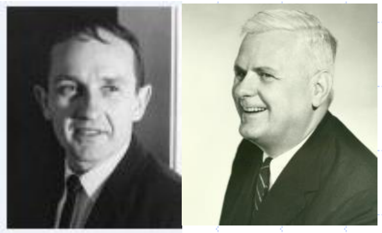
From Montague Semantics to MTT-Semantics: A Meaningful Comparison

Stergios Chatzikyriakidis and Zhaohui Luo

(ESSLLI 2019: Lecture 1.1 by ZL)

Type-Theoretical Semantics

- Montague Semantics
  - R. Montague (1930–1971)
  - Dominating in linguistic semantics since 1970s
  - Set-theoretic with Church's simple type theory as intermediate lang



- MTT-semantics: formal semantics in modern type theories
  - Examples of MTTs:
    - Predicative: MLTT (Martin-Löf 73,84); MLTT$_h$ (h-logic in HoTT)
    - Impredicative: pCIC (Coq) and UTT (Luo 1994)
  - Ranta (1994): formal semantics in Martin-Löf's type theory
  - Recent development on MTT-semantics

    ➔ full-scale alternative to Montague semantics

- Recent development on rich typing in NL semantics
  - Asher, Bekki, Cooper, Grudzińska, Retoré, …
    - S. Chatzikyriakidis and Z. Luo (eds.) Modern Perspectives in Type Theoretical Sem. Springer, 2017. (Collection on rich typing & …)
  - MTT-semantics is one of these developments.
    - Z. Luo. Formal Semantics in Modern Type Theories with Coercive Subtyping. Linguistics and Philosophy, 35(6). 2012.
    - S. Chatzikyriakidis and Z. Luo. Formal Semantics in Modern Type Theories. Wiley/ISTE. (Monograph on MTT-semantics, to appear)

Simple Type Theory v.s. Modern Type Theories

- Compare them from two aspects
  - Type structure
  - Logic
- Type structures
  - Church's simple type theory (in Montague's semantics)
    - Base types: e, t (plus s …)
    - Function types: $\alpha \rightarrow \beta$ (e.g., $e \rightarrow e \rightarrow t$)
  - Modern type theories (next page: example type constructor)
    - Dependent types (Π-types, Σ-types, …)
    - Inductive types (Nat, Fin(n), …)
    - Universes – types of types (logical, linguistic, …)

  Note: MTTs are defined by proof-theoretic rules.

Πx:A.B(x) – Example

- Πx:Human.Child(x)
  - Type of functions mapping h to Child(h), type of h's children.
- A→Prop (Πx:A.Prop)
  - Type of predicates over A
- Π-polymorphism
  - small : ΠA:CN.(A→Prop)
  - small(Elephant) : Elephant→Prop
  - small(Mouse) : Mouse→Prop
  - small(Table) : Table→Prop
  - … …

$$\frac{\Gamma \vdash A\ \mathit{type} \quad \Gamma,\ x{:}A \vdash B\ \mathit{type}}{\Gamma \vdash \Pi x{:}A.B\ \mathit{type}}$$

$$\frac{\Gamma,\ x{:}A \vdash b : B}{\Gamma \vdash \lambda x{:}A.b : \Pi x{:}A.B}$$

$$\frac{\Gamma \vdash f : \Pi x{:}A.B \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : [a/x]B}$$

$$\frac{\Gamma,\ x{:}A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x{:}A.b)(a) = [a/x]b : [a/x]B}$$

Logic

- Simple type theory
  - Higher-order logic (traditional, truth-based)
  - Formulae in t, predicates in e→t, …
- Modern type theories
  - Propositions as types (with proof terms)
  - Formula A is provable/true
    ⇔ there is a proof of A
    ⇔ there is an object p : A

$$A = \forall P : N \to Prop\, \forall x : N.\, P(x) \Rightarrow P(x)$$

$$p = \lambda P : N \to Prop\ \lambda x : N\ \lambda y : P(x).\, y$$

$$p : A$$

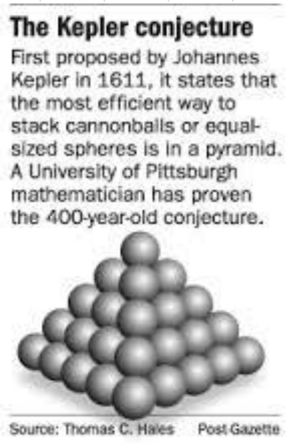
  - So, <u>decidability</u> of type-checking is essential for the PaT logic (and other things like implementation of proof assistants).
    - Given p and A, it should be decidable whether p : A.
    - Note: this is different from A's truth, which is in general undecidable.

History of MTTs and MTT-semantics

- Modern (Dependent) Type Theories
  - Russell (1903,1925) & Ramsay (1926) on simple type theory (for foundation of classical math)
  - Martin-Lof (1973,1984) on predicative type theory (for constructive math; also cf., Feferman, Friedman, Myhill)
  - Impredicative TTs: CC (Coquand 1987), UTT (Luo 1994), …
  - Applications of proof assistants (see next page)
- MTT-semantics
  - Sundholm (1986), Ranta (1994), Luo (1998,2009), Retore (2013), Chatzikyriakidis & Luo (2013,2016), Bekki (2014), Grudzinska (2017) …
  - Recent development on MTT-semantics (last decade or so), some of which to be reported in these lectures.

An episode: MTT-based technology and applications

- Proof technology based on type theories
  - Proof assistants – computer systems for formal reasoning
    - MTT-based: ALF/Agda, Coq, Lego, NuPRL, Plastic, …
    - HOL-based: Isabelle, HOL, …
- Applications of proof assistants
  - Math: formalisation of mathematics – eg,
    - 4-colour theorem (on map colouring) in Coq
    - Kepler conjecture (on sphere packing) in Isabelle/HOL
  - Computer Science:
    - program verification and advanced programming
  - Computational Linguistics
    - E.g., MTT-sem based NL reasoning in Coq (Chatzikyriakidis & Luo 2014)


The Kepler conjecture
First proposed by Johannes Kepler in 1611, it states that the most efficient way to stack cannonballs or equal-sized spheres is in a pyramid. A University of Pittsburgh mathematician has proven the 400-year-old conjecture.

Source: Thomas C. Hales Post-Gazette

Course Plan (Lectures I-V)

I. Introduction
- Course overview (ZL)
- Intro to Montague semantics and MTT-semantics (SC)

II. Features of MTT-sem and differences with MG (ZL)
- Rich type structure
- Powerful tools for semantic constructions
- Both model-theoretic and proof-theoretic
- Others (eg, judgemental interpretation and identity criteria)

III. Modification (SC)
- Adjectival modifications: basic and advanced
- Adverbial modification

IV. Copredication (SC)

- Dot-types (A•B) for copredication
- Interaction with quantification (case involving counting)

V. Dependent types in event semantics (ZL)

- Dependent types in linguistic semantics in general
- Selection restriction in MTT-event semantics
- Dependent event types

- Material to distribute (on course web site)
  - Lecture slides by SC and ZL
  - Course proposal for this course (good summary)
- Coming soon:
  - S. Chatzikyriakidis and Z. Luo. Formal Semantics in Modern Type Theories. Wiley/ISTE. (To finish 2019)
- Some papers and material can be found at
  - http://www.cs.rhul.ac.uk/home/zhaohui/lexsem.html

From Montague Semantics to MTT-semantics: A Meaningful Comparison

Lecture 1.2: Introduction (given by SC)

Stergios Chatzikyriakidis and Zhaohui Luo

CLASP centre for linguistic theory and studies in probability

August 12, 2019

Introducing Montague (briefly) and MTTs

- What are the main differences between MG and MTTs
  - To answer this, at least partly, we present a brief overview of the respective systems, Montague's IL and Luo's MTT

The Syntax of IL: Types

- Basic Types
  1. t (truth-values)
  2. e (individual entities)
- Exponential Types

  If a, b are types, then $a \rightarrow b$ is a type.

- Intensional Types

  If a is a type, then $s \rightarrow a$ is a type. (the type of the intension of a)

The Syntax of IL: Types

- The basic types t, e correspond to the sentences (denotations of propositions) and the individual constants (the denotations of proper names), respectively, in IL.
- An exponential type $a \rightarrow b$ is the functional type whose elements are functions from a to b.
- intensional type $s \rightarrow a$ are functions from indices (possible worlds) to denotations of the type a.

A note on the typing system

- This is not Church's original type system
  - There are no intensional function types in Church's system
  - From a TT point of view Montague's system is a little bit peculiar: two basic types as in Church (e, t correspond to ι, o in Church)

A note on the typing system

- This is not Church's original type system
  - There are no intensional function types in Church's system
  - From a TT point of view Montague's system is a little bit peculiar: two basic types as in Church (e, t correspond to ι, o in Church)
    - But: an extra type s (type of world,time pairs) which is only used in functions (does not exist as a basic type)
    - Later reformulation by Gallin [1975] introduces s as a basic type (and in general it is a neater system (see Muskens, 1996 for a discussion and justification)

The Set ME_a of Well Formed Expressions of Type a

1. Every variable of type a is in ME_a.
2. Every constant of type a is in ME_a.
3. If $\alpha \in ME_a$ and u is a variable in ME_b, then $\lambda u \alpha \in ME_{a \to b}$.
4. If $\alpha \in ME_{a \to b}$ and $\beta \in ME_a$, then $\alpha(\beta) \in ME_b$.
5. If $\alpha, \beta \in ME_a$, then $\alpha = \beta \in ME_t$.

The Set ME_a of Well Formed Expressions of Type a

- If $\phi, \psi \in ME_t$, then so are
  1. $\neg\phi$
  2. $\phi \vee \psi$
  3. $\phi \wedge \psi$
  4. $\phi \rightarrow \psi$
  5. $\phi \leftrightarrow \psi$.
- If $\phi \in ME_t$ and u is a variable in ME_a, then $\forall u\phi$ and $\exists u\phi \in ME_t$.
- If $\phi \in ME_t$, then $\in ME_t$.
- If $\alpha \in ME_a$, then ${}^\wedge\alpha \in ME_{s\rightarrow a}$
- If $\alpha \in ME_{s\rightarrow a}$, then ${}^\vee\alpha \in ME_a$

The Semantics of IL: Models

- A model $M = \{A, W, F\}$ for IL is an ordered triple such that A and W are non empty sets (of entities and possible worlds, respectively).
- For each type a of IL, F is a function from the non-logical constants of ME_a to interpretations of these constants.
- For each constant $c \in ME_a$ $F(c)$ is a function $f_c{:}W \rightarrow D_a$, where D_a is the domain of possible denotations for expressions of type a.

The Set of Domains of Type *a*

Each non-logical constant of type a has a domain of possible denotations D_a.

The set of these domains is defined recursively as follows.

- 1. $D_e = A$
  2. $D_t = 0, 1$
  3. $D_{a \to b} = {D_b}^{D_a}$
  4. $D_{s \to a} = {D_a}^{W}$

The Interpretation Function $[\![\alpha]\!]^{\mathcal{M},w,g}$

- Let g be an assignment function such that for any variable $x \in a$ of IL, $g(x) \in D_a$.
- For a model M of IL, assume $w \in W$.
- The expression $[\![\alpha]\!]^{\mathcal{M},w,g}$ denotes the evaluation of α in a model M, relative to a world w and an assignment function g.

The Semantic Rules of IL

1. If α is a non-logical constant, then $[\![\alpha\ ]\!]^{\mathcal{M},w,g} =$ (F(α))(w).
2. If α is a variable, then $[\![\alpha]\!]^{\mathcal{M},w,g} =$ g(α).
3. If $\alpha \in ME_a$ and u is a variable of type b, then $[\![\lambda u.\alpha]\!]^{\mathcal{M},w,g} =$ $h{:}D_b \to D_a$ such that for any $d \in D_b, h(d) = [\![\alpha]\!]^{\mathcal{M},w,g[d/u]}$.
4. If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_{<b>}$, then $[\![\alpha(\beta)]\!]^{\mathcal{M},w,g} =$ $[\![\alpha]\!]^{\mathcal{M},w,g}([\![\beta]\!]^{\mathcal{M},w,g})$.
5. If $\alpha, \beta \in ME_a$, then $[\![\alpha = \beta]\!]^{\mathcal{M},w,g} = 1$ iff $[\![\alpha]\!]^{\mathcal{M},w,g} =$ $[\![\beta]\!]^{\mathcal{M},w,g}$.

The Semantic Rules of IL

1. If $\phi \in ME_t$, then $[\![\neg\phi]\!]^{\mathcal{M},w,g} = 1$ iff $[\![\phi]\!]^{\mathcal{M},w,g} = 0$.
2. If $\phi, \psi \in ME_t$, then $[\![\phi \vee \psi]\!]^{\mathcal{M},w,g} = 1$ iff $[\![\phi]\!]^{\mathcal{M},w,g} = 1$, or $[\![\psi]\!]^{\mathcal{M},w,g} = 1$.
3. If $\phi, \psi \in ME_t$, then $[\![\phi \wedge \psi]\!]^{\mathcal{M},w,g} = 1$ iff $[\![\phi]\!]^{\mathcal{M},w,g}= 1$, and $[\![\psi]\!]^{\mathcal{M},w,g} = 1$.
4. If $\phi, \psi \in ME_t$, then $[\![\phi \rightarrow \psi]\!]^{\mathcal{M},w,g} = 1$ iff $[\![\phi]\!]^{\mathcal{M},w,g} = 0$, or $[\![\psi]\!]^{\mathcal{M},w,g} = 1$.
5. If $\phi, \psi \in ME_t$, then $[\![\phi \leftrightarrow \psi]\!]^{\mathcal{M},w,g} = 1$ iff $[\![\phi]\!]^{\mathcal{M},w,g} = [\![\psi]\!]^{\mathcal{M},w,g}$.

The Semantic Rules of IL

1. If $\phi \in ME_t$ and u is a variable of type b, then $[\![\forall u\phi]\!]^{\mathcal{M},w,g} = 1$ iff for all $d \in D_b$, $[\![\phi]\!]^{\mathcal{M},w,g[d/u]} = 1$.
2. If $\phi \in ME_t$ and u is a variable of type b, then $[\![\exists u\phi]\!]^{\mathcal{M},w,g} = 1$ iff for some $d \in D_b$, $[\![\phi]\!]^{\mathcal{M},w,g[d/u]} = 1$.
3. If $\phi \in ME_t$, then $[\![\neg\phi]\!]^{\mathcal{M},w,g} = 1$ iff for all $w \in W$, $[\![\phi]\!]^{\mathcal{M},w,g}$ $= 1$.
4. If $\alpha \in ME_a$, then $[\![{}^\wedge\alpha]\!]^{\mathcal{M},w,g} = h{:}W \rightarrow D_a$ such that for every $w \in W, h(w) = [\![\alpha]\!]^{\mathcal{M},w,g}$
5. If $\alpha \in MEs \rightarrow a$, then $[\![{}^\vee\alpha]\!]^{\mathcal{M},w,g} = ([\![\alpha]\!]^{\mathcal{M},w,g})$(w).

MG: Summary

1. A higher order logic based on Simple Type Theory (STT)
    1. Very simple (monolithing) typing system in the style of Church (with the addition of type s appearing only in function types)

    1. Set-theoretic interpretation via Henkin models
    2. Further equipped with intensional notions via Kripkean models (intensionality)
        1. Very powerful model-theoretic semantics!

MTTs and linguistic semantics

- Starts with the seminal work by Ranta Ranta [1994] and earlier (e.g. Sundholm Sundholm [1989])

MTTs and linguistic semantics

- Starts with the seminal work by Ranta Ranta [1994] and earlier (e.g. Sundholm Sundholm [1989])
  - Many more after that Boldini [2000], Cooper [2005], Dapoigny and Barlatier [2009], Bekki [2014], Retoré [2013]
  - How they are useful and in what ways they are different from STT?

Basic Types: Rich Typing

- In STT, the domain of individuals is monolithic, i.e. one basic entity type

Basic Types: Rich Typing

- In STT, the domain of individuals is monolithic, i.e. one basic entity type
  - Function types for different types of individuals, e.g. *man*, *human* are not basic types but function types

Basic Types: Rich Typing

- In STT, the domain of individuals is monolithic, i.e. one basic entity type
    - Function types for different types of individuals, e.g. *man*, *human* are not basic types but function types
- In MTTs, no such restriction exists: the universe of entities can be many-sorted

Basic Types: Rich Typing

- In STT, the domain of individuals is monolithic, i.e. one basic entity type
    - Function types for different types of individuals, e.g. *man*, *human* are not basic types but function types
- In MTTs, no such restriction exists: the universe of entities can be many-sorted
    - Arbitrary number of types available, e.g. $man, chair : Type$ (this is the approach by Ranta, Boldini, Luo and colleagues among others)

Basic Types: Rich Typing

- Many-sortedness

Basic Types: Rich Typing

- Many-sortedness
  - Allows us interpret common nouns are not predicates but Types!

Basic Types: Rich Typing

- Many-sortedness
  - Allows us interpret common nouns are not predicates but Types!

    MG $man : e \to t$

    MTTs $man : Type$
  - Hybrid systems: use both predicates and types when needed (Retoré [2013], e.g. *man* as both a predicate and a type)

Basic Types: Rich Typing

- Selectional restrictions as type mismatch: *the ham sandwich talks*

Basic Types: Rich Typing

- Selectional restrictions as type mismatch: *the ham sandwich talks*
    - $talk : human \rightarrow Prop$
    - $the\ ham : ham$ (with $ham : Type$)
    - Functional application not possible!

Subtyping

- Rich selection of types

Subtyping

- Rich selection of types
  - Subtyping mechanism becomes more essential: otherwise the system becomes too rigid
  - Even things like *the man walks* would not be possible with no subtyping mechanism

Subtyping

- Rich selection of types
  - Subtyping mechanism becomes more essential: otherwise the system becomes too rigid
  - Even things like *the man walks* would not be possible with no subtyping mechanism
    - $walk : animal \to Prop$
    - $the\ man : man$ (with $man : Type$)
    - Fine if $man \leq human$

Different Systems of Subtyping

- Classic case: Subsumptive subtyping

$$\frac{a{:}A,\ A \leq B}{a{:}B}$$

- a term of type A can be used in a context where a term of type B is required instead just in case $A \leq B$

Different Systems of Subtyping

- Coercive subtyping (Luo and colleagues)

Different Systems of Subtyping

- Coercive subtyping (Luo and colleagues)
  - Can be seen as an abbreviation mechanism

Different Systems of Subtyping

- Coercive subtyping (Luo and colleagues)
  - Can be seen as an abbreviation mechanism
    - A is a (proper) subtype of B ($A < B$) if there is a unique implicit coercion c from type A to type B
    - An object a of type A can be used in any context $\mathfrak{C}_B[\_]$ that expects an object of type B: $\mathfrak{C}_B[a]$ is legal (well-typed) and equal to $\mathfrak{C}_B[c(a)]$.

Different Systems of Subtyping

- Coercive subtyping (Luo and colleagues)
  - Can be seen as an abbreviation mechanism
    - A is a (proper) subtype of B ($A < B$) if there is a unique implicit coercion c from type A to type B
    - An object a of type A can be used in any context $\mathfrak{C}_B[\_]$ that expects an object of type B: $\mathfrak{C}_B[a]$ is legal (well-typed) and equal to $\mathfrak{C}_B[c(a)]$.
- Metatheoretically more advantageous: canonicity is respected
  - Long story!

Dependent Typing

- STT involves basic types and function types constructed out of the basic types

Dependent Typing

- STT involves basic types and function types constructed out of the basic types
- MTTs offer a range of other more advanced typing structures

Dependent Typing

- STT involves basic types and function types constructed out of the basic types
- MTTs offer a range of other more advanced typing structures
  - Dependent Typing
    - A family of types that may depend on some value

Complex Types and Dependent Typing

- Dependent Types Π and Σ

Complex Types and Dependent Typing

- Dependent Types Π and Σ
  - When A is a type and P is a predicate over A, $\Pi x{:}A.P(x)$ is the dependent function type that stands for the universally quantified proposition $\forall x{:}A.P(x)$
  - Π for polymorphic typing: $\Pi A{:}CN.(A \rightarrow Prop) \rightarrow (A \rightarrow Prop)$

Complex Types and Dependent Typing

- Dependent Types Π and Σ
  - When A is a type and P is a predicate over A, $\Pi x{:}A.P(x)$ is the dependent function type that stands for the universally quantified proposition $\forall x{:}A.P(x)$
  - Π for polymorphic typing: $\Pi A{:}CN.(A \rightarrow Prop) \rightarrow (A \rightarrow Prop)$
  - A is a type and B is an A-indexed family of types, then $\Sigma x{:}A.B(x)$, is a type, consisting of pairs (a, b) such that a is of type A and b is of type $B(a)$.
  - Adjectival modification as involving Σ types Ranta [1994], Chatzikyriakidis and Luo [2017]: $[\![heavy\_book]\!] = \Sigma x : book.heavy(x)$

Intro to MTTs-Universes

- Universes
  - A universe is a collection of (the names of) types into a type (Martin Löf, 1984).

Intro to MTTs-Universes

- Universes
  - A universe is a collection of (the names of) types into a type (Martin Löf, 1984).
  - Universes can help semantic representations. For example, one may use the universe $\mathrm{CN} : \mathit{Type}$ of all common noun interpretations and, for each type A that interprets a common noun, there is a name $\overline{A}$ in CN. For example,

    $$\overline{man} : \mathrm{CN} \quad \text{and} \quad T_{\mathrm{CN}}(\overline{man}) = man.$$

    In practice, we do not distinguish a type in CN and its name by omitting the overlines and the operator T_{CN} by simply writing, for instance, $man : CN$.

Intro to MTTs-Universes

- We can construct more fine-grained universes.

Intro to MTTs-Universes

- We can construct more fine-grained universes.
  - For example, one may want to have a type that extends over the universe including type *human* and its subtypes

Intro to MTTs-Universes

- We can construct more fine-grained universes.
  - For example, one may want to have a type that extends over the universe including type *human* and its subtypes
    - Such a universe can be constructed, by giving its introduction rules

$$\frac{}{\mathit{mayor}, \mathit{student}\ldots : \mathrm{CN}_H} \qquad \frac{A : \mathrm{CN}_H}{A : \mathrm{CN}}$$

Contexts

- Context in type theory is a formal notion

Contexts

- Context in type theory is a formal notion
    - Various way of thinking about contexts
        - List of variable declarations, where variables stand for proofs of the corresponding assumptions

Contexts

- Context in type theory is a formal notion
  - Various way of thinking about contexts
    - List of variable declarations, where variables stand for proofs of the corresponding assumptions
    - A sequence of type judgements

Contexts

- Context in type theory is a formal notion
  - Various way of thinking about contexts
    - List of variable declarations, where variables stand for proofs of the corresponding assumptions
    - A sequence of type judgements
    - Formally, a context is an expression of the form:

$$\Gamma = x_1 : A_1, x_2 : A_2(x_1), \ldots, x_n : A_n(x_1, \ldots, x_{n-1})$$

    - A series of types, and a series of proof objects for these types
    - Any type may depend on any of the previous proof objects
  - They have been used instead of possible worlds for belief intensionality [Ranta, 1994, Chatzikyriakidis and Luo, 2013]
    - A belief context is a collection of some agent's belief (prone to hyperintensional problems, more in lecture 3)

Anaphora

- Anaphora using dependent typing [Ranta, 1994, Boldini, 2000, Chatzikyriakidis and Luo, 2014a]

Anaphora

- Anaphora using dependent typing [Ranta, 1994, Boldini, 2000, Chatzikyriakidis and Luo, 2014a]
- Consider the following:

Anaphora

- Anaphora using dependent typing [Ranta, 1994, Boldini, 2000, Chatzikyriakidis and Luo, 2014a]
- Consider the following:

  A farmer owns a donkey. He loves it.

- Following the end of the first sentence, we have:

Anaphora

- Anaphora using dependent typing [Ranta, 1994, Boldini, 2000, Chatzikyriakidis and Luo, 2014a]
- Consider the following:

  A farmer owns a donkey. He loves it.

- Following the end of the first sentence, we have:

  $$x_1 : (\Sigma x : Farmer)(\Sigma y : Donkey)(own(x, y))$$

- The pronouns pick variables already declared using the projection operators (π_1 and π_2)

  $$x_2 : (love(\pi_1(x_1), \pi_1(\pi_2(x_1)))$$

- Extra reading for people interested: use of signatures. Very informally: more elaborate mechanism than contexts

Recapping: some differences

- Single sorted vs a many-sorted type system
- Dependent typing not available in STT, and thus also IL
- Rich typing and other elaborate mechanisms (contexts, disjoint union types, universes)
- Proof-theoretic specification (supporting effective reasoning)
    - Work on formalizing MTT semantics in proof-assistants
        - See Chatzikyriakidis and Luo [2014b, 2016]
        - Github for a couple of small libraries for NL semantics in Coq (`https://github.com/StergiosCha/CoqNL`)
        - The webpage from the FraCoq and DFraCoq systems `https://github.com/GU-CLASP/FraCoq`

Daniel Gallin. Intensional and higher-order modal logic: with applications to montague semantics. 1975.

Reinhard Muskens. Combining montague semantics and discourse representation. *Linguistics and philosophy*, 19(2):143–186, 1996.

A. Ranta. *Type-Theoretical Grammar*. Oxford University Press, 1994.

G. Sundholm. Constructive generalized quantifiers. *Synthese*, 79 (1):1–12, 1989.

P. Boldini. Formalizing context in intuitionistic type theory. *Fundamenta Informaticae*, 42(2):1–23, 2000.

R. Cooper. Records and record types in semantic theory. *J. Logic and Compututation*, 15(2), 2005.

R. Dapoigny and P. Barlatier. Modeling contexts with dependent types. *Fundamenta Informaticae*, 21, 2009.

D. Bekki. Representing anaphora with dependent types. *LACL 2014, LNCS 8535*, 2014.

C. Retoré. The montagovian generative lexicon Tyn: a type theoretical framework for natural language semantics. In R. Matthes and A. Schubert, editors, *Proc of TYPES2013*, 2013.

S. Chatzikyriakidis and Z. Luo. Adjectival and adverbial modification: The view from modern type theories. *Journal of Logic, Language and Information*, 2017.

S. Chatzikyriakidis and Z. Luo. Adjectives in a modern type-theoretical setting. In G. Morrill and J.M Nederhof, editors, *Proceedings of Formal Grammar 2013. LNCS 8036*, pages 159–174, 2013.

S. Chatzikyriakidis and Z. Luo. Using signatures in type theory to represent situations. *Logic and Engineering of Natural Language Semantics 11. Tokyo*, 2014a.

S. Chatzikyriakidis and Z. Luo. Natural language inference in Coq. *J. of Logic, Language and Information.*, 23(4):441–480, 2014b.

Stergios Chatzikyriakidis and Zhaohui Luo. Proof assistants for natural language semantics. In *International Conference on*

Logical Aspects of Computational Linguistics, pages 85–98. Springer, 2016.

From Montague Semantics to MTT-Semantics: A Meaningful Comparison

Stergios Chatzikyriakidis and Zhaohui Luo

(ESSLLI 2019: Lecture 2 by ZL)

MTT-semantics: some key features

- MTT-semantic features in comparison with Montague
  - Overview, some detailed further in later lectures
  - Only focussing on several, others omitted
- Features of MTT-semantics (& differences with MG)
  - Rich type structure
  - Powerful tools for semantic constructions
  - Both model-theoretic and proof-theoretic
  - Others (if time permits: eg, judgemental interpretation and identity criteria)

I. Rich type structures & CNs-as-types

- In MTTs, many types with rich structures
  - Dependent types (Π-types, Σ-types, …)
  - Inductive types (Nat, Fin(n), …)
  - Universes – types of types (logical, linguistic, …)

  In contrast, simple type theory has only e, t, $\alpha \rightarrow \beta$.
- In linguistic semantics, this allows CNs-as-types.
  - For example, adjectival modification (see SC's lecture)

Classification	Inference	Example	MTT-types/mechanisms
Intersective	$Adj[N] \Rightarrow N \ \& \ Adj$	handsome man	Σ-types
Subsective	$Adj[N] \Rightarrow N$	large mouse	Π-polymorphism
Privative	$Adj[N] \Rightarrow \neg N$	fake gun	disjoint union types
Non-committal	$Adj[N] \Rightarrow\ ?$	alleged criminal	modal collection

- Several advantages of CNs-as-types (not predicates)
  - Selection restriction by typing
    - Capturing category errors
    - MG cannot do this due to CNs-as-predicates.
  - Compatibility with subtyping
    - Subtypes or "subsorts" (cf, Asher, Partee): Phy, Event, Containers, …
    - MG has been problematic in this because of CNs-as-predicates.
  - Proper treatment of copredication
    - Dot-types in MTTs (see SC's latter lecture)
    - Problematic if CNs-as-predicates

I.1. Selection Restriction

- (*) The table talks.
  - Is (*) meaningful? (a category error)
- In MG, yes: (*) has a truth value
  - talk(the table) is false in the intended model.
- In MTT-semantics, no: (*) is not meaningful.
  - "the table" is of type Table, not of type Human and, hence, talk(the table) is ill-typed as talk requires that its argument be of type Human.
  - In MTT-semantics, meaningfulness = well-typedness

I.2. Compatibility with subtyping

- Researchers considered various subtypes/subsorts:
  - Phy, Info ≤ e (Asher 2011 on copredication)
  - Basket ≤ Container, … (Partee-Borschev 2014, on adjectival modification)
  - Events … (c.f., ZL's lecture on Friday)
- Such are incompatible with CNs-as-predicates, although they are OK with CNs-as-types.
  - Let's consider an example – copredication.
  - I'm only using this as an example – see SC's latter Lecture 4.

Copredication

- Copredication (Asher, Pustejovsky, …)
  - John picked up and mastered the book.
  - The lunch was delicious but took forever.
  - The newspaper you are reading is being sued by Mia.
  - … …
- How to deal with this in formal semantics
  - Dot-objects (eg, Asher 2011, in the Montagovian setting)
  - It has a problem: subtyping and CNs-as-predicates strategy do not fit with reach other …

Subtyping problem in the Montagovian setting

- Problematic example (in Montague with CNs-as-pred)
  - heavy : Phy→t and book : Phy•Info→t
  - heavy book = λx:Phy. book(x) ∧ heavy(x) ?
  - In order for this, we'd need Phy ≤ Phy•Info (#)

    But, intuitively, this is not the case (the opposite should be)!
  - A higher type of heavy : (Phy→t)→(Phy→t) would not help.
- In MTT-semantics, because CNs are interpreted as types, things work as intended.
  - heavy : Phy→Prop and Book ≤ Phy•Info ≤ Phy
  - So, heavy(b) : Prop is well-typed, for b : Book.

❖ In MTT-semantics, CNs are types – we have:

"John picked up and mastered the book."

$[\![\text{book}]\!] \leq \text{PHY} \bullet \text{INFO}$

$[\![\text{pick up}]\!] : \text{Human} \rightarrow \text{PHY} \rightarrow \text{Prop}$

$\qquad \leq \text{Human} \rightarrow \text{PHY}\bullet\text{INFO} \rightarrow \text{Prop}$

$\qquad \leq \text{Human} \rightarrow [\![\text{book}]\!] \rightarrow \text{Prop}$

$[\![\text{master}]\!] : \text{Human} \rightarrow \text{INFO} \rightarrow \text{Prop}$

$\qquad \leq \text{Human} \rightarrow \text{PHY}\bullet\text{INFO} \rightarrow \text{Prop}$

$\qquad \leq \text{Human} \rightarrow [\![\text{book}]\!] \rightarrow \text{Prop}$

Hence, both have the same type (in LType – cf, SC's Lect 1.2) and therefore can be coordinated by "and" to form "picked up and mastered" in the above sentence.

Remark: CNs as types in MTT-semantics – so things work.

II. MTT-tools for semantic constructions

- Rich typing → powerful tools
- Examples:
  - Π-polymorphism via universes
  - Overloading by coercive subtyping
  - Π/Σ-organisation (omitted here)
  - … …

II.1. Sense selection via overloading

- Sense enumeration (cf, Pustejovsky 1995 and others)
  - Homonymy
  - Automated selection
  - Existing treatments (eg, Asher et al via disjoint union types)
- For example,
  1. John runs quickly.
  2. John runs a bank.

with homonymous meanings

1. $[\text{run}]_1 : \text{Human}\rightarrow\text{Prop}$
2. $[\text{run}]_2 : \text{Human}\rightarrow\text{Institution}\rightarrow\text{Prop}$

"run" is <u>overloaded</u> – how to disambiguate?

Overloading via coercive subtyping

- Overloading can be represented by coercions

  Eg

$$\begin{tikzcd}
 & {[run]_1 : Human \to Prop} \\
run : 1_{run} \arrow[ur, "c_1"] \arrow[dr, "c_2"'] & \\
 & {[run]_2 : Human \to Institution \to Prop}
\end{tikzcd}$$

- Now, “John runs quickly” = “John [run]$_1$ quickly”.

  “John runs a bank” = “John [run]$_2$ a bank”.
- Homonymous meanings can be represented so that automated selection can be done according to typing.
- Remark: This could not be done if CNs-as-predicates.

II.2. Π-polymorphism – an example mechanism

- Π-polymorphism offers several applications.
  - They are not available in MG where CNs are interpreted as predicates.
- Π-types: Informally (borrowing set-theoretical notations):

  $$\Pi x{:}A.B[x] = \{\, f \mid \text{for any } a : A,\ f(a) : B[a] \,\}$$

  These f's are dependent functions.
- Example
  - $\Pi x{:}\mathrm{Human}.\mathrm{Child}(x)$, type of functions mapping h to Child(h), the type of children of h (may be an empty type).
- Notational conventions:
  - $A \rightarrow B$ stands for $\Pi x{:}A.B(x)$ when $x \notin FV(B)$.
  - In other words, $A \rightarrow B$ are just special cases of Π-types.
  - So, a type theory with Π-types and Prop contains simple type theory.

Π-polymorphism – a first informal look

- How to model predicate-modifying adverbs (eg, quickly)?
  - Informally, it can take a verb and return a verb.
- Montague: quickly : $(e\rightarrow t)\rightarrow(e\rightarrow t)$
  quickly(run) : $e\rightarrow t$
- MTT-semantics: quickly : $(A_q\rightarrow Prop)\rightarrow(A_q\rightarrow Prop)$, where A_q is the domain/type for quickly.
  - What about other verbs? (eg, A_{talk}=Human, …)
  - Can we do it generically – one type of all adverbs?
- Π-polymorphism: quickly : $\Pi A{:}CN.\ (A\rightarrow Prop)\rightarrow(A\rightarrow Prop)$
- Question: What is CN?
  Answer: CN is a universe of types – next slide.

Universes – types of types

- Universe of types
  - Martin-Löf introduced the notion of universe (1973, 1984)
  - A universe is a type of types (Note: the collection Type of all types is not a type itself – otherwise, logical paradox.)
- Examples
  - Math: needing a universe to define type-valued functions
    - f(n) = N x … x N (n times)
  - MTT-semantics: for example,
    - CN is the universe of types that are (interpretations of) CNs. We have:

      Human : CN, Book : CN, Σ(Man,handsome) : CN,   … …
    - We can then have: quickly : ΠA:CN. (A→Prop)→(A→Prop)

Modelling subsective adjectives

- Examples: large, skilful, …
- Nature of such adjectives
  - Their meanings are dependent on the nouns they modify.
  - "a large mouse" is not a large animal ("large" in "a large mouse" is only large for mice, not for other animals/entities.)
- This leads to proposal of using Π-polymorphism:
  - large : $\Pi A{:}CN.\ (A \rightarrow Prop)$
    - CN – type universe of all (interpretations of) CNs
  - $large(Mouse) : Mouse \rightarrow Prop$
  - $[large\ mouse] = \Sigma x{:}Mouse.\ large(Mouse)(x)$

Another example – type of quantifiers

- Generalised quantifiers
  - Examples: some, three, a/an, all, …
  - In sentences like: “Some students work hard.”
- With Π-polymorphism, the type of binary quantifiers is (Lungu 2014):

$$\Pi A{:}CN.\ (A \to Prop) \to Prop$$

For Q of the above type, N : CN and V : N→Prop

$$Q(N,V) : Prop$$

E.g., Some(Student, work_hard) : Prop

CNs-as-predicates in MTTs?

- What if using an MTT but CNs-as-predicates?
  - In an MTT, one could still formally follow Montague:
    - use a <u>single type e</u> of all entities, and
    - use <u>predicates</u> of type e→t to interpret CNs.
- First, this seems unnecessary, at least.
  - Why doesn't one just use simple type theory STT?
  - STT is a simpler "<u>subsystem</u>" – why much bigger system?
- Secondly, most (if not all) of the advantages would be lost …

III. MTT-sem is both model/proof-theoretic

- Model-theoretic semantics (traditional)
  - Meaning as denotation (Tarski, …)
  - Montague: NL → (simple TT) → set theory
- Proof-theoretic semantics
  - Meaning as inferential use (proof/consequence)
  - Gentzen, Prawitz, …, Martin-Löf
  - e.g., Martin-Löf's meaning theory
- MTT-semantics
  - Both model-theoretic and proof-theoretic – in what sense?
  - What does this imply?

Formal semantics in Modern Type Theories (MTT-semantics) is both model-theoretic and proof-theoretic.

- NL → MTT (representational, model-theoretic)
  - MTT as meaning-carrying language with its <u>types</u> representing collections (or "sets") and <u>signatures</u> representing situations
- MTT → meaning theory (inferential roles, proof-theoretic)
  - MTT-judgements, which are semantic representations, can be understood proof-theoretically by means of their inferential roles
- Z. Luo. Formal Semantics in Modern Type Theories: Is It Model-theoretic, Proof-theoretic, or Both? Invited talk at LACL14.

MTT-semantics being model-theoretic

- MTTs offer powerful representations.
- Rich type structure
  - Collections represented by types
  - Eg, CNs and their adjective modifications (see earlier slides)
  - Wide coverage – a major advantage of model-theoretic sem
- Useful contextual mechanisms – signatures
  - Various phenomena in linguistic semantics
    (eg, coercion & infinity)
  - Situations (incomplete world) represented by signatures
    (next slide)

MTT-semantics being model-theoretic (cont^{ed})

- Signatures Σ as in (cf, Edin LF [Harper et al 1987])

$$\Gamma \vdash_{\Sigma} a : A$$

with $\Sigma = c_1{:}A_1, \ldots, c_n{:}A_n$

- New forms besides c:A [Luo LACL14]

$$\ldots,\ c{:}A,\ \ldots,\ A \leq_c B,\ \ldots,\ c \sim a : A,\ \ldots$$

  - Subtyping entries (cf, Lungu's PhD thesis 2018)
  - Manifest entries (can be emulated by coercive subtyping)
- *Theorem (conservativity)*

  The extension with new signature entries preserves the meta-theoretic properties for coherent signatures.

MTT-semantics being proof-theoretic

- MTTs are representational with proof-theoretic sem
  - Not available before – cf, use theory of meaning
- MTT-based proof technology
  - Reasoning based on MTT-semantics can be carried out in proof assistants like Coq:
    - pretty straightforward but nice application of proof technology to NL reasoning (not-so-straightforward in the past …)
  - Some Coq codes can be found in: (example next slide)
    - Z. Luo. Contextual analysis of word meanings in type-theoretical semantics. Logical Aspects in Computational Linguistics. 2011.
    - S. Chatzikyriakidis & Z. Luo. NL Inference in Coq. JoLLI 23(4). 2014.
    - S. Chatzikyriakidis & Z. Luo. Proof assistants for NL semantics. LACL 2016.
    - T. Xue et al. Propositional Forms of Judgemental Interpretations. NLCS 2018.

Coq code (homonymy by overloading)

```
Definition CN := Set.
Parameters Bank Institution Human Man : CN.
Parameter John : Man.
Axiom mh : Man->Human.  Coercion mh : Man >-> Human.
Axiom bi : Bank->Institution.  Coercion bi : Bank >-> Institution.

(* unit type for "run" *)
Inductive Onerun : Set := run.
Definition T1 := Human->Prop.
Definition T2 := Human->Institution->Prop.
Parameter run1 : T1.
Parameter run2 : T2.
Definition r1 (r:Onerun) : T1 := run1.  Coercion r1 : Onerun >-> T1.
Definition r2 (r:Onerun) : T2 := run2.  Coercion r2 : Onerun >-> T2.

(* John runs quickly *)
Parameter quickly : forall (A:CN), (A->Prop)->(A->Prop).
Definition john_runs_quickly := quickly (run:T1) John.
(* John runs a bank *)
Definition john_runs_a_bank := exists b:Bank, (run:T2) John b.
```

Why important?

- Model-theoretic – powerful semantic tools
  - Much richer typing mechanisms for formal semantics
  - Powerful contextual mechanism to model situations
- Proof-theoretic – practical reasoning on computers
  - Existing technology: proof assistants (Coq, Agda, Lego/Plastic, NuPRL)
  - Applications to NL reasoning
- Leading to both of
  - Wide-range modelling as in model-theoretic semantics
  - Effective inference based on proof-theoretic semantics

Remark: MTT-semantics offers a new perspective – new possibility not available before!

IV. Several Further Features of MTTs

- Other features/topics:
  - Judgemental interpretations (Xue, Luo & Chatzikyriakidis 18)
  - Identity criteria of CNs (Luo 2012, Chatzikyriakidis & Luo 2018 – see SC's latter lecture)
  - Proof irrelevance (Luo 2019)
- First, introducing the last one (and, the others if time permits)

Proof Irrelevance

- Example to show:
  - Potential problem introduced by proof terms in MTTs (and how to solve it by proof irrelevance)
  - From another angle, MTTs are very powerful for semantics.
- Proof irrelevance
  - Any two proofs of the same proposition are the same.
  - To have adequate MTT-semantics, proof irrelevance needs be enforced in the underlying type theory.
  - Eg, in impredicative TTs like UTT, we can have

$$\frac{\Gamma \vdash P : Prop \quad \Gamma \vdash p : P \quad \Gamma \vdash q : P}{\Gamma \vdash p = q : P}$$

Examples in NL semantics

- Identity criteria for CNs [Luo 12, Chatzikyriakidis & Luo 18]
  - A handsome man is a pair (m,p) of type Σ(Man, handsome).
  - Two handsome men are the same iff they are the same man
    ➔ proof irrelevance (eg, proofs of handsome(m) are the same).
- Counting (the same problem as above)
  - Most students who passed some exams are happy.
    - Most z : [Σx:Student Σy:Exam.pass(x,y)]. happy(π_1(z))
    - Incorrect counting that takes proofs of Σy:Exam.pass(x,y) into account
  - I believe proof irrelevance provides a clean/easier solution.
    - Most z : [Σx:Student $\exists$y:Exam.pass(x,y)]. happy(π_1(z))
    - Correct counting by proof irrelevance (for the $\exists$-proposition)

Counting and Anaphora

- A problem when both are involved.
  - Thanks to Justyna Grudziñska for bringing this example to my attention.
- Most farmers who own a donkey beat it.
  - (#) Most z : [$\Sigma x{:}F\ \Sigma y{:}D.\ own(x,y)$]. $beat(\pi_1(z), \pi_1(\pi_2(z)))$
  - Incorrect counting as proofs in Σ are taken into account.
  - Note that, if you use traditional $\exists$ for both Σ to get correct counting, anaphora are problems (untyped – π_i don't exist)!
- A problem not solved satisfactorily before
  - Sundholm (1989) realised it, but only proposed an ad hoc solution.
  - Tanaka (2015) studied a similar solution (ad hoc & complicated).

- A solution in UTT (or MLTT$_h$), using both Σ and $\exists$:
  - Most $z : [\Sigma x{:}F\ \exists y{:}D.\ own(x,y)].$
    $\forall z' : [\Sigma y'{:}D.\ own(\pi_1(z), y')].\ beat(\pi_1(z), \pi_1(z'))$
  - Correct counting because of proof irrelevance (for $\exists$-prop).
  - Correct anaphoric reference because of Σ.
- Details in my LACompLing18 paper (in press)
  - Title: "Proof Irrelevance in Type-Theoretical Semantics"
  - First (?) to do this in a "standard" logical system, rather than non-standard ones such as the non-monotonic Dynamic Predicate Logic.

Judgemental Interpretation

- Judgements v.s. propositions
- Example
  - John is a student.
  - j : Student (Here, Student is a type, not a predicate.)
- What about
  - John is not a student.
  - If John is a student, he is happy.
- The following are wrong:
  - (#) ¬(j : Student) -- illegitimate
  - (#) (j : Student) is not the case. -- a meta-level assertion

- So, we introduce IS(Student,x) : Prop
  - $\neg$IS(Student,j)
  - IS(Student,j) $\Rightarrow$ happy(j)
  - These are well-typed propositions.
- $IS_A(B,a)$
  - Introduced axiomatically.
  - Justified by heterogeneous equality in type theory.
  - Xue, Luo and Chatzikyriakidis. Propositional Forms of Judgemental Interpretations. NLCS18, Oxford. 2018.

From Montague Semantics to MTT-semantics: A Meaningful Comparison

Lecture 3: Case Study on modification (given by SC)

Stergios Chatzikyriakidis and Zhaohui Luo

centre for
linguistic theory
and studies in probability

August 14 2019

Overview

- Case study on modification
    - Aspects of both adjectival and adverbial modification
- Providing a standard/representative account in the Montagovian tradition along with an MTT one
    - Not meant to be exhaustive: MG approaches to modification are numerous
        - Our goal is not to compare against every conceivable MG account, but to provide comparisons that will highlight differences and similarities between the two systems

Modification: adjectival

- Traditional coarse grained classification of adjectives [Kamp(1975), Partee(2010)]
  - Intersective (e.g. *black*)
  - Subsective (e.g. *skillful*, *small*)
  - Non-subsective
    - Privative (e.g. *fake*)
    - Non-committal (e.g. *alleged*)

Modification: adjectival

- Classification based on basic inferential properties

Classification	Inference	Example	MTT-types/mechanisms
Intersective	Adj[N] $\Longrightarrow$ N & Adj	handsome man	Σ-types
Subsective	Adj[N] $\Longrightarrow$ N	large mouse	Π-polymorphism
Privative	Adj[N] $\Longrightarrow$ $\neg$N	fake gun	disjoint union types
Non-committal	Adj[N] $\Longrightarrow$?	alleged criminal	modal collection

Modification: adjectival

- Other types of adjectives we are going to be looking at
  - Gradable adjectives
  - Multidimensional adjectives

Intersective/Subsective/Non-subsective: the view from Montague Semantics

- Standard Montagovian approach: adjectives are higher-order properties
  - Type for adjectives: $(s \rightarrow (e \rightarrow t)) \rightarrow (s \rightarrow (e \rightarrow t))$ (intersectives are taken by some authors to be of type $s \rightarrow (e \rightarrow t)$)
  - Inferential properties are then captured by using meaning postulates
    - $\exists P : e \rightarrow t. \forall Q : e \rightarrow t. \forall x : e. ADJ'(Q)(x) \rightarrow P(x) \wedge Q(x)$ for each intersective adjective
    - $\forall Q : e \rightarrow t. \forall x : e. ADJ'(Q)(x) \rightarrow Q(x)$ for each subsective adjective
    - $\forall Q : e \rightarrow t. \forall x : e. ADJ'(Q)(x) \rightarrow \neg Q(x)$ for each privative adjective
    - $\forall Q : e \rightarrow t. \forall x : . ADJ'(Q)(x) \rightarrow ?$ for each non-committal adjective

Intersective/Subsective/Non-subsective: the view from Montague Semantics

- Issues to consider: relies on meaning postulates
  - Meaning postulate for intersective adjectives does not exactly work. Why is this so?
    - Well, the existentially quantified P property is basically the property of the adjective, e.g. for *black*, it has to be property *Black*
    - Very difficult to secure that this property will be the one that the adjective needs
  - This existential quantification is vacuous, e.g. there is always the truth predicate satisfying this existential

Intersective/Subsective/Non-subsective: the view from MTTs

- Adjectival modification as involving Σ types, in line with Ranta (1994)
  - Intersective adjectives as simple predicate types and subsective as polymorphic types over the CN universe:
    - $[\![black]\!] : Object \to Prop$
    - $[\![small]\!] : \Pi A{:}CN.(A \to Prop)$ (the A argument is implicit)
    - For *black man*, we have: $\Sigma m{:}\, [\![man]\!] \,.\, [\![black]\!](m) < [\![man]\!]$ (via π_1)
      $< \Sigma m{:}\, [\![human]\!] \,.\, [\![black]\!](m)$ (via subtyping propagation)
      $< [\![human]\!]$ (via π_1)
  - For *small man*:
    - $\Sigma m{:}\, [\![man]\!] \,.\, [\![small]\!]\, [\![man]\!](m) < [\![man]\!]$ (via π_1)
    - BUT NOT: $\Sigma m{:}\, [\![man]\!] \,.\, [\![small]\!]\, [\![man]\!](m) < \Sigma a{:}\, [\![animal]\!] \,.\, [\![small]\!]\, [\![man]\!](a)$
    - Many instances of small: $small([\![man]\!])$ is of type $[\![man]\!] \to Prop$, $small([\![animal]\!])$ is of type $[\![animal]\!] \to Prop$

Intersective/Subsective/Non-subsective: the view from MTTs

- Privative adjectives like *fake*
  - We follow Partee (2007) and argue that privative adjectives are actually subsective adjectives which operate on CNs with extended denotations
    - For exaple, the denotation of *fur* is expanded to include both *real* and *fake* furs:

      (1) I don't care whether that fur is fake fur or real fur.

      (2) I don't care whether that fur is fake or real.

    - $G = G_R + G_F$ with $inl(r){:}G_R$ and $inr(f){:}G_F$
    - define: $real_g(inl(r)) = True \quad and \quad real_g(inr(f)) = False$; $fake_g(inl(r)) = False \quad and \quad fake_g(inr(f)) = True$.

Intersective/Subsective/Non-subsective: the view from MTTs

- Use Π-polymorphism to give a generic typing for privative adjectives like *fake*
  - Note that this conforms to Partee's idea that privative adjectives are in fact subsective
    - When fake is instantiated for a specific common noun A, we obtain a specific meaning $fake(A)$ for that CN
      $fake,\ real\ :\ \Pi A : \text{CN}.\ (A \to Prop)$
      $fake(G) = fake_g$

Intersective/Subsective/Non-subsective: the view from MTTs

- We can define the adjective modifications by means of Σ
  - $\textit{fake gun} = \Sigma g : G.\ \textit{fake}(G, g)$
  - $\textit{real gun} = \Sigma g : G.\ \textit{real}(G, g)$

Intersective/Subsective/Non-subsective: the view from MTTs

- Note: there are no inferences associated with non-committal adjectives in this categorization
  - But, we want to say something about adjectives like *alleged*
    - We will use *alleged* as our example
  - Consider the example *John is an alleged murderer*
    - Informally: someone alleged that John is a murderer
    - Treatment: use modal collections!

Intersective/Subsective/Non-subsective: the view from MTTs

- $\exists h : Human.\ H_{h,alleged}(IS_{Human}(Murderer, j))$
- For each non-committal adjective, we have a corresponding modal operator.
  - for alleged, $H_{h,alleged}(P)$ says that h alleges P
  - for any $h : Human$ and any non-committal adjective Adj, the modal operator $H_{h,Adj}$ is a predicate over propositions:

$$H_{h,Adj} : Prop \to Prop.$$

  - $H_{h,Adj}$ is a collection of propositions
    - $H_{h,alleged}$ is the collection of allegations made by h.

$alleged_B : \text{CN} \to \text{CN} = \lambda A : \text{CN}.\ \Sigma x : B.\ \exists h : Human.\ H_{h,alleged}(IS_B(A, x))$

Intersective/Subsective/Non-subsective: the view from MTTs

- Inferential properties by means of typing (at least for intersectives, subsectives and privatives) vs meaning postulates
- No higher order types vs higher order types
- No intensional types vs intensional types (in the sense of Montague)
- Rich typing stuctures help (Σ and Π types, disjoint union types) help in giving semantics for adjectives

Gradable adjectives: Kennedy 2007

- Gradable adjectives involve a degree parameter that is context sensitive, e.g. *tall for a woman, tall for a man* etc.
- A prominent, representative analysis by [Kennedy(2007)] unfolds as follows:
  - The positive form of an adjective works as follows:
    - It is a higher order measure function, where measure function is a function from entities to degrees ($e \to d$)
    - The definition: $pos = \lambda F : e \to d.\lambda x : e.(F(x) \geq \mathbf{s}(F))$
    - **s** is a context sensitive function from measure functions to degrees
    - Composing comparison classes with adjectives via Heim's . combinator. Consider composing the comparison class basketball player, $BB : e \to t$ and $tall : e \to d$ to $BB(tall) : e \to d$. Normal functional application will not work here, thus the use of the "dot" combinator

Gradable adjectives: MTT-semantics

- We concentrate here on the analysis sketched in [Chatzikyriakidis and Luo(2019a), Chatzikyriakidis and Luo(2019b)]
    - Crucial assumptions
        - Use of indexed types, common nouns are indexed with degree parameters
        - Use of type polymorphism to get the desired context-dependency

Gradable adjectives: MTT-semantics

- Let us take *tall* as an example
  - We introduce the universe *Degree* of all degree types, where $Height, Width... : Degree$
- Arguments of gradable adjectives are not simple types, but rather types indexed by degree parameters
  - For example, *human* can be refined he family of types indexed by heights: $HHuman : Height \rightarrow Type$
    - $HHuman(n)$ is the type of humans of height n.
      ,
- Definition of a height function
  - $height : \Pi i : Height.\ HHuman(i) \rightarrow Height$
  - $height(i, h) = i$.

Gradable adjectives: MTT-semantics

- We need a polymorphic standard for *tall*
  - The value is dependent on the noun, the adjective, and sometimes even some other contextual information
    - These correspond to types, a predicates and a contexts (in type theory)
    - Polymorphism and type dependency in MTTs is used
  - For $D : Degree$, we introduce the indexed universe $\mathrm{CN}_G(D)$, a subuniverse of CN, consisting of the CNs with the indexed degree

Gradable adjectives: MTT-semantics

- General rules for $\text{CN}_G(D)$,

$$\frac{D : \mathit{Degree}}{\text{CN}_G(D) : \mathit{Type}} \qquad \frac{D : \mathit{Degree} \quad A : \text{CN}_G(D)}{A : \text{CN}}$$

- The polymorphic standard, STND.
  - $\mathrm{STND} : \Pi D : \mathit{Degree}\ \Pi A : \text{CN}_G(D).\ \mathit{ADJ}(A) \to D$
    $\mathit{ADJ}(A)$ is the type of syntactic forms of adjectives whose semantic domain is A
- Definition of tall:
  - $\mathit{tall} : \Pi i : \mathit{Height}.\ \mathit{HHuman}(i) \to \mathit{Prop}$
  - $\mathit{tall}(i, h) = \mathit{height}(i, h) \geq \mathrm{STND}(\mathit{Height}, \mathit{Human}, \mathit{TALL})$

Gradable adjectives: Comparison

- The accounts achieve similar results. However:
  - Indexing on the noun by means of a degree gives one for free the fact that we are not talking about tallness in general but tallness with respect to the relevant class
    - No need for the “dot” combinator
  - The polymorphic STND function is, we believe, a more straightforward interpretation of Kennedy’s context sensitive function from measure functions (adjectives basically) to degrees

Multidimensional adjectives

- Gradability across more than one domain
  - E.g. consider *healthy*: to be considered healthy, one has to be healthy across all "health dimensions" (or most of them)
    - Healthy is an example of a positive multidimensional adjective, where quantification is across all (or most) of its degrees
  - Consider its antonym, *sick*: to be considered sick, one dimension is enough
    - Sick is an example of a negative multidimensional adjective

Multidimensional adjectives

- The account by [Sassoon(2012)]: introduction of a higher order DIM predicate, taking predicates as arguments (healthy, sick)
- $Healthy = \lambda x.\forall Q \in DIM(healthy).healthy\text{-}wrt(x, Q)$
- $Sick = \lambda x.\exists Q \in DIM(healthy).healthy\text{-}wrt(x, Q)$

Multidimensional adjectives: MTTs

- Account put forth in [Chatzikyriakidis and Luo(2019b)]
  - Consider *healthy*.
  - We introduce the inductive (basically an enumerated type) *Health*
    - Inductive *Health* : D := Heart | Blood_ pressure | Cholesterol.
  - We then define:
    - $Healthy = \lambda x{:}Human.\forall h : Health.healthy(h)(x)$
    - $Sick = \lambda x{:}Human.\neg(\forall h : Health.healthy(h)(x))$

Multidimensional nouns

- A noun like *bird*, at least according to theories like Prototype and Exemplar theories is argued to involve a rich couple of dimensions
    - For something to count as a bird, a couple of different dimensions (for example, dimensions like *winged*, *small*, *can breed*, etc.) have to be taken into consideration
    - Conceptual structure of a noun like *bird* will involve an ideal value for each dimension.
    - Similarity measure is mapping entities to degrees, representing how far from the ideal dimensions of the prototype the values for the respective entities are (it is claimed that this is represented as a weighted sum)

Multidimensional nouns

- Dimensions integrate (another way of putting it is collapse) into a unique degree, thus not accessible for quantification as it is the case with multidimensional adjectives
- Viewing common nouns as types seems to be compatible with this.
- In order for an object to be of a CN type, the standard of membership w.r.t the weighted sum of its similarity degrees to the ideal values in the dimensions of the noun has to be exceeded.
- Between these two types of multidimensionality, i.e. multidimensional adjectives like *healthy* and multidimensional nouns like *bird*, we find social nouns like *linguist, artist*.
- Behave like multidimensional adjectives, with dimensions accessible:
  - He is an artist in many respects.

Multidimensional nouns: formalizing

- Types in these cases become more elaborate
    - Consider *artist*, and the inductive type for all its dimensions
        - Inductive $Art : D := a_1 \mid a_2 \mid a_3$
        - $artist = \Sigma h : Human.\forall a : Art.DIM_A(h, a) \times a \geq ...$ (we need a definition of a standard for multidimensional adjectives) with $DIM_{CN} : \Pi h : Human.\ \Pi a : Art.\ F(a)$ and $F : Art \rightarrow Nat$
        - Working on this!

Multidimensional adjectives

- MTT rich typing and typing structures allows one to introduce suitable structures for multidimensional adjectives
- The account seems to be fit to extend to cases of multidimensional nouns

Modification: adverbial

- Basic typings for adverbs
  - Montague Semantics
    - $(s \to (e \to t)) \to (s \to (e \to t))$
    - $t \to t$
  - MTT-semantics
    - $\Pi A : \textsc{cn}.\ (A \to Prop) \to (A \to Prop)$
    - $Prop \to Prop$

Modification: veridical adverbs

- If a proposition A comprised of the adverb plus a proposition B (or the VP) is true, then the B is true as well
  - Fortunately, John opened the door $\Longrightarrow$ John opened the door
- A classic Montagovian analysis(e.g. [Montague(1970), Kamp(1975), Parsons(1972)] can of course do this by providing a meaning postulate

Modification: veridical adverbs

- Define $VER_{Prop}(v)$ (with $VER_{Prop} : Prop \rightarrow Prop$)
  - $VER_{Prop}(v) = \forall P : Prop.(P \rightarrow v)$
- With this we can prove:
  1. $fortunately(v) \rightarrow v$ (for $v : Prop$)
  2. $\forall P : Prop.\ (P \rightarrow v) \rightarrow fortunately(v)$

Modification: manner adverbs

- Classic treatments involve neo-davidsonian assumptions
- *slowly*: the event under consideration is a slow one.
    - However, it is the manner of the event rather than the event itself that is slow
        - Inclusion of manners in the semantic ontology [Dik(1975), Schäfer(2008)]
        - *John wrote illegibly* $=$ $\exists e[subject(john, e) \wedge write(e) \wedge \exists m[manner(m, e) \wedge illegible(e)]]$

Modification: manner adverbs in MTTs

- We introduce the type $Event : Type$.
- We further introduce $Manner : Type$
  - Assume the family of types $Event : Manner \to Type$ indexed by manners
    - $Event_m$ (with $m : Manner$) is the type of events of manner m.
    - typing: $ADV_{manner} : \Pi m : Manner.\ \Pi A : \mathrm{CN}.\ (A \to Event_m \to Prop) \to (A \to Event_m \to Prop)$ (m and A are implicit)
  - Lexical entry for *illegibly*:
    - $illegibly = \lambda P : (Human \to Event_m \to Prop).\lambda x : Human.\lambda E : Event_m.P(x)(E) \wedge illeggible(m)$
      with $illegible : Manner \to Prop$

Manner adverb modification: Brief Comparison

- MTT semantics can capture at least as much as the MG approaches
  - One might argue that the rich typing structures give a more principled way of talking about manners, events etc.
    - Within a Montagovian setting, based on Church's simple types, one has to do something extra to accommodate additional types like *Manner*

Adverb modification: Intensional Adverbs

- Epistemic adverbs create opaque contexts for both the subject and the object, while adverbs like *intentionally*, only for the object:
  - Oedipus allegedly married Jocaste.
  - Oedipus intentionally married Jocaste.
    - From the first, we obtain:
    1. Oedipus allegedly married Jocaste $\nRightarrow$ the son of Laius allegedly married Jocaste
    2. Oedipus allegedly married Jocaste $\nRightarrow$ Oedipus allegedly married his mother

    - From the second we obtain:
    1. Oedipus intentionally married Jocaste $\Rightarrow$ The son of Laius intentionally married Jocaste
    2. Oedipus intentionally married Jocaste $\nRightarrow$ Oedipus intentionally married his mother

Adverb modification: Intensional Adverbs

- Montague semantics: use possible worlds semantics, intensional typings for adverbs
    - Generalizing to the worst case: all adverbs, even non-intensional ones have intensional typings

Adverb modification: Intensional Adverbs

- We concentrate on *intentionallly* here
- [Chatzikyriakidis and Luo(2017)] propose that A intentionally P means that A has the intention P and furthermore fulfilled this intention, i.e. P holds
  - Introduction of intention contexts, which represent an agent's collection of intentions
    $\mathcal{D}_p = x_1 : A_1, ..., x_n : A_n(x_1, ..., x_{n-1})$
  - Use of a generalized intention operator I_p
    - $Intentionally = \lambda x : Human.\lambda P : Human \to Prop.\ I_x(P(x))$

Adverb modification: Intensional Adverbs

- Suffers from hyperintensionality!
  - Classical treatment of beliefs by [Ranta(1994)], which is used by [Chatzikyriakidis and Luo(2017)] is prone to hyperintensional problems
    - Beliefs are closed under derivability: if one believes P, s/he believes every proposition that is logically equivalent to P. This extends to intentional contexts as well
    - if $M(O, J)$ is part of Oedipus' intentional context, then one can derive that Oedipus intentionally married his mother given that $M(O, J) = M(O, MoO)$

Adverb modification: Intensional Adverbs

- Solution: Modal collections
  - We introduce modal collections, representing agents' collections of beliefs, intentions etc.
    - $B_h : Prop \rightarrow Prop \ (B_h(P) if P \in B_h \ )$
    - $I_h : Prop \rightarrow Prop \ (I_h(P) if P \in I_h \ )$
    - $Intentionally = \lambda x : Human.\lambda P : Human \rightarrow Prop.\ I_x(P(x))$
    - If $B_h(P)$ and you can derive Q from P, you cannot derive $B_h(Q)$ if Q is not in B_h

Modification: End Remarks

- A nice test case for formal semantics: diverse and notoriously difficult
  - A further nice test to compare alternative theories of formal semantics
- MTTs vs Montague Semantics
  - Inferences via typing instead of meaning postulates
  - Rich typing structures provide a more natural habitat for more fine-grained phenomena, where more types are needed (e.g. manners, events)
  - Direct support of proof-theoretic reasoning
  - No need for intensional typing, let alone for cases it is not needed

H. Kamp.
Formal semantics of natural language.
In E. Keenan, editor, *Two Theories about Adjectives*, pages 123–155. Cambridge University Press, 1975.

B. Partee.
Privative adjectives: Subsective plus coercion.
In R. Bauerle and U. Reyle, editors, *Presuppositions and Discourse: Essays Offered to Hans Kamp*, pages 123–155. Emerald group publishing, 2010.

Christopher Kennedy.
Vagueness and grammar: The semantics of relative and absolute gradable adjectives.
Linguistics and philosophy, 30(1):1–45, 2007.

S. Chatzikyriakidis and S. Luo.
Gradability in MTT-semantics.
In *TbiLLC 2019: Thirteenth International Tbilisi Symposium on Language, Logic and Computation*, 2019a.

Stergios Chatzikyriakidis and Z Luo.
Formal Semantics in Modern Type Theories.
Wiley & ISTE Science Publishing Ltd, 2019b.

G.W. Sassoon.
A typology of multidimensional adjectives.
Journal of semantics, 2012.

R. Montague.
English as a formal language.
In B. Visentini, editor, *Linguaggi nella Societ e nella*. Milan: Edizioni di Comunit,l, 1970.

T. Parsons.
Some problems concerning the logic of grammatical modifiers.
In D. Davidson and G. Harman, editors, *Semantics of Natural Language.*, pages 127–141. Dordrecht:Reidel, 1972.

Simon Dik.
The semantic representation of manner adverbials.

In Albert Kraak, editor, *Linguistics in the Netherlands 19721973*. Assen: Van Gorcum, 1975.

Martin Schäfer.
Resolving scope in manner modification.
In *Empirical issues in syntax and semantics*, volume 7, pages 351–372. 2008.

Stergios Chatzikyriakidis and Zhaohui Luo.
Adjectival and adverbial modification: The view from modern type theories.
Journal of Logic, Language and Information, 26(1):45–88, 2017.

A. Ranta.
Type-Theoretical Grammar.
Oxford University Press, 1994.

From Montague Semantics to MTT-semantics: A Meaningful Comparison

Lecture 4: Case study on copredication

Stergios Chatzikyriakidis and Zhaohui Luo

August 15 2019

CLASP centre for linguistic theory and studies in probability

Copredication

- More than one predicate, representing a verb or an adjective and requiring different types of arguments

  (1) John picked up and mastered the book

  (2) The lunch was delicious but took forever

- *Picked up* seems to require a physical object while *mastered* an informational one. *The book* is thus interpreted as involving both a physical and an informational aspect
- Similarly, *lunch* appears to involve both an a *event* as well as a *food* aspect
- Originally (at least to a great extent), discussed in Pustejovksy (1995)
  - It has given rise to a number of approaches since then (Cooper 2008, Asher 2011, Luo 2010, Chatzikyriakidis and Luo 2015, Gotham 2014, 2016 among others)

Copredication and Individuation criteria

- Copredication interacts in interesting ways with individuation. Consider the following:

  (3) John picked up and mastered three books

- What the above sentence means is that John picked up three physical objects and mastered three informational ones
  - It cannot mean that John picked up three physical objects but mastered one informational object (e.g. a case where John picked up three copies of the same book) or vice versa
  - Accounts of copredication should be able to correctly capture the individuation facts
  - Our account of individuation should be elaborate enough in order to capture this data

The earlier accounts in MTT semantics: Luo's 2010 account

- The account for *dot-types* as proposed in [Luo(2010), Xue and Luo(2012)] proceeds as follows (we use the example of physical and informational objects):
  - $\textsc{Phy}$ and $\textsc{Info}$ are the types of physical objects and informational objects
  - $\textsc{Phy} \bullet \textsc{Info}$ is the dot-type of objects having both a $\textsc{Phy}$ and an $\textsc{Info}$ aspect
  - The *dot-type* is a subtype of its constituent types
    $\textsc{Phy} \bullet \textsc{Info} < \textsc{Phy}$
    $\textsc{Phy} \bullet \textsc{Info} < \textsc{Info}$
  - The two aspects of *book* are then represented as: $Book < \textsc{Phy} \bullet \textsc{Info}$

CLASP centre for linguistic theory and studies in probability

Luo's account of dot types- An example

- Let us consider the following example:

(4) John picked up and mastered the book.

- The verbs are given the following typings:

$$\begin{aligned} [\![\mathit{pick\ up}]\!] &: [\![\mathit{human}]\!] \to \textrm{P}{\small\textrm{HY}} \to \mathit{Prop} \\ [\![\mathit{master}]\!] &: [\![\mathit{human}]\!] \to \textrm{I}{\small\textrm{NFO}} \to \mathit{Prop} \end{aligned}$$

- Due to contravariance of function types (and the declared subtyping relations of course) we have:

$$\begin{aligned} [\![\mathit{pick\ up}]\!] &: [\![\mathit{human}]\!] \to \textrm{P}{\small\textrm{HY}} \to \mathit{Prop} \\ &< [\![\mathit{human}]\!] \to \textrm{P}{\small\textrm{HY}} \bullet \textrm{I}{\small\textrm{NFO}} \to \mathit{Prop} \\ &< [\![\mathit{human}]\!] \to [\![\mathit{book}]\!] \to \mathit{Prop} \end{aligned}$$

CLASP centre for linguistic theory and studies in probability

Luo's account of dot types- An example

- Continued

$$\begin{array}{rcl} [\![\mathit{master}]\!] & : & [\![\mathit{human}]\!] \to \textsc{Info} \to \mathit{Prop} \\ & < & [\![\mathit{human}]\!] \to \textsc{Phy} \bullet \textsc{Info} \to \mathit{Prop} \\ & < & [\![\mathit{human}]\!] \to [\![\mathit{book}]\!] \to \mathit{Prop} \end{array}$$

- Therefore, $[\![\mathit{pick\ up}]\!]$ and $[\![\mathit{master}]\!]$ can both be used in a context where terms of type $[\![\mathit{human}]\!] \to [\![\mathit{book}]\!] \to \mathit{Prop}$ are required
- See the paper for the formal details and introduction/elimination rules for dot-types

CLASP centre for linguistic theory and studies in probability

Chatzikyriakidis and Luo 2015 account

- The authors there try to show that the account as is, will give the correct identity criteria
- Formalization in the proof assistant Coq
  - There is no universe construction in Coq, so the universe is CN is taken to be Set
  - *Man* and *Human* are declared as types. The third line involves declaring subtyping relations
  - The quantifier *three* is given a definition according to which three elements different to each other are hold for a property P
  - The rest of the code follows the analysis in Luo (2011)

Chatzikyriakidis and Luo 2015 account

```
Definition CN:= Set.
Parameter Man Human: CN. Parameter John: Man.
Axiom mh: Man->Human. Coercion mh: Man>->Human.
Parameter Phy Info: CN.
Record PhyInfo: CN:= mkPhyInfo{phy:>Phy;info:>Info}.
Definition Three:= fun(A: CN)(P: A->Prop)=>exists x: A,P x /\
(exists y: A,P y/\(exists z: A,P z /\ x<>y /\ y<>z /\ x<>z)).
```

CLASP centre for linguistic theory and studies in probability

Chatzikyriakidis and Luo 2015

- We need to be able to capture the following inferences:
  1. John picked up and mastered three books ⇒ John picked up three books and mastered three books
  2. John picked up and mastered three books ⇒ John picked up three physical objects and mastered three informational objects
- We formulated the desired inferences as Coq theorems
- We attempt to prove them using Coq's proof tactics
  - 1. is easilyproven
  - 2. does not go through

CLASP centre for linguistic theory and studies in probability

Chatzikyriakidis and Luo 2015

- The authors (well, us!) introduce axioms for equality under subtyping:

  In general, when $X <_c Y$, we do not have $x \neq_X y \implies (x \neq_Y y)$ unless c is injective. For the atomic types like Book and $\textsc{Phy}$, *the equality on a subtype coincides with that of the supertype and so we can axiomatically assume this.* (Chatzikyriakidis and Luo, 2015)
- The following two axioms are encoded in Coq in this respect

```
Variable PHY:forall x:Book, forall y:Book,
not(x=y:>Book)-> not(x=y:>Phy).
Variable INFO:forall x:Book, forall y:Book,
not(x=y:>Book)-> not(x=y:>Info).
```

- These axioms help in proving the examples involving *three*. But at a cost!

CLASP centre for linguistic theory and studies in probability

Chatzikyriakidis and Luo 2015

- The account overgenerates
  - Examples like the following can also be proven:

    (5) John picked up three books and John mastered every book $\Rightarrow$ John mastered three books.

  - Thanks to Matthew Gotham who pointed this out to us!
- In general: this is not a principled solution but a hack

centre for linguistic theory and studies in probability

What is the problem?

- The previous accounts relied on the implicit idea that every type defines its own equivalence relation
  - CNs are basically setoids (as e.g. argued in Luo 2012). Thus, the interpretation of a CN is not just a type, rather a type associated with an identity criterion for that CN)
  - A type plus an equivalence relation on the type
    - ⋆ An IC if you wish
- However, this has not been explicitly specified in the previous accounts of individuation
- Individuation in copredication is a clear case where this idea needs to get explicit
  - This is what we are going to do here

CLASP centre for linguistic theory and studies in probability

Criteria of Identity/Individuation: CNs as Setoids

- Individuation is the process by which objects in a particular collection are distinguished from one another
  - Provides us with means to count and a sameness criterion
- In linguistic semantics, individuation is related to the idea that a CN may have its own identity criterion for individuation [Geach(1962)]
- Mathematically, the association of an equivalence relation (the identity criterion) CNs
  - In constructive mathematics, a set or a type is indeed a collection of objects together with an equivalence relation that serves as identity criterion of that collection

CLASP centre for linguistic theory and studies in probability

CNs as Setoids

- CNs are not just types
  - Types plus an identity criterion for that specific CN

    (6) $(A, =)$

    where A is a type and $=: A \to A \to Prop$ is an equivalence relation over A
  - The difference between CNs-as-Types and CNs-as-Setoids

    (7) $[human] = Human : Type$ (CNs-as-types view)

    (8) $[human] = (Human, =_h)$ (CNs-as-Setoids view)

CLASP centre for linguistic theory and studies in probability

CNs as Setoids

- Consider the following examples and their semantic interpretations:

(9) Three men talk.

(10) Three humans talk.

(11) $\exists x, y, z : Man.\ x \neq_M y \ \wedge y \neq_M z \ \wedge x \neq_M z \ \wedge talk(x) \ \wedge talk(y) \ \wedge talk(z)$

(12) $\exists x, y, z : Human.\ x \neq_H y \ \wedge y \neq_H z \ \wedge x \neq_H z \ \wedge talk(x) \ \wedge talk(y) \ \wedge talk(z)$

where $\textrm{MAN} = (Man, =_M)$ and $\textrm{HUMAN} = (Human, =_H)$ are setoids and the identity criterion for men and that for humans are used to express that x, y and z are distinct from each other.

CNs as Setoids

- Necessary to consider the individuation criteria explicitly by using the identity criteria $=_M$ and $=_H$
- The relationship between the $\textsc{Man}$ and $\textsc{Human}$ is one where the first inherits the IC from the second

(13) $(=_M) = (=_H)|_{Man}$

CLASP centre for linguistic theory and studies in probability

CNs as Setoids: Subsetoids

- $\mathrm{A} = (A, =_{\mathrm{A}})$ is a $\mathrm{sub\text{-}setoid}$ of $\mathrm{B} = (B, =_{\mathrm{B})}$, notation $\mathrm{A} \sqsubseteq \mathrm{B}$, iff
  - $A \leq B$ and $=_{\mathrm{A}}$ is the same as $(=_{\mathrm{B}})$
- Some examples:

(14) $\textsc{Man} \sqsubseteq \textsc{Human}$

(15) $(RTable, =_t) \sqsubseteq (Table, = t)$
where $RTable$ is: $\Sigma x{:}Table.red(x)$ is the domain of red tables and $=_t$ is the equivalence relation representing the identity criterion for tables

CLASP centre for linguistic theory and studies in probability

CNs as Setoids: Subsetoids

- In restricted domains like *Man* or *RTable*, the identity criteria coincide with those in *Human* and *Table*
- For these cases, one can ignore the IC, i.e. one can use the simpler CNs-as-Types approach
  - More sophisticated cases like copredication with quantification however need IC

(16) John picked up and mastered three books.

- Double distinctness

(17) John picked up and mastered three books $\Rightarrow$ John picked up three physical objects and mastered three informational objects

centre for linguistic theory and studies in probability

CNs as Setoids: Copredication

- Let us split the example into its conjuncts

(18) $Three(Book, \textsc{Phy}, pick\ up(j))$.

(19) $Three(Book, \textsc{Info}, master(j))$.

- Note: the CN *book* in 18 refers to a different collection from that referred to by *book* in 19

(20) $\textsc{Book}_1 = (Book, =_p)$

(21) $\textsc{Book}_2 = (Book, =_i)$

CNs as Setoids: Copredication

- How the identity criterion for *books* is determined
  - why do we use $=_p$ in 18 and $=_i$ in 19?
    - ★ The verb (and its semantics) determines the identity criterion of the object CN.

$$IC^{N,V} \Rightarrow \begin{cases} =_p & \text{if Dom(V)} = \textsc{Phy} \\ =_i & \text{if Dom(V)} = \textsc{Info} \\ ??? & \text{if Dom(V)} = \textsc{Phy} \bullet \textsc{Info} \end{cases}$$

    - ★ In order to deal with the dot-type case, we have to define setoids for dot-types!

CLASP centre for linguistic theory and studies in probability

CNs as Setoids: Copredication

- Let $\mathrm{A} = (A, =_{\mathrm{A}})$ and $\mathrm{B} = (B, =_{\mathrm{B}})$ be setoids. Then, the *dot-setoid* $\mathrm{A} \bullet \mathrm{B}$ is defined as follows:
  - $\mathrm{A} \bullet \mathrm{B} = (A \bullet B,\ =_{\mathrm{A} \bullet \mathrm{B}})$
    where $\langle a_1, b_1 \rangle =_{\mathrm{A} \bullet \mathrm{B}} \langle a_2, b_2 \rangle$ if, and only if, $(a_1 =_{\mathrm{A} a_2}) \vee (b_1 =_{\mathrm{B}} b_2)$.

CLASP centre for linguistic theory and studies in probability

CNs as Setoids: Copredication

- The semantics for *three*
  Let A be a type and $\mathrm{B} = (B, =_{\mathrm{B}})$ a setoid such that $A \leq B$, and $P : B \to Prop$ a predicate over B:
- $Three(A, \mathrm{B}, P) = \exists x, y, z : A.\ D[\mathrm{B}](x, y, z)\ \wedge P(x)\ \wedge P(y)\ \wedge P(z)$.
  where $D[\mathrm{B}](x, y, z) = x \neq_{\mathrm{B}} y\ \wedge y \neq_{\mathrm{B}} z\ \wedge x \neq_{\mathrm{B}} z$.

CNs as Setoids: Copredication

- With these definitions, the desired semantics of our copredication cases are derived
- $\mathit{Three}(\mathit{Book}, \textsc{Phy} \bullet \textsc{Info}, \mathit{pm}(j))$
- $\exists x, y, z : \mathit{Book}.D[\textsc{Phy}](x, y, z)\ \&\ D[\textsc{Info}](x, y, z)\ \&\ \mathit{pm}(j, x)\ \&\ \mathit{pm}(j, y)\ \&\ \mathit{pm}(j, z)$
  - Note that this is achieved through defining the equivalence relation for dot-types by means of disjunction of both identity criteria and, then, we obtain double distinctness by negating the disjunction.

CNs as Setoids: Copredication

- Verbs plus adjectives in quantified copredication
  - Consider the following example:

    (22) John mastered three heavy books.

  - The interpretation needed here: John mastered three informational objects that are also heavy as physical objects
    - ★ Both the verb and the adjective have a word on the IC
- First step: adjectival modification
  - $HBook = \Sigma(Book, heavy)$ or $\Sigma x{:}Book.heavy(x)$

CNs as Setoids: Copredication

- The interpretation we get:

  $Three(HBook, \textsc{Phy} \bullet \textsc{Info}, master(j))$
- Expanding:

  $\exists x, y, z : HBook.D[\textsc{Phy}](x, y, z)\ \&\ D[\textsc{Info}](x, y, z)$
  $\&\ master(j, x)\ \&\ master(j, y)\ \&\ master(j, z)$

CLASP centre for linguistic theory and studies in probability

Gotham 2016

- If we want to be precise, Gotham's account [Gotham(2016)] is not Montagovian *per se*, since the underlying logic is not set theory but mereology
  - However, mereological accounts within the mainstream Montagovian tradition have been quite common from [Link(1983)] onwards
  - We thus consider such accounts within the Montagovian general tradition

Gotham 2016: The key components of the account

- Both complex and plural objects exist
- Complex objects are denoted with with the $+$ operator while plural objects with the $\oplus$ operator (with assumes a join semilattice ála [Link(1983)]
  - For example, *book* denotes the set of composite objects p + i, where p is a physical book and i is an informational book
  - Gotham further assumes that any property that holds of p holds of $p + i$, and likewise any property that holds of i also holds of $p + i$.
    - $\forall P.P(p) \rightarrow P(p + i)$ and $\forall P.P(i) \rightarrow P(p + i)$
- Important to note: Plural objects exist in Link's system but not complex objects!

Gotham 2016: The key components of the account

- Lexical entries are more complex than what is traditionally assumed in the sense that besides deciding extensions, they further specify a distinctness criterion
- First proposal for the type of common nouns: $e \to (t \times \mathbf{R})$
  - $(t \times R)$ is a product type, with t a truth value and R the type of relations, basically an abbreviation for $e \to e \to t$.
- Final proposal: $e \to (t \times ((e \to \mathbf{R}) \to t))$ (abbreviated to $e \to T$)
  - the second projection π_2 is a set of functions that map the type e argument to a relation of type $e \to \mathbf{R}$
    - ★ where $R \sqsubseteq R_{cn}$, and R_{cn} is taken to be the individuation relation given by the noun or the predicate.

CLASP centre for linguistic theory and studies in probability

Gotham 2016: The key components of the account

- Common nouns involve an individuation relation in their lexical entries.
  - These are equivalences for different objects exist, for example physical, informational etc.
    1. $\mathrm{PHY} = \lambda x, y : e.\mathit{phys\text{-}equiv}(x)(y)$
    2. $\mathrm{INFO} = \lambda x, y : e.\mathit{info\text{-}equivequiv}(x)(y)$
- Lexical entry for *book*:
  $\lambda y : e.^{*}\mathit{book}(y), \lambda f : e \to R.f(y) \sqsubseteq (\mathrm{PHYS} \sqcup \mathrm{INFO})$

CLASP centre for linguistic theory and studies in probability

Gotham 2016: The key components of the account

- A non-compressibility statement denoting that no two members of a plurality stand to a relation R (see [Gotham(2014)] for the formal definition of compressibility)
- A further assumption that verbs also somehow point to the individuation criteria that have to be used
  1. For this, a form of generalized implication is used
- Lastly an Ω operator is used that helps in choosing the individuation criteria for each of the arguments
  - Ω computes the least upper bound of the set of relations R.
    - E.g., in the case of *books* this is $(\mathrm{PHYS} \sqcup \mathrm{INFO})$, for *master* it is (*INFO*) and so on.

CLASP centre for linguistic theory and studies in probability

Comparing the Approaches

- Common claim in both: common nouns need to be extended with the addition of individuation criteria
  - Gotham extends the traditional notion of predicates as sets to a considerably more complex type ($e \to (t \times ((e \to \mathbf{R}) \to t)))$
  - We claim that CNs are not just types anymore, but rather setoids with their second component being an equivalence relation on the type $((A, =))$ (Gotham's R)

Comparing the Approaches

- Deciding the IC for verbs or adjectives
  - Gotham uses the Ω function
  - For our account, the IC criteria for the arguments of the verb are decided by the type of the argument and whether it is a complex object or not

Comparing the Approaches

- Potential problem?
  - Gotham uses a mereological account to capture the individuation criteria
    - ★ makes an additional crucial, non-trivial assumption: the introduction of the + operator and the axiom $P(a) \Rightarrow P(a+b) \wedge P(b+a)$, for any predicate P
    - ★ Crucial for the account to work
    - ★ Plays the role of the definition we have for dot-setoids.
    - ★ One can question its naturalness, as [Gotham(2016)] seems to be doing:

One can imagine other ways of implementing this mereological approach to , both in terms of what composite objects there are and in terms of what their properties are. That in turn would require a revision to the definition of compressibility in Section 3.1.1. The approach adopted in this article is adequate for criteria of individuation to be determined compositionally, and can be adapted if revisions of this kind turn out to be necessary [GOT 17: fn2]

CLASP centre for linguistic theory and studies in probability

Comparing the Approaches

- Our account and introduction of follows the standard way of forming types in MTTs, i.e. via formation/introduction/elimination/computational rules. This seems to be an advantage of our approach
- Most importantly, a more serious, formal flaw
  - Assume the following predicate P:
    $P(p) = \text{true if } p = a + x$, $P(p) = \text{false otherwise}$
  - Then, $P(a)$ is false and $P(a + x)$ is true and thus P does not satisfy Gotham's axiom
  - We do not know how serious of a a flaw this is and whether a trivial fixing is available.

Z. Luo.
Type-theoretical semantics with coercive subtyping.
Semantics and Linguistic Theory 20 (SALT20), Vancouver, 2010.

T. Xue and Z. Luo.
Dot-types and their implementation.
Logical Aspects of Computational Linguistics (LACL 2012). LNCS 7351, 2012.

P.T. Geach.
Reference and Generality: An examination of some Medieval and Modern Theories.
Cornell University Press, 1962.

Matthew Gotham.
Composing criteria of individuation in copredication.
Journal of Semantics, 34(2):333–371, 2016.

G. Link.
The logical analysis of plurals and mass terms: A lattice-theoretical approach.

In Schwarze C. Bauerle R. and von Stechow A, editors, *Meaning, Use an Interpretation of Language*, pages 302–323. Mouton De Gruyter, Berlin, 1983.

M. Gotham.
Copredication, quantification and individuationy.
PhD thesis, University College London, 2014.

CLASP centre for linguistic theory and studies in probability

From Montague Semantics to MTT-Semantics: A Meaningful Comparison

Stergios Chatzikyriakidis and Zhaohui Luo

(ESSLLI 2019: Lecture 5 by ZL)

Dependent Types in Event Semantics

- What is a dependent type?
  - An example of its use in linguistic semantics, involving:
    1. Refined (dependent) types of events Evt(h)
    2. Parameterised coercion (coercion whose type is Π-type)
- Dependent event types (Luo & Soloviev, WoLLIC 2017)
  - DETs in the Montagovian/Davidsonian setting
    - Do this for the sake of easier understanding (and further applications).
    - You can add DETs to MTTs (MTT-event semantics) – omitted here.
  - Event quantification problem and its solution with DETs
  - Formal extension of simple type theory by DETs (and its meta-theoretic properties)

What is a dependent type?

- Dependent type
  - A type that depends on objects.
    - Vect(n) – type of lists $[a_1, \ldots, a_n]$ whose length is n
    - Child(h) – type of children of h (h : Human)
    - Evt(h) – type of event whose agent is h
  - Note: a dependent type is *not* a type depending on types.
    - Pred(A) = A→Prop depends on type A, but it is not a dependent type.

An example

- Example:

Julie just started *War and Peace*, which Tolstoy finished after many years of hard work. But that won't last because she never gets through long novels.

- Formal semantics?
  - This example in (Asher & Luo 2012) was the first use of parameterised coercions for linguistic semantics in literature.
- Three issues
  - Linguistic coercions
  - Multiple coercions (for "start/finish/last W&P"): which one?
    - start W&P → start <u>writing</u>/<u>reading</u> W&P

Linguistic coercions

- Representing linguistic coercions
  - By coercions in the framework of coercive subtyping
- Consider

  Julie enjoyed a book.

  $\exists x : Book.\ enjoy(j, x)$
- Is the formula well-typed? Well, only yes if:

  $Book \leq_{reading} Event,$

  Julie enjoyed reading a book.

  $\exists x : Book.\ enjoy(j, reading(x))$
- Coercion "reading" is of type Book→Event.

Multiple coercions

- The following two need <u>different</u> coercions:
  - Julie started W&P. (start W&P → start reading W&P)
  - Tolstoy finished W&P. (finish W&P → finish writing W&P)
- Coercion scopes interleave & overlap, for example, in the above paragraph,
  - Coercions "reading" (for start/last) and "writing" (for finish) have interleaving and overlapping scopes.
- How to deal with this?
  - Use parameterised coercions, those with parameters quantified by Π.

- Assume:

$$start,\ finish,\ last : \Pi h : Human.\ (Evt(h) \rightarrow Prop)$$
$$read,\ write : \Pi h : Human.\ (Book \rightarrow Evt(h))$$

- Consider coercions
  - c : Πh:Human. Book→Evt$_A$(h)

$$Book \leq_{c(h)} Evt_A(h) \text{ for } h : Human.$$

$$c(h,b) = \begin{cases} write(h,b) & \text{if } h \text{ wrote } b, \\ read(h,b) & \text{otherwise.} \end{cases}$$

Julie just started *War and Peace*, which Tolstoy finished after many years of hard work. But that won't last because she never gets through long novels.

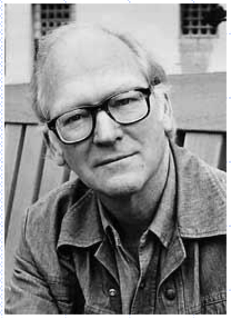
$$\begin{aligned}&start(j, W\&P)\\ &\&\ finish(t, W\&P)\\ &\&\ \neg last(j, W\&P)\\ &\&\ \forall lb : (\Sigma b{:}Book.long(b)).\ finish(j, \pi_1(lb))\end{aligned}$$

$$\begin{aligned}&start(j, c(j, W\&P))\\ &\&\ finish(t, c(t, W\&P))\\ &\&\ \neg last(j, c(j, W\&P))\\ &\&\ \forall lb : (\Sigma b{:}Book.long(b)).\ finish(j, c(j, \pi_1(lb)))\end{aligned}$$

$$\begin{aligned}&start(j, read(j, W\&P))\\ &\&\ finish(t, write(t, W\&P))\\ &\&\ \neg last(j, read(j, W\&P))\\ &\&\ \forall lb : (\Sigma b{:}Book.long(b)).\ finish(j, c(j, \pi_1(lb)))\end{aligned}$$

- A detailed analysis of one of the formulas:
  - start(j, W&P) — start(j) : Evt(j)→Prop
    = start(j, c(j,W&P)) — W&P : Book $<_{c(j)}$ Evt(j)
    = start(j, reading(W&P)) — by defn of c(j)
  - Similarly, eg, finish(t, W&P) = finish(t,writing(&P)).
- Therefore, the semantics is correct as intended.

Davidsonian event semantics



- Original motivation: adverbial modifications

  (1) John buttered the toast.

  (2) John buttered the toast with the knife in the kitchen.

  Do we have (2) ⇒ (1)?

  - Cumbersome in MG with meaning postulates (next slide)
  - Davidson (1967): verbs tacitly introduce existentially quantified events, doing away with meaning postulates.

- In neo-Davidsonian notation (1980s) with thematic roles (slide)

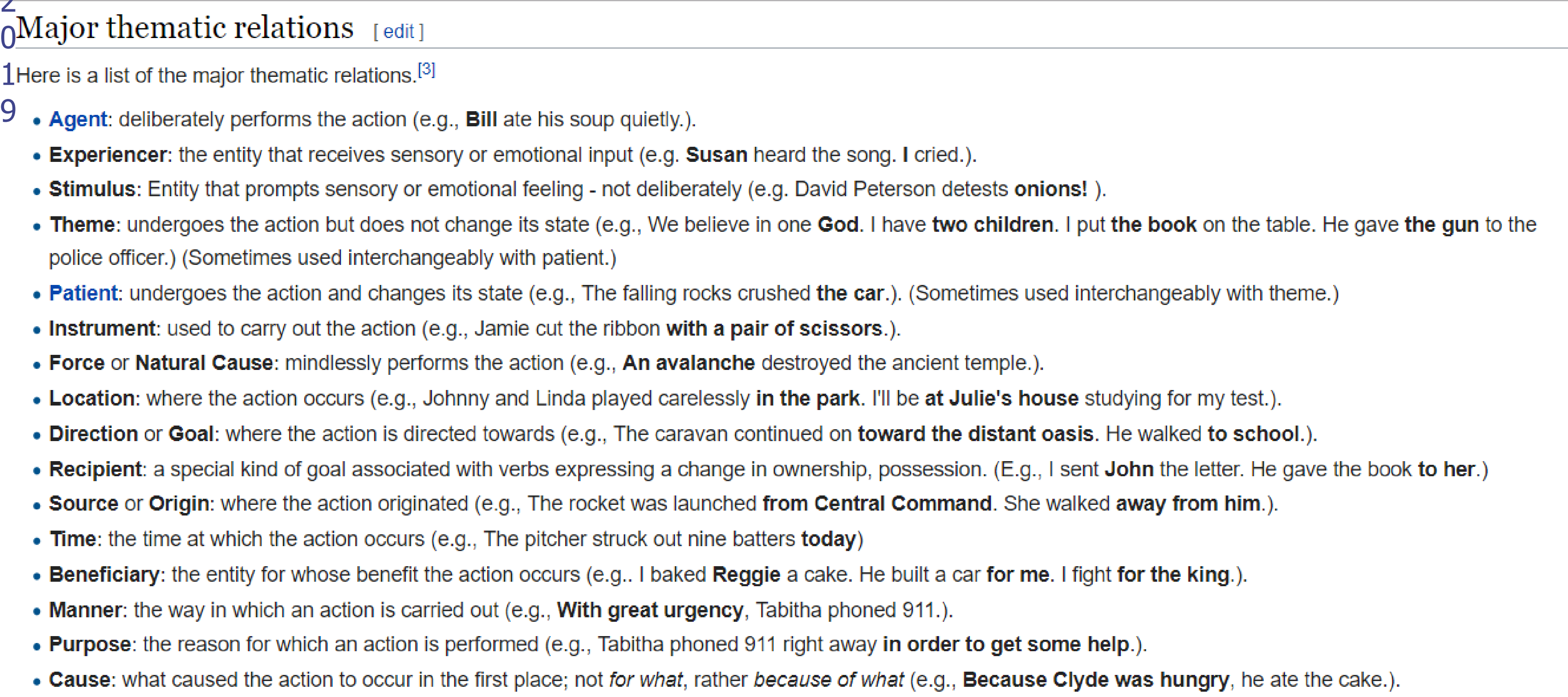
  (1′) ∃e:Event. butter(e)
  & agent(e)=john & patient(e)=toast

  (2′) ∃e:Event. butter(e) & with(e,knife) & at(e,kitchen)
  & agent(e)=john & patient(e)=toast

  Obviously, (2′) ⇒ (1′)

MG approaches without events

- (1) John buttered the toast.
  (1s) butter(john,toast), where butter : $\mathbf{e}^2 \rightarrow \mathbf{t}$.
- (2) John buttered the toast with the knife in the kitchen.
  (2s) butter(j,t,k,m), where butter : $\mathbf{e}^4 \rightarrow \mathbf{t}$
  (2t) kitchen(knife(butter(john)))(toast),
  where butter : $\mathbf{e}^2 \rightarrow \mathbf{t}$, knife/kitchen : $(\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t})$
- Both need <u>meaning postulates</u> to get, eg,
  (2s) $\Rightarrow$ (1s)
  (2t) $\Rightarrow$ (1s)
  rather ad hoc.



Major thematic relations [edit]

Here is a list of the major thematic relations.[3]

- Agent: deliberately performs the action (e.g., **Bill** ate his soup quietly.).
- **Experiencer**: the entity that receives sensory or emotional input (e.g. **Susan** heard the song. **I** cried.).
- **Stimulus**: Entity that prompts sensory or emotional feeling - not deliberately (e.g. David Peterson detests **onions!**).
- **Theme**: undergoes the action but does not change its state (e.g., We believe in one **God**. I have **two children**. I put **the book** on the table. He gave **the gun** to the police officer.) (Sometimes used interchangeably with patient.)
- Patient: undergoes the action and changes its state (e.g., The falling rocks crushed **the car**.). (Sometimes used interchangeably with theme.)
- **Instrument**: used to carry out the action (e.g., Jamie cut the ribbon **with a pair of scissors**.).
- **Force** or **Natural Cause**: mindlessly performs the action (e.g., **An avalanche** destroyed the ancient temple.).
- **Location**: where the action occurs (e.g., Johnny and Linda played carelessly **in the park**. I'll be **at Julie's house** studying for my test.).
- **Direction** or **Goal**: where the action is directed towards (e.g., The caravan continued on **toward the distant oasis**. He walked **to school**.).
- **Recipient**: a special kind of goal associated with verbs expressing a change in ownership, possession. (E.g., I sent **John** the letter. He gave the book **to her**.)
- **Source** or **Origin**: where the action originated (e.g., The rocket was launched **from Central Command**. She walked **away from him**.).
- **Time**: the time at which the action occurs (e.g., The pitcher struck out nine batters **today**)
- **Beneficiary**: the entity for whose benefit the action occurs (e.g.. I baked **Reggie** a cake. He built a car **for me**. I fight **for the king**.).
- **Manner**: the way in which an action is carried out (e.g., **With great urgency**, Tabitha phoned 911.).
- **Purpose**: the reason for which an action is performed (e.g., Tabitha phoned 911 right away **in order to get some help**.).
- **Cause**: what caused the action to occur in the first place; not *for what*, rather *because of what* (e.g., **Because Clyde was hungry**, he ate the cake.).

12

Problems in Event-semantics + Montague

- For example, "event quantification problem" (EQP)
- Incompatibility between event semantics and MG.

(1) Nobody talked.

Intended neo-Davidsonian event semantics is (2):

(2) $\neg\exists x{:}\mathbf{e}.\ human(x)\ \&\ \exists v{:}Event.\ talk(v)\ \&\ agent(v,x)$

But the incorrect semantics (3) is also possible – it is well-typed:

(3) $\exists v{:}Event.\ \neg\exists x{:}\mathbf{e}.\ human(x)\ \&\ talk(v)\ \&\ agent(v,x)$

which moves the event quantifier "$\exists v{:}Event$" in (2) to the left.

Some proposed solutions to EQP

- Many different proposals (only mentioning two below)
  - Purpose: to force scope of event quantifier to be narrower.
- Champollion's quantificational event sem. [2010, 2015]
  - Trick: taking a set E of events as argument, but **talk**(e) …
    - talk : (Event→**t**)→**t** with talk(E) = ∃e:Event. e∈E & **talk**(e)
  - Debatable: intuitive meanings, compositionality & complexity
- Winter-Zwarts [2011] & de Groote [2014]
  - Use Abstract Categorial Grammar (see, eg, [de Groote 01])
    - ACG structure prevents incorrect interpretation.
  - Seemingly coincidental (and what if one does not use ACG?)
- Our proposal: dependent event types (solution to EQP & …)

Dependent event types [Luo & Soloviev (WoLLIC17)]

- Dependent event types
  - Refining event structure by (dependent) typing
  - Applications include
    - A solution to EQP
    - Selection restrictions in MTT-semantics with events
- How:

  Refining event structure:

  Event ➔ Evt(a)/Evt(a,p)

  which are event types dependent on thematic roles a/p, called agents/patients, respectively.

DETs and their subtyping relationships

- For a:Agent and p:Patient, consider DETs

  Event, $Evt_A(a)$, $Evt_P(p)$, $Evt_{AP}(a,p)$

- Subsumptive subtyping

$$\frac{a : A \quad A \leq B}{a : B}$$

- Subtyping between DETs (eg, Any event with agent a and patient p is an event with agent a.)

$$\begin{tikzcd}
 & Evt_A(a) \arrow[dr, dash, "\leq"] & \\
Evt_{AP}(a,p) \arrow[ur, dash, "\leq"] \arrow[dr, dash, "\leq"'] & & Event \\
 & Evt_P(p) \arrow[ur, dash, "\leq"'] &
\end{tikzcd}$$

Two systems with DETs

- Extension of Montague's simple TT with DETs
  - C_e extends Church's STT (1940) with DETs
  - Montague's system is familiar for many – hopefully better understanding of DETs.
- Extension of modern type theories with DETs
  - T[E] extends type theory T with DETs (e.g., T = UTT).
  - This shows how DETs work in MTTs.

(Here, we only present C_e, for the sake of easier understanding and applications.)

Dependent event types in Montagovian setting

- Eg. John talked loudly.
  - talk, loud : Event→**t**
  - agent : Event→**e**→**t**
- (neo-)Davidsonian event semantics

$$\exists e : Event.\ talk(e)\ \&\ loud(e)\ \&\ agent(e, j)$$

- Dependent event types in Montagovian setting:

$$\exists e : Evt_A(j).\ talk(e)\ \&\ loud(e)$$

which is well-typed because $Evt_A(j) \leq Event$.

C_e: extending Church's simple TT with DETs

- First, Church's simple type theory (1940)
  - Employed in Montague's semantics (c.f., Gallin 1975)
  - Its rules are presented in the Natural Deduction style as follows.

- Rules for sorts/judgements and λ-calculus

$$\frac{}{\mathbf{e}\ \mathit{type}} \qquad \frac{}{\mathbf{t}\ \mathit{type}} \qquad \frac{}{x : A\ [x : A]} \qquad \frac{}{P\ \mathit{true}\ [P\ \mathit{true}]}$$

$$\frac{A\ \mathit{type} \quad B\ \mathit{type}}{A \to B\ \mathit{type}} \qquad \frac{b : B\ [x : A] \quad x \notin FV(B)}{\lambda x{:}A.b : A \to B} \qquad \frac{f : A \to B \quad a : A}{f(a) : B}$$

Note: the side condition in the λ-rule is there only for DETs.

- Rules for truth of logical formulas

$$\frac{P : \mathbf{t} \quad Q : \mathbf{t}}{P \supset Q : \mathbf{t}} \qquad \frac{Q\ \mathit{true}\ [P\ \mathit{true}]}{P \supset Q\ \mathit{true}} \qquad \frac{P \supset Q\ \mathit{true} \quad P\ \mathit{true}}{Q\ \mathit{true}}$$

$$\frac{A\ \mathit{type} \quad P : \mathbf{t}\ [x : A]}{\forall(A, x.P) : \mathbf{t}} \qquad \frac{P\ \mathit{true}\ [x : A]}{\forall(A, x.P)\ \mathit{true}} \qquad \frac{\forall(A, x.P[x])\ \mathit{true} \quad a : A}{P[a]\ \mathit{true}}$$

- Rule for "conversion" of logical formulas (λ-conversion omitted)

$$\frac{P\ \mathit{true} \quad Q : \mathbf{t}}{Q\ \mathit{true}} \quad (P \simeq Q)$$

Dependent event types in C_e

$$\frac{}{Agent\ type} \qquad \frac{}{Patient\ type}$$

$$\frac{}{Event\ type} \qquad \frac{a : Agent}{Evt_A(a)\ type} \qquad \frac{p : Patient}{Evt_P(p)\ type} \qquad \frac{a : Agent \quad p : Patient}{Evt_{AP}(a,p)\ type}$$

$$\frac{a : Agent \quad p : Patient}{Evt_{AP}(a,p) \leq Evt_A(a)} \qquad \frac{a : Agent \quad p : Patient}{Evt_{AP}(a,p) \leq Evt_P(p)} \qquad \frac{a : Agent}{Evt_A(a) \leq Event} \qquad \frac{p : Patient}{Evt_P(p) \leq Event}$$

$$\frac{A\ type}{A \leq A} \qquad \frac{A \leq B \quad B \leq C}{A \leq C} \qquad \frac{A' \leq A \quad B \leq B'}{A \to B \leq A' \to B'}$$

$$\frac{A \simeq B}{A \leq B} \qquad \frac{a : A \quad A \leq B}{a : B}$$

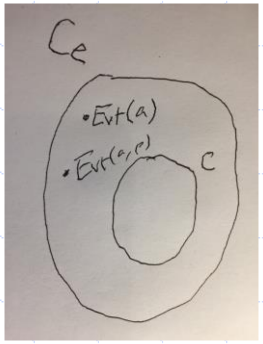
$$\begin{tikzcd}
 & Evt_A(a) \arrow[dr, dash, "\leq"] & \\
Evt_{AP}(a,p) \arrow[ur, dash, "\leq"] \arrow[dr, dash, "\leq"'] & & Event \\
 & Evt_P(p) \arrow[ur, dash, "\leq"'] &
\end{tikzcd}$$

Conservativity

Background notes

(1) Conservative extension: "J in C and |- J in C_e, then |- J in C."

(2) Logical consistency is preserved by conservative extensions.



Theorem. C_e is a conservative extension over Church's simple type theory.

- Proof.
  - Define R : $C_e \rightarrow C$ that preserves derivations.
    - R maps Evt(…) to Event and Agent/Patient to **e**.
    - R(t)=t for $t \in C$.
  - For any C_e-derivation D, R(D) is a C-derivation.

Corollary. C_e is logically consistent.

- Remark
  - Why meta-theory?
  - Any extension of an existing logical system need be shown to be "OK" (eg, still logically consistent).
  - Logical consistency is a most basic requirement for a foundational semantic language (either directly or indirectly).
- TTR, "Type Theory with Records", is not a type theory as usually understood: it is a set-theoretical system of notations.

DET-solution to EQP

(1) Nobody talked.

Neo-Davidsonian in Montague's setting (repeated):

(2) $\neg\exists x{:}\mathbf{e}.\ \mathrm{human}(x)\ \&\ \exists v{:}\mathrm{Event}.\ \mathrm{talk}(v)\ \&\ \mathrm{agent}(v,x)$

(3) $\exists v{:}\mathrm{Event}.\ \neg\exists x{:}\mathbf{e}.\ \mathrm{human}(x)\ \&\ \mathrm{talk}(v)\ \&\ \mathrm{agent}(v,x)$

The incorrect (3) is well-typed.

Dependent event types in Montague's setting:

(4) $\neg\exists x{:}\mathbf{e}.\ \mathrm{human}(x)\ \&\ \exists v{:}\mathrm{Evt}_A(x).\ \mathrm{talk}(v)$

(#) $\exists v{:}\mathrm{Evt}_A(x).\ \neg\exists x{:}\mathbf{e}.\ \mathrm{human}(x)\ \&\ \mathrm{talk}(v)$

where (#) is ill-typed since the first "x" is outside scope of "$\exists x{:}\mathbf{e}$".

Selectional restrictions with events

- (#) Tables talk.
  - Montague: $\forall x{:}\mathbf{e}.talk(x)$ – well-typed but false (talk : $\mathbf{e}\rightarrow\mathbf{t}$)
  - MTT-sem: $\forall x{:}Table.talk(x)$ – ill-typed (talk : Human$\rightarrow$Prop)
- What happens when we have events? (talk : Event$\rightarrow\mathbf{t}$/Prop)
  - Montague: $\forall x{:}\mathbf{e}\ \exists v{:}Event.\ talk(v)\ \&\ agent(v)=x$ (well-typed)
  - MTT-sem: $\forall x{:}Table\ \exists v{:}Evt_A(x).\ talk(v)$

    where we have Table $\leq$ Agent. (Also well-typed!)

So? There are three approaches to enforce selectional restriction with events:

1. Refining typing for verb phrases (like talk)
2. Refining typing of thematic roles (like agent)
3. Further refining dependent event types by subtyping

- Approach 1: Instead of (neo-Davidsonian) talk : Event→**t**,
  - talk_h : Human→Event→Prop (Davidson's original proposal), or
  - talk_d : ∏h:Human. $\text{Evt}_A(h)$→Prop (dependent typing)

  Then, "Tables talk" is ill-typed – table x is not a human:
  - (#) ∀x:Table ∃v:Event. $\text{talk}_h(x,v)$ & agent(v)=x
  - (#) ∀x:Table ∃v:$\text{Evt}_A(x)$. $\text{talk}_d(x,v)$
- Approach 2: Instead of (neo-Davidsonian) agent:Event→**e**,
  - agent_h : Event→Human (with codomain being Human)

  Then, "Tables talk" is ill-typed – table x is not a human:
  - (#) ∀x:Table ∃v:Event. talk(v) & $\text{agent}_h(v)$=x

- Approach 3: refined DETs
  - Let $T \leq_c$ Agent. (Consider subtypes of Agent, wlg.)
    - $Evt_A[T] : T \rightarrow Type$
    - $Evt_A[T](a) = Evt_A(c(a))$, for any $a : T$.
- Examples
  - Men talk.
    - $\forall x{:}Man\ \exists v{:}Evt_A[Human](x).\ talk(v)$ (OK because $Man \leq Human$)
  - Tables talk.
    - (#) $\forall x{:}Table\ \exists v{:}Evt_A[Human](x).\ talk(v)$ (ill-typed - x is not a human.)
  - John picked up and mastered the book.
    - $\exists v{:}Evt_{AP}[Human, P \bullet I](j,b).\ pick\text{-}up(v) \& master(v)$, where $b : Book \leq P \bullet I$.
- Note: this approach is more flexible/powerful.