts that describe only informational books (allowing e.g. multi-volume books)?
- Shifting the central node?
- Something like: Frames as structures to constrain systematic coercions (constrained by what counts as the formal meaning component)

Denying a different premise?

(a) non-ambiguous nouns (including polysemous nouns) denote functions e.g., from worlds/situations to sets of entities, and

(b) if informational entities, eventualities and physical entities etc. are of a different type, then

(c) there is no function expressible in the simply-typed λ-calculus that can characterise a set of entities that are, say physical and/or informational

Does the frame theoretic approach amount to denying (a)?

- There is a different means of composition: e.g. syntax (L-Tag) driven unification
- But this does not seem crucial to a frame-based account
- Rather it is the assumption of a richer structure (frames) that is doing the work

An aside on types and sorts

Traditionally semantic types perform two roles

- avoiding paradoxes (e.g., Curry's paradox)
- marking conceptual distinctions between entities

Curry's paradox (Curry, 1942; Löb, 1945)

Suppose p is the proposition $p \rightarrow q$ (that q is true if p is true)

If p is false, then $p \rightarrow q$ is false, and so p is true (a contradiction).

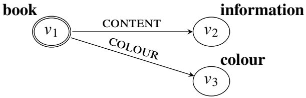
Therefore p is true and so q is true.

But that means we can prove the truth of any formula that we substitute for q, even those that are false.

But these roles can be separated e.g., Kohlhase 1992, 1994

- **Types** to avoid paradoxes
- **Sorts** to mark conceptual distinctions between entities

Frame semantics as a monotyped theory?

(8) 

Types and Attributes

- Semantic types as we know them characterise ATTRIBUTES
- For some type σ, every attribute is of type $\langle \sigma, \sigma \rangle$
- Frames are structures of attributes

Types or Sorts?

- E.g., Petersen 2015 refers to **book**, **information** etc. as types in a type hierarchy
- However, these are better thought of as *sorts*
  - They stand in containment relations in the hierarchy e.g., **book** $\sqsubset$ **physical**
  - But they are not input into type constructors
- So, arguably, this is a mono-typed, multi-sorted semantics, with extra structures (frames)

Summary: The same type response

Our formalised version of Chomsky's argument

(a) non-ambiguous nouns (including polysemous nouns) denote functions e.g., from worlds/situations to sets of entities, and

(b) if informational entities, eventualities and physical entities etc. are of a different type, then

(c) there is no function expressible in the simply-typed λ-calculus that can characterise a set of entities that are, say physical and/or informational/eventualities

We've seen a couple of ways in which we can deny the antecedent in (b)

- Keeping the system of simple types
  - Minimally adding some mapping operations and distinguishing between sorts of type e (e.g., eventualities, physical entities and informational entities)
- Opting for a mono-typed semantics with sorts

Outstanding issues

How many types should we collapse?

- Kinds, degrees, roles, tropes, ...
- Can we replicate the explanatory work done by these types (and complex types formed with them)?
  - E.g., how can we characterise degrees, scales and scale structures within a mono-typed theory?
- For non-monotyped approaches, how equivalence relations do we need to define?

Simple Type Theory

(9) **Types.**
From a non-empty set **BasTyp** of basic types, the set **Typ** of types is the smallest set such that:

a. **BasTyp** $\subseteq$ **Typ**

b. $\langle \sigma, \tau \rangle \in$ **Typ** if $\sigma, \tau \in$ **Typ** (functional type constructor)

	BasTyp	Type constructors
Montague	$\{e, t\}$	(9b) and $\langle s, \sigma \rangle \in$ **Typ** if $\sigma \in$ **Typ**
Gallin	$\{e, t, s\}$	(9b)
Degree semantics	$\{e, t, s, d\}$	(9b)
Neo-Davidsonian	$\{e, t, s, v\}$	(9b)

So two possible ways to amend (traditional) simple type theory

- Adjust **BasTyp**
- Add type constructors

Types or type constructors?

(2) From a non-empty set **BasTyp** of basic types, the set **Typ** of types is the smallest set such that:

a. **BasTyp** $\subseteq$ **Typ**

b. $\langle\sigma,\tau\rangle \in$ **Typ** if $\sigma,\tau \in$ **Typ** (functional type constructor)

Formal semanticists like adding basic types to **BasType**

- Degrees, Eventualities, Roles, Concepts, Tropes, ...

But adding a type constructor is an alternative possibility

- Some examples:
  - Product types e.g., (Gotham, 2014; Sutton and Filip, 2020)
  - Dot types e.g, (Asher and Pustejovsky, 2006)

No in-principle reason not to go for type constructors

- Common in programming languages
  - tuples, lists, dataframes etc.

Two strategies

(a) non-ambiguous nouns denote functions e.g., from worlds/situations to sets of entities, and

(b) if informational entities, eventualities and physical entities etc. are of a different type, then

(c) there is no function expressible in the simply-typed λ-calculus that can characterise a set of entities that are, say physical and/or informational/eventualities

Add at least one type constructor, e.g., dot types

- (c) is true, but harmless
- Keep a simply typed semantics, add at least one type constructor

Rich type theories

- Polysemy is one of many phenomena that indicates the need for more structure in semantics
- Richly typed semantics adds this structure
- Move from a system of simple types to a system of rich types

Dot types

Background

- Original proposal: Pustejovsky 1994, 1995
- Developed into Type Compositional Logic (TCL, Asher and Pustejovsky 2006; Asher 2011)
    - More type constructors and more basic types

Philosophical grounding

- Polysemous expressions refer to entities that have different *aspects*
- E.g., *lunch* refers to something that has a food aspect and an eating-event aspect
- Modifiers like *delicious* draw on the food aspect
- Modifiers like *half-hour* draw on the event aspect

Formalising dot types

Minimally: An additional type constructor

- Construct dot types from any two other types

(10) From a non-empty set **BasTyp** of basic types, the set **Typ** of types is the smallest set such that:

a. **BasTyp** $\subseteq$ **Typ**

b. $\langle \sigma, \tau \rangle \in$ **Typ** if $\sigma, \tau \in$ **Typ** (functional type constructor)

c. $\sigma \bullet \tau \in$ **Typ** if $\sigma, \tau \in$ **Typ** (dot type constructor)

- For types *p* ($phys$) and *v* (ev) for physical entities and eventualities. . .
- . . . *lunch* denotes entities of type $p \bullet v$
    - entities that have a physical entity aspect and an eventuality aspect

Dot types and modification

Example: *book*

- where *p* is the type for physical object and *i* is the type for informational entity
- ⟦book⟧ ↦ a property of entities, namely books, that have both a physical and informational aspect:

(11) $book \mapsto \lambda w.\lambda x_{:p \bullet i}.BOOK_w(x)$

Elaboration functions (simplified)

- Intuitive idea: to elaborate on/pick out an aspect of an object

(12) $[\![\text{lunch was delicious}]\!] =$
$\lambda w.\exists x_{:p}\exists v_{:v\bullet p}[LUNCH(v) \wedge O\text{-}Elab(x,v) \wedge DELICIOUS(x)]$

- The full system of TCL uses type presuppositions and subtyping relations
  - Beyond our scope

Note on dot-type based responses

Only a simplified picture

- The options for implementing a semantics with dot types are wide:

1. A richer, but nonetheless simple type theory
2. Implementation in category theory (Asher, 2011)
3. Richly typed approaches with dot types (Chatzikyriakidis and Luo, 2015)

Take home message

- It is possible, to model polysemy with a semantics based upon a conservatively extended simple type theory
- End of today: Some reasons for opting for a richer theory of types

A puzzle about the denotations of dot-type expressions

Question:

- Suppose a, a lunch, is of type $v \bullet p$ (event dot physical entity)
- What is a? An object? If so, what sort?

Complex Objects?

- E.g. Asher and Pustejovsky (2006) deny this

Regular objects?

- Okay, but in what sense are, say lunches, regular objects?

The role of types in simply typed semantics

- Types are metalanguage descriptions of categories of expressions

(13) If $\phi \in ME_t$ and u is in Var_a, then $[\![\exists u\phi]\!]^{M,g} = 1$ iff for some e in D_a, $[\![\exists u\phi]\!]^{M,g^e_u} = 1$ (Dowty et al., 1981, p. 92)

- Types feature in the metalanguage as subscripts on sets
- We cannot refer to types directly in the object language
- Hunch: If types reflect our basic ontological categories, why can we not refer to them within the object language of our semantic theory?

From simple to rich type theory

Background

- Seminal work by Ranta (1994)
- Implementing a NL semantics based on Martin-Löf 1984
- Often, but not always more proof theoretic

Move 1: Let types feature as part of the object language

- Simply Typed Semantics: Construct arbitrarily complex expressions of some type which are then interpreted (e.g. in a model)
- Richly typed semantics: Construct types themselves of arbitrary complexity
  - Types have witnesses (things of that type)
  - But are individuated also in terms of their structure (fine-grained intensionality)

Example

Simple types: Expressions of some type

- *alex* : *e*, $\lambda x.Mother\_of_w(x)$: $\langle e, e \rangle$
- e.g. $Mother\_of_w(alex)^{M,g} = billie : e$
- There is a sense in which the interpretation of the formulas depend, respectively, on the interpretations of *alex* and *billie*

Rich types: Types with a structure

- In richly typed semantics, we have structured types and entities can be of some type or not
- $Mother\_of(alex), Mother\_of(billie)$ are types
- $Mother\_of$ is a type constructor
  - It maps individuals of some type into a type of individuals (that of being a mother)
- e.g. *billie* : $Mother\_of(alex)$
  - *billie* witnesses the type of being Alex's mother
- The type $Mother\_of(billie)$ **depends** on the value *billie*

From simple to rich type theory

Move 1: Let types feature as part of the object language

Move 2: Treat propositions as types

- Curry-Howard Correspondence (Curry and Feys, 1958; Howard, 1980)

Simple Type Theory (STT)	Rich Type Theory (RTT)
Sets of worlds	Types
Flat	Structured
Individuated by set membership	Individuated by witness set *and* structure

- A means of encoding hyperintensionality
  - The types $2+2=4$ and $5-3=2$ have the same witnesses (situations, worlds etc.)
  - But can be individuated in terms of structure (and the manner of construction)

Treating Polysemy in RTT semantics

Today: Two examples within Type Theory with Records (TTR) e.g. Cooper 2012, 2023

- Setting Modern Type Theory (MTT, e.g., Luo 2010; Chatzikyriakidis and Luo 2020)
- Some mention of MTT tomorrow for polysemy and quantification

Example 1: Pustejovskian 'aspects' based analysis without dot types

- Cooper 2011

Example 2: Polysemy without aspects

- Sutton 2022

Very short introduction to TTR

Record Types

(14) $\begin{bmatrix} \mathrm{x} & : & Ind \\ \mathrm{c}_1 & : & cat(x) \end{bmatrix}$

- *There is a cat*
- Pred logic analogue: $\lambda w.\exists x.cat_w(x)$

- Propositions in TTR (situation types)
- Witnesses are records (situations)
- Labels x, c_1 are like discourse referents
- *Ind* is a basic type
- $cat(x)$ is a type constructor: constructs a type given a value for the label x

Records

- Situations

(15) $\begin{bmatrix} \mathrm{x} & = & felix \\ \mathrm{c}_1 & = & s_1 \end{bmatrix}$

- (15) : (14) iff
  - *felix* : *Ind*
  - $s_1 : cat(felix)$

Non-Polysemous Common Nouns in TTR

CNs denote Properties

- Not functions from worlds to sets of entities
- Functions from records (situations) to a record type (a proposition)

(16) $\mathit{cat} \mapsto \lambda r : \left[\begin{array}{lll} \mathrm{x} & : & \mathit{Ind} \end{array}\right] . \left[\begin{array}{lll} \mathrm{c_{cat}} & : & \mathrm{cat}(r.\mathrm{x}) \end{array}\right]$

- Functions from records of some type: $\lambda r : [\mathrm{x} : \mathit{Ind}]$
  - I.e., situations that contain some individual
- to a proposition
  - I.e., the type of situations in which the entity labelled x is a cat

Simplified example

- We can treat proper names as GQs: functions from a property to the proposition that some individual has that property

(17) $\mathit{Felix} \mapsto \lambda P : \mathit{Ppty}.P([\mathrm{x} = \mathit{felix}])$

(18) $\mathit{cat} \mapsto \lambda r : \left[\ \mathrm{x} : \mathit{Ind}\ \right].\left[\ \mathrm{c}_{\mathrm{cat}} : \mathrm{cat}(r.\mathrm{x})\ \right] \quad : \mathit{Ppty}$

(19) $\mathit{Felix\ is\ a\ cat} \mapsto \begin{bmatrix} \mathit{felix} & : & \mathit{Ind} \\ \mathrm{c}_{\mathrm{cat}} & : & \mathrm{cat}(\mathit{felix}) \end{bmatrix}$

Important theoretical point:

- CNs do not (directly) denote as properties of individuals
- CNs denote properties of situations that contain individuals

Aspects modelled with type constructors (Cooper, 2011)

- No dot type constructor needed to represent aspects
- lunch_ev_fd($r.x, e, f$) constructs a type given values for r.x, e, and f
- I.e. the type of situation in which the entity labelled by x in r has two aspects:
  - that of being f of type $food$
  - that of being e of type $event$

(20) *lunch*

$$\mapsto \lambda r : \left[\ \text{x} \ : \ Ind\ \right] . \begin{bmatrix} \text{f} & : & food \\ \text{e} & : & event \\ \text{c}_{\text{lunch}} & : & \text{lunch\_ev\_fd}(r.x, e, f) \end{bmatrix}$$

In words:

- A property of situations that contain some individual
  - Individual understood rather broadly
- Returns the proposition that there is some food and some event that are aspects of the individual contained in the relevant situation

Comments on Cooper's analysis

Advantages:

- No special type constructor to model polysemy
- Predicates are anyway treated as type constructors, and aspects are a special kind of ternary relation

Puzzle:

- As with the Asher-Pustejovsky approach
  - We can't really say what *the* individual that is the lunch is

Alternative:

- As with the Asher-Pustejovsky approach
  - We can't really say what *the* individual that is the lunch is
- We could treat polysemous nouns as denoting less mysterious entities

Multi-participant situations (Sutton, 2022)

Polysemous nouns denote situations that contain multiple participants

- polysemous Ns constrain situations to witness at least two entities
- e.g., *lunch*: to witness at least some event and some physical entity
- the resulting record type constrains the event to be a lunch eating event and the individual to be the food
- Additionally neo-Davidsonian inspired thematic role relations

(21) $$lunch \mapsto \lambda r : \begin{bmatrix} \mathrm{x} & : & Phys \\ \mathrm{e} & : & Ev \end{bmatrix} . \begin{bmatrix} \mathrm{c_{food}} & : & \mathrm{food}(r.\mathrm{x}) \\ \mathrm{c_{eat}} & : & \mathrm{eat\_lunch}(r.\mathrm{e}) \\ \mathrm{c_{pat}} & : & \mathrm{patient}(r.\mathrm{x}, r.\mathrm{e}) \end{bmatrix}$$

Features of the multi-participant analysis

(21) $\textit{lunch} \mapsto \lambda r : \begin{bmatrix} \text{x} & : & \textit{Phys} \\ \text{e} & : & \textit{Ev} \end{bmatrix} . \begin{bmatrix} \text{c}_{\text{food}} & : & \text{food}(r.\text{x}) \\ \text{c}_{\text{eat}} & : & \text{eat\_lunch}(r.\text{e}) \\ \text{c}_{\text{pat}} & : & \text{patient}(r.\text{x}, r.\text{e}) \end{bmatrix}$

Nothing to see here

- If CNs only indirectly denote entities, via denoting situations then we only have pretty vanilla entities here
  - situations, physical entities, eventualities

Relations like Patient explain restrictions on copredication

(22) The statement in the envelope is inaccurate. (Phys, Inf)
(23) ?The statement in the envelope lasted half an hour. (Phys, Ev)
(24) The inaccurate statement lasted half an hour. (Inf, Ev)

- (Phys, Ev) is bad because there is no *contents* relation between them
- See also Ortega-Andrés and Vicente (2019) (realization relations)

Copredication can improve with context support

Yesterday: copredication can improve with sufficient contextual support:

(25) a. Context: The police took verbal statements from witnesses, but all were simultaneously transcribed. The shorter transcriptions are on the desk.

b. Every statement that took less than 5 minutes is on the desk

- What does *transcribe* contribute to the context?
- Plausibly: a relation between the stating eventuality and a physical entity (the transcription)
- I.e., exactly what was missing, thereby licensing copredication

Summary: Rich type theoretical approaches

Advantages

- No special machinery that is bespoke to polysemy
  - Cf. richer simply typed approaches
- Sufficient structure to be able to distinguish between senses
  - But also to relate them (copredication)

Cost

- Some major departures from semantics in the Montague-Lewis tradition
  - A different account of propositions
  - Something like indirect denotation of entities for proper nouns etc.

Same type vs. Richer type responses

What does enriching the type theory do?

- At base level: It introduces *structure*
- E.g., structured types or more structured object language expressions

Reminder: Düsseldorf Frames

- These also introduce structure
- Simply typed attributes are related in a frame structure

What does simplifying the type theory do?

- At base level: It *destroys structure*
- E.g., eventualities and physical objects are treated a alike from the perspective of what the semantics can 'see'
  - Assuaged somewhat by sorting the domain?

The effects of polysemy on semantic theory

Polysemy and copredication are challenging, given traditional assumptions

- Chomsky's argument, distinctions between basic types etc.

This seems to force a choice:

- Impoverish: eradicate at least some type distinctions
- Enrich: Introduce finer grained types, but most importantly, new ways of putting types together

Open questions:

- Will eliminating structure to model polysemy prevent us from modelling other phenomena that may rely on that structure?

Richness and thinness

- Options for treating polysemy (Hogeweg and Vicente, 2020):
  - Richer lexicon (add structure)
  - Thinner (remove structure)
  - “very thin view and a very rich view may turn out to be indistinguishable in the long run”

A puzzle about sorts for monotyped semantics

(Nb. Does not apply to frame semantics)

The use of sorts for selectional restrictions

- Non-polysemous case:
- $\lambda w.\lambda x_{:\sigma}.Cat_w(x)$
- The relevant entities of type σ are of the sort e.g., *Physical*
- Explains e.g. *??a five minute cat*

A re-emergence of the problem?

- $\lambda w.\lambda x_{:\sigma}.Lunch_w(x)$
- What sorts are the relevant entities of type σ?
- $Phys + Ev$ for some sort combinator $+$?
- Sorts start to look a lot like types
- If we object to extra type constructors, shouldn't we object to extra sort constructors?

Tomorrow

A new puzzle:

(26) Three informative books are on the shelf.

- Seems to require that the books are both physically distinct and informationally distinct
- But, this can be overridden by context
- Question: What (if anything) do modifiers contribute semantically towards individuation criteria?

Thank you!

References I

Asher, N. (2011). *Lexical Meaning in Context: A Web of Words*. Cambridge University Press.

Asher, N. and J. Pustejovsky (2006). A type composition logic for generative lexicon. *Journal of Cognitive Science*, 1–38. reprinted in *Advances in Generative Lexicon Theory*, Kluwer Academic Publishers, 2010. doi:https://doi.org/10.1007/978-94-007-5189-7_310.1007/978-94-007-5189-7_3.

Babonnaud, W., L. Kallmeyer, and R. Osswald (2016). Polysemy and coercion: A frame-based approach using LTAG and Hybrid Logic. In *Logical Aspects of Computational Linguistics, 9th International Conference, Lecture Notes in Computer Science 10054*, pp. 18–33. Berlin: Springer.

Barsalou, L. W. (1992). Frames, concepts, and conceptual fields. In E. Kittay and A. Lehrer (Eds.), *Frames, fields, and contrasts: New essays in semantic and lexical organization*, pp. 21–74. Erlbaum.

Chatzikyriakidis, S. and Z. Luo (2015, July). Individuation criteria, dot-types and copredication: A view from modern type theories. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, Chicago, USA, pp. 39–50. Association for Computational Linguistics.

Chatzikyriakidis, S. and Z. Luo (2020). *Formal Semantics in Modern Type Theories*. London, UK Hoboken, NJ: ISTE, Ltd. Wiley.

References II

Chomsky, N. (2000). *New horizons in the study of language and mind*. Cambridge University Press.

Collins, J. (2017). The copredication argument. *Inquiry* (7), 675–702.

Cooper, R. (2011). Copredication, quantification and frames. In S. Pogodalla and J.-P. Prost (Eds.), *Logical Aspects of Computational Linguistics. Number 6736 in Lecture Notes in Computer Science*, pp. 64–79. Springer.

Cooper, R. (2012). Type Theory and Semantics in Flux. In R. Kempson, T. Fernando, and N. Asher (Eds.), *Philosophy of Linguistics, Handbook of the Philosophy of Science*, pp. 271–323. Elsevier.

Cooper, R. (2023). *From Perception to Communication: a Theory of Types for Action and Meaning*. Oxford University Press.

Curry, H. B. (1942). The Inconsistency of Certain Formal Logics. *Journal of Symbolic Logic* (3), 115–117.

Curry, H. B. and R. Feys (1958). *Combinatory Logic Volume 1*. North-Holland Publishing.

Dowty, D. R., R. E. Wall, and S. Peters (1981). *Introduction to Montague semantics*. Dordrecht: Klewer.

Gotham, M. (2014). *Copredication, Quantification and Individuation*. Ph. D. thesis, University College London.

References III

Gotham, M. (2017, 08). Composing Criteria of Individuation in Copredication. *Journal of Semantics 34*(2), 333–371.

Gotham, M. (2021, 12). Property Inheritance, Deferred Reference and Copredication. *Journal of Semantics 39*(1), 87–116.

Hogeweg, L. and A. Vicente (2020). On the nature of the lexicon: The status of rich lexical meanings. *Journal of Linguistics 56*(4), 865–891.

Howard, W. (1980). The formulae-as-types notion of construction. In R. Hindley and J. P. Seldin (Eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 479–490. Academic Press.

Kallmeyer, L. and R. Osswald (2017). Modeling quantification with polysemous nouns. In *IWCS 2017 ? 12th International Conference on Computational Semantics ? Short papers*.

Kohlhase, M. (1992). Unification in order-sorted type theory. In A. Voronkov (Ed.), *Proceedings of the International Conference on Logic Programming and Automated Reasoning LPAR'92 (LNAI 624)*, pp. 421–432. Springer Verlag.

Kohlhase, M. (1994). *A Mechanization of Sorted Higher-Order Logic Based on the Resolution Principle*. Ph. D. thesis, Universität des Saarlandes, Germany. PhD dissertation.

Liebesman, D. and O. Magidor (2017). Copredication and property inheritance. *Philosophical Issues 27*, 131–166.

References IV

Liebesman, D. and O. Magidor (2019). Copredication, counting, and criteria of individuation: A response to gotham. *Journal of Semantics 36*, 549–561.

Liefke, K. (2014). *A single-type semantics for natural language*. Ph. D. thesis, Doctoral dissertation, Center for Logic and Philosophy of Science, Tilburg University.

Liefke, K. and M. Werning (2018, 08). Evidence for Single-Type Semantics?An Alternative To e/t-Based Dual-Type Semantics. *Journal of Semantics 35*(4), 639–685.

Löb, M. (1945). Solution of a Problem of Leon Henkin. *Journal of Symbolic Logic* (2), 115–118.

Löbner, S. (2015). Functional concepts and frames. In T. Gamerschlag, D. Gerland, R. Osswald, and W. Petersen (Eds.), *Meaning, Frames, and Conceptual Representation*, pp. 15–42. Düsseldorf University Press.

Luo, Z. (2010). Type-theoretical semantics with coercive subtyping. In *Semantics and linguistic theory*, Volume 20, pp. 38–56.

Martin-Löf, P. (1984). *Intuitionistic Type Theory*. Naples: Bibliopolis.

Ortega-Andrés, M. and A. Vicente (2019). Polysemy and co-predication. *Glossa* (1), 1–23.

References V

Partee, B. (2007). ‘type theory and natural language: do we need two basic types?’. 10th Meeting of the Seminar: Mathematical Methods Applied to Linguistics. Moscow State University. Moscow.

Petersen, W. (2015). Representation of concepts as frames. In T. Gamerschlag, D. Gerland, R. Osswald, and W. Petersen (Eds.), *Meaning, Frames, and Conceptual Representation*, pp. 43–67. Düsseldorf University Press.

Pietroski, P. (2003). The Character of Natural Language Semantics. In A. Barber (Ed.), *Epistemology of Language*, pp. 217–256. Oxford University Press.

Pustejovsky, J. (1994). Semantic typing and degrees of polymorphism. In C. Martin-Vide (Ed.), *Current issues in mathematical linguistics*, pp. 221–238. Elsevier.

Pustejovsky, J. (1995). *The Generative Lexicon*. MIT Press.

Ranta, A. (1994). *Type-Theoretical Grammar*. Oxford: Clarendon Press.

Sutton, P. R. (2022). Restrictions on copredication: a situation theoretic approach. *Semantics and Linguistic Theory (SALT) 32*, 335–355.

Sutton, P. R. and H. Filip (2020). Informational object nouns and the mass/count distinction. *Proceedings of Sinn und Bedeutung 24 2*, 319–335.