Logical Approaches to (Compositional) Natural Language Semantics

Kristina Liefke, MCMP

Summer School on Mathematical Philosophy for Female Students

LMU Munich, July 28, 2014

This Session & The Summer School

- Many lectures/tutorials have presupposed the possibility of translating natural language sentences into interpretable logical formulas.
- But: This translation procedure has not been made explicit.

1. In this session, we introduce a procedure for the translation of natural language, which is inspired by the work of Montague:

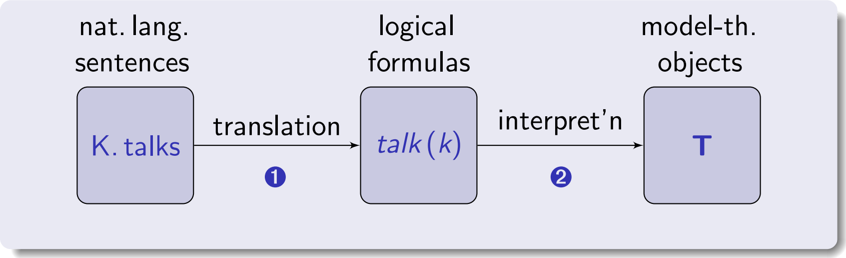
   Kristina talks about Montague.

   Kristina $\rightsquigarrow$ *k* Montague $\rightsquigarrow$ *m* talk $\rightsquigarrow$ *talk* about $\rightsquigarrow$ *about*

   Kristina talks about Montague $\rightsquigarrow$ $about(m, talk, k)$

2. We will then use this procedure to provide a (formal) semantics for natural language. ➥Montague Grammar

The (Rough) Plan



Compositional Semantics

We will be concerned with compositional – not lexical – semantics:

Lexical semantics studies the meaning of individual words:

$[\![\text{talk}]\!]$:= "to convey or express ideas, thought, information etc. by means of speech"

➥ Compositional semantics studies the way in which complex phrases obtain a meaning from their constituents:

$[\![\text{Kristina}]\!] = [\![k]\!]$ $[\![\text{Montague}]\!] = [\![m]\!]$ $[\![\text{talk}]\!] = [\![talk]\!]$ $[\![\text{about}]\!] = [\![about]\!]$
$[\![\text{Kristina talks about Montague}]\!] = [\![about(m, talk, k)]\!]$

Principle (Semantic compositionality) (Partee, 1984)

The meaning of an expression is a function of the meanings of its constituents and their mode of combination.

Compositional Semantics

We will be concerned with compositional – not lexical – semantics:

Lexical semantics studies the meaning of individual words.

➥ Compositional semantics studies the way in which complex phrases obtain a meaning from their constituents:

Montague:

$[\![\text{Kristina}]\!] = [\![\text{k}']\!]$ $[\![\text{Montague}]\!] = [\![\text{m}']\!]$ $[\![\text{talk}]\!] = [\![\text{talk}']\!]$ $[\![\text{about}]\!] = [\![\text{about}']\!]$

$[\![\text{Kristina talks about Montague}]\!] = [\![\text{about}'(\text{m}', \text{talk}', \text{k}')]\!]$

↩ 'Word-prime semantics' (Crouch and King, 2008), cf. (Carlson, 1977)

Principle (Semantic compositionality) (Partee, 1984)

The meaning of an expression is a function of the meanings of its constituents and their mode of combination.

Word-Prime Semantics

(Carlson, 1977, Foreword)

In the spring of 1976, Terry Parsons and Barbara Partee taught a course on Montague grammar, which I attended. On the second to the final day of class, Terry went around the room asking the students if there were any questions at all that remained unanswered, and promised to answer them on the last day of class. I asked if he really meant ANY question at all, which he emphatically said that he meant. As I had encountered a few questions in my lifetime that remained at least partially unresolved, I decided to ask one of them. What is life? What is the meaning of life? After all, Barbara and Terry had promised to provide answers to any question at all.

Word-Prime Semantics

(Carlson, 1977, Foreword)

On the final day of class Barbara wore her Montague grammar T-shirt, and she and Terry busied themselves answering our questions. At long last, they came to my question. I anticipated a protracted and involved answer, but their reply was crisp and succinct. First Barbara, chalk in hand, showed me the meaning of life.

$$^{\wedge}\mathbf{life}'$$

Terry then stepped up and showed me what life really is.

$$^{\vee\wedge}\mathbf{life}'$$

As we were asked to show on a homework assignment earlier in the year, this is equivalent to: $\mathbf{life}'$.

Leaving me astounded that I had been living in such darkness for all these years, the class then turned to the much stickier problem of pronouns.

Why Do Compositional Semantics?

1. Explain the productivity and systematicity of linguistic understanding:

   You understand (1) even if you have not come across it before:

   (1) Penny Maddy has agreed to have her picture posted on the door of Room 1205.

2. Obtain objects (here: formulas) to which we can apply our formal techniques. (These formulas are free of ambiguities).

   Only (2b), not (2a), can be analyzed in epistemic logic:

   (2) a. Mary knows that Penny is a mathematical philosopher.
   b. $K(\textit{mathematical-philosopher}(\textit{penny}), \textit{mary})$

   $\Rightarrow \textit{mathematical-philosopher}(\textit{penny})$ (by **T**)

   $\Rightarrow K(K(\textit{math.-philosopher}(\textit{penny}), \textit{mary}), \textit{mary})$ (by **4**)

Why Do Compositional Semantics?

3. Evaluate the truth or falsity of natural language sentences (via the truth/falsity of their translating formulas):

> 'Penny is a philosopher' is **true** (or **false**) in M under g
> iff $\models_{M,g}$ $\mathit{philosopher}(\mathit{penny})$ (resp. $\dashv_{M,g}$ $\mathit{philosopher}(\mathit{penny})$)

4. Predict the relation of entailment/equivalence betw. sentences (via the entailment relation between their transl. formulas):

> 'Penny is a philosopher' **entails** 'Philosophers exist' in M, g
> iff $\models_{M,g}$ $\mathit{philosopher}(\mathit{penny}) \Rightarrow (\exists x.\mathit{philosopher}(x))$

5. Explain speakers' judgements about consistency, presupposition, anaphoric relations, etc.

The (Concrete) Plan

1. Why a logical approach to NL semantics?
2. A challenge for this approach: the logic/language mismatch
3. Montague's solution . . . : typed lambda logic
4. Applying Montague's solution: extensional (formal) semantics
5. Montagovian extensions: intensional semantics

+ Non-Montagovian extensions: hyperintens'l & situation sem
+ My work on formal semantics: robustness & minimal models

6. Conclusion

A (Historical) Challenge

Frege, Russell, etc. assume the translation of natural language (NL) sentences into formulas of first-order predicate logic.

This translation faces two challenges:

Underlying problem Many NL expressions have a different syntactic form than their predicate-logical translations:

grammatical form $\neq$ logical form

➥ The language-to-logic translation is not fully compositional:

Principle (Compositionality of translations)

The (logical) translation of an expression is a function of the translations of its constituents and their mode of combination.

↩ This significantly reduces the utility of our logical translation.

A (Historical) Challenge

This translation faces two challenges:

1. Many logical translations have different 'constituents' than their associated NL sentences:

Translations of **simple sentences** often allow a division into logical subject (i.e. argument) and logical predicate (i.e. function):

(3) a. $_{\text{NP}}$Penny $_{\text{VP}}$is a philosopher b. $_{\text{fct}}$*philosopher* $_{\text{arg}}$(*penny*)

Translations of **complex sentences** often defy this division:

(4) a. $_{\text{NP}}$Mary $_{\text{VP}}$knows that Penny is a philosopher.

b. $_{\text{fct'}}$*K* (*philosopher* (*penny*), $_{\text{arg}}$*mary* $_{\text{fct''}}$)

The Logic/Language Mismatch

2. Some constituents of NL sentences do not have corresponding contiguous parts in the sentences' translating logical formulas (Russell, 1905):

Some elements of grammatical form **correspond with contiguous parts** of logical form:

(3) a. Penny is a philosopher   b. *philosopher* (*penny*)

Some elements of grammatical form **only correspond with non-contiguous parts** of logical form:

(5) a. A philosopher talks

b. $\exists x.$ *philosopher* $(x) \wedge$ *talk* (x)

↩ In (5b), the translation of a is spread over the entire formula!

Montague's Solution

(Montague, 1970a)

1. Replace first-order predicate logic by a **higher-order** logic with **lambda** abstraction ➥ typed lambda logic

2. Translate NL sentences into equivalents of their translating formulas from (4b) and (5b), which match the sentences' grammatical form:

Some elements of grammatical form **only correspond with <u>non-contiguous</u> parts** of logical form:

(5) a. A philosopher talks

b. $\exists x.\ \mathit{philosopher}\ (x) \wedge\ \mathit{talk}\ (x)$

$\Leftrightarrow \lambda P_1 \lambda P\, \exists x. P_1(x) \wedge P(x)\ (\mathit{philosopher})\ (\mathit{talk})$

a | philosopher | talks

a philosopher | talks

Matching Logic and Language (Montague, 1970a)

Richard Montague
(*1930, Stockton,
†1971, Los Angeles)

I reject the contention that an important theoretical difference exists between formal and natural languages.

(Montague, 1970a, p. 188)

There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed I consider it possible to comprehend the syntax and semantics of both kinds of languages within a single natural and mathematically precise theory.

(Montague, 1970b, p. 222)

Matching Logic and Language (Montague, 1970a)

Richard Montague
(*1930, Stockton,
†1971, Los Angeles)

I reject the contention that an important theoretical difference exists between formal and natural languages.

(Montague, 1970a, p. 188)

Montague's thesis Natural languages can be described as interpreted formal systems.

(Bach, 1986, p. 574)

Chomsky's thesis Natural languages can be described as formal systems.

(Bach, 1986, p. 574)

Montague's Solution: The λ-Calculus (Church, 1985)

- The λ-calculus is a theory of functions.
- λ is a binding operator (like ∃ or ∀) that
  - binds a variable (e.g. x), and
  - takes scope over expressions (e.g. A) which (typically) contain bound occurrences of this variable.

- $\lambda x.\, A$ denotes a function which, when applied to some argument d, returns the value of A with x interpreted as d:

$$[\![\lambda x.\, A]\!]^g \quad := \quad \{\langle d, [\![A]\!]^{g[d/x]}\rangle \mid d \in D\}\,,$$

where D is the range of x's values. ↩ λ-abstraction

- **Example λ-terms**
  - $\lambda x.\, \mathit{philosopher}(x)$
  - $\lambda P\, \exists x.\, \mathit{philosopher}(x) \wedge P(x)$
  - $\lambda P.\, P(\mathit{penny})$
  - $\lambda P_1 \lambda P\, \exists x.\, P_1(x) \wedge P(x)$

The λ-Calculus: Application (Church, 1985)

The dual of λ-abstraction: function application

$A(B)$ denotes the result of applying the denotation of A to the denotation of B:

$$[\![A(B)]\!]^g \quad := \quad [\![A]\!]^g([\![B]\!]^g).$$

Example terms

- $\lambda x.\, philosopher(x)\ (penny)$
- $\lambda P.\, P(penny)\ (philosopher)$
- $\lambda P\, \exists x.\, philosopher(x) \wedge P(x)\ (talk)$
- $\lambda P_1 \lambda P\, \exists x.\, P_1(x) \wedge P(x)\ (philosopher)$

The Rules of λ-Conversion (Church, 1985)

Application and abstraction are governed by three rules:

Rule 1: β-conversion (function application)

The substitution of all free occurrences of a variable in a formula with a suitable argument:

$$(\lambda x.A)(B) =_\beta A\{x := B\}, \text{ if } x \text{ is free for } B \text{ in } A$$

- $\lambda x.\, philosopher(x)\,(penny) \;=_\beta\; philosopher\,(penny)$
- $\lambda P_1 \lambda P\, \exists x. P_1(x) \wedge P(x)\,(philosopher)\,(talk)$
  $=_\beta \lambda P \exists x. (philosopher)(x) \wedge P(x)\,(talk)$
  $=_\beta \exists x. (philosopher)(x) \wedge (talk)(x)$

↩ Our translation of NL sentences will use many β-conversions.

The Rules of λ-Conversion (Church, 1985)

We will use α-conversions to avoid the obtaining of non-equivalent terms by 'variable collision':

Observe: $\lambda y \lambda x.\, find(y)(x)\; x \;=_\beta\; \lambda x.\, find\,(x)\,(x)$

But: ↝ 'find someone' ↝ 'find **oneself**'

Problem: The conversion is **not meaning-preserving!**

Solution: Use the α-equivalent of $\lambda y \lambda x.\, find(y)(x)$: $\lambda y \lambda \mathbf{z}.\, find(y)(\mathbf{z})$

$\lambda y \lambda \mathbf{z}.\, find(y)(\mathbf{z})\; x \;=_\beta\; \lambda x.\, find\,(x)\,(\mathbf{z})$

↝ 'find someone' ↝ 'find someone'

Rule 2: α-conversion (alphabetic variants)

The renaming of bound variables:

$$\lambda x.A \;=_\alpha\; \lambda y.A\{x := y\}, \text{ if } y \text{ is free for } x \text{ in } A$$

The Rules of λ-Conversion (Church, 1985)

Rule 1: β-conversion (function application)

The substitution of all free occurrences of a variable in a formula with a suitable argument:

$$(\lambda x.A)(B) =_\beta A\{x := B\}$$, if x is free for B in A

Rule 2: α-conversion (alphabetic variants)

The renaming of bound variables:

$$\lambda x.A =_\alpha \lambda y.A\{x := y\}$$, if y is free for x in A

Rule 3: η-conversion (identifying co-extensional functions)

The replacement of $\lambda x.A(x)$ by A:

$$\lambda x.A(x) =_\eta A$$, if x is not free in A.

λ-Logic (Beeson, 2005)

A problem:

- The λ-calculus only has 2 operators: application, abstraction
- The translations of NL sentences typically use connectives and quantifiers: $\wedge, \vee, \neg, \rightarrow, \leftrightarrow, =, \forall, \exists, \ldots$

The solution: Extend the λ-calculus to a λ-logic:
(This logic combines application and abstraction with the familiar logical connectives and quantifiers)

Term-forming rules

(i) All non-logical constants and variables are terms, $\bot$ is a term;
$_\lambda$(ii) If A and B are terms, then $A(B)$ is a term;
$_\lambda$(iii) If A is a term and x a variable, then $\lambda x. A$ is a term;
(iv) If B and C are terms, then $(B \rightarrow C)$ is a term.

Notation

(Henkin, 1950)

From $\bot$ and $\to$, the other connectives and quantifiers are easily obtained:

$\top$	stands for	$\bot \to \bot$
$\forall x.\, A$	stands for	$(\lambda x.\top) \to (\lambda x.\, A)$
$B = C$	stands for	$\forall Y.\, Y(B) \to Y(C)$
$\neg B$	stands for	$\lambda x.\, B(x) = \bot$
$B \wedge C$	stands for	$(\lambda x.(\lambda X.\, X(B = C)) = (\lambda X.\, X(\top)))$
	...	

Typed λ-Logic

(Church, 1940), cf. (Curry and Feys, 1958)

Another problem:

- (Untyped) λ-logics allow the self-application of predicates.
- ↩ But this gives rise to the familiar Russell-style paradoxes.
- ↩ This also disables a 'logical' explanation of grammatical well-formedness/distributional phenomena.

The solution: 1. Introduce a type system, cf. (Russell, 1908):

Types (↪ '1Type')

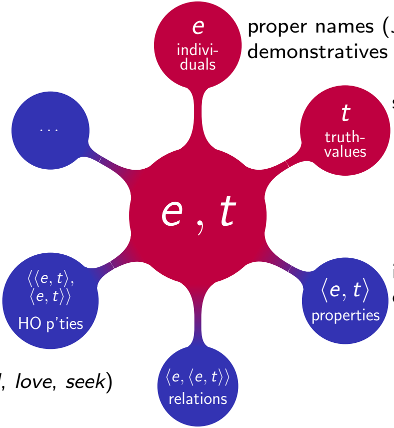
(i) Basic types: e (for individuals/'entities'), t (for truth-values)

(ii) Complex types: $\alpha \to \beta$ (written $\langle \alpha, \beta \rangle$), where α, β are types (for functions from type-α to type-β objects)

1Type $\ni$ $\{ \langle e, t \rangle , \langle e, \langle e, t \rangle \rangle , \langle \langle e, t \rangle, t \rangle , \langle \langle \langle e, t \rangle, t \rangle, t \rangle , \ldots \}$

Merits of Typing 1

Types structure the semantic domains underlying natural language:



- e: proper names (*John*, *Mary*, *Bill*), demonstratives (*he*, *she*, *that*)
- t: sentences (*John is a man*, *John loves Mary*, *Bill walks quickly*)
- $\langle e, t\rangle$: intrans. verbs (*walk*, *talk*), common nouns (*unicorn*, *man*, *woman*)
- $\langle e, \langle e, t\rangle\rangle$: transitive verbs (*find*, *love*, *seek*)
- $\langle\langle e, t\rangle, \langle e, t\rangle\rangle$: verb phrase adverbs (*quickly*, *rapidly*, *allegedly*)
- . . .: sentence adverbs, s-complement verbs, prepositions

Merits of Typing 2

Types provide a formal basis for syntactic categories, and explain the grammatical well-formedness of NL expressions:

Penny talks.✓	$talk : \langle e, t\rangle,\ penny : e$
A philosopher talks.✓	$a : \langle\langle e, t\rangle, \langle\langle e, t\rangle, t\rangle\rangle,\ philosopher, talk : \langle e, t\rangle$
Mary meets Penny.✓	$meet : \langle e, \langle e, t\rangle\rangle,\ mary, penny : e$
Mary talks Penny.*	$talk : \_\_\langle e, t\rangle\_,\ mary : e,\ penny : e$
Mary meets ____.*	$meet : \langle e, \langle e, t\rangle\rangle,\ mary : e$

Typed λ-Logic (Church, 1940), cf. (Curry and Feys, 1958)

1. Introduce a type system for our λ-logic. ✓
2. Type the λ-logical terms. ↩

TY_1 A λ-logic with basic types e and t, cf. (Church, 1940)

To type TY_1 terms, we need the definition of conjoinable types:

Conjoinable types (➥ 'CoType') (Partee and Rooth, 1983)

Types of the form $\langle \alpha_1, \langle \ldots \langle \alpha_n, t \rangle \rangle \rangle$ that 'end in t'.

$$\text{CoType} \ni \{\ t\ ,\ \langle e,t\rangle\ ,\ \langle e,\langle e,t\rangle\rangle\ ,\ \langle\langle e,t\rangle,t\rangle\ ,\ \langle\langle\langle e,t\rangle,t\rangle,t\rangle\ ,\ \ldots\ \}$$

TY_1 Terms

Basic TY_1 terms

- A set, $L := \cup_{\alpha \in \mathbf{1Type}} L_\alpha$, of non-logical constants;
- A set, $\mathcal{V} := \cup_{\alpha \in \mathbf{1Type}} \mathcal{V}_\alpha$, of variables.

TY_1 terms

The set T_α of TY_1 terms of the type α is defined as follows:

(i) $L_\alpha, \mathcal{V}_\alpha \subseteq T_\alpha, \quad \bot \in T_{\mathbf{t}}$;

$_\lambda$(ii) If $A \in T_{\langle\alpha,\beta\rangle}$ and $B \in T_\alpha$, then $A(B) \in T_\beta$;

$_\lambda$(iii) If $A \in T_\beta$ and $x \in \mathcal{V}_\alpha$, then $(\lambda x.\, A) \in T_{\langle\alpha,\beta\rangle}$;

(iv) If $B, C \in T_{\epsilon \in \mathbf{CoType}}$, then $(B \to C) \in T_{\mathbf{t}}$.

We require that terms involving $\top, \forall, =, \neg, \wedge$ are suitably typed.

TY_1 Frames

$D_\alpha :=$ the set of objects (the domain) of the type α:

- $\mathcal{A} :=$ the set of individuals ('entities'): the domain of type e.
- **2** $:=$ the set $\{\mathbf{T}, \mathbf{F}\}$ of truth-values: the domain of type t.
- $D_{\langle \alpha, \beta \rangle} \subseteq \{f \mid f : D_\alpha \to D_\beta\} :=$ a subset of the set of fcts from objects of type α to objects of type β: the domain of type $\langle \alpha, \beta \rangle$.

General TY_1 frame

A set $F = \{D_\alpha \mid \alpha \in 1\text{Type}\}$ of non-empty TY_1 domains.

The generality of frames ensures the recursive axiomatizability of the entailment relation, and the completeness of TY_1.

General TY_1 Models

- TY_1 terms in L are related to TY_1 objects in F via
  interpretation functions $I_F : L \to F$, s.t. $I_F(c_\alpha) \in D_\alpha$;
  variable assignments $g_F : \mathcal{V} \to F$, s.t. $g_F(x_\alpha) \in D_\alpha$.
- We write $g_F[d_\alpha/\mathbf{x}_\alpha]$ for the assignment g'_F s.t. $g'_F(\mathbf{x}) = d$ and $g'_F(\mathbf{y}_\alpha) = g_F(\mathbf{y})$ if $\mathbf{x} \neq \mathbf{y}$.
- We denote the set of all assignments g_F w.r.t. F by $\mathcal{G}_F$.

Definition (General TY_1 model)

A triple $M_F = \langle F, I_F, V_F \rangle$, where $V_F : (\mathcal{G}_F \times \cup_\alpha T_\alpha) \to F$ is s.t.

(i) $V_F(g_F, c) := I_F(c)$ if $c \in L$,
$V_F(g_F, x) := g_F(x)$ if $x \in \mathcal{V}$;

$_\lambda$(ii) $V_F(g_F, A(B)) := V_F(g_F, A)\big(V_F(g_F, B)\big)$;

$_\lambda$(ii) $V_F(g_F, \lambda x_\alpha.A) := \big\{\big\langle d, V_F(g_F[d/x], A)\big\rangle \mid d \in D_\beta\big\}$.

Truth, Entailment, and Equivalence

TY_1 truth

ϕ_t is true in M_F, g_F (i.e. $\models_{M_F, g_F} \phi$) iff $V_F(g_F, \phi) = \mathbf{T}$

ϕ_t is false in M_F, g_F (i.e. $=\!|_{M_F, g_F} \phi$) iff $V_F(g_F, \phi) = \mathbf{F}$
iff $\not\models_{M_F, g_F} \phi$ iff $V_F(g_F, \neg\phi) = \mathbf{T}$

TY_1 entailment

$\Gamma = \{\gamma \,|\, \gamma \in T_{\epsilon \in \mathrm{CoType}}\}$ entails $\Delta = \{\delta \,|\, \delta \in T_\epsilon\}$ (i.e. $\Gamma \models_g \Delta$) if

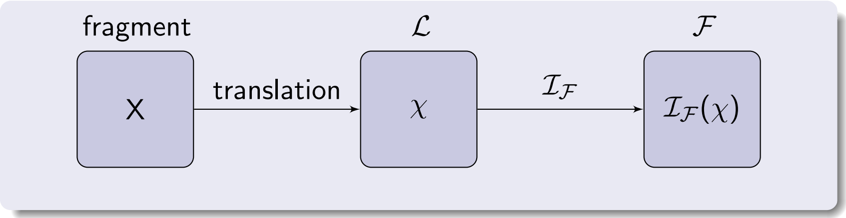
$$\bigcap_{\gamma \in \Gamma} V_F(g_F, \gamma) \subseteq \bigcup_{\delta \in \Delta} V_F(g_F, \delta) \quad \text{for all } M_F, g_F.$$

- We define TY_1 equivalence as mutual TY_1 entailment.
- The behavior of TY_1 entailment is characterized by classical sequent rules.

From Natural Language to TY_1 Semantics

Indirect interpretation We interpret (a fragment of) natural language via its translation into the language of TY_1:

1. Formalize a fragment of natural language.
2. Develop the lang. $\mathcal{L}$ and models $\langle \mathcal{F}, \mathcal{I}_{\mathcal{F}} \rangle$ of the interpr. logic.
3. Provide a set of translation rules from expressions X of the fragment to terms χ of the logic.



The Fragment

1. Lexical insertion rules:

Label	Rule			Traditional name
(LI 1)	DET	$\longrightarrow$	every, the, a	Determiner
(LI 2)	NP	$\longrightarrow$	Mary, Penny, she$_n$	Noun Phrase
(LI 3)	NP	$\longrightarrow$	who(m)	Noun Phrase
(LI 4)	N	$\longrightarrow$	philosopher	Common Noun
(LI 5)	IV	$\longrightarrow$	talk, exist	Intransitive Verb
(LI 6)	TV	$\longrightarrow$	meet, be	Transitive Verb
(LI 7)	SCV	$\longrightarrow$	know	Sentence-Comp. Verb
(LI 8)	C	$\longrightarrow$	that	Complementizer
(LI 9)	ADJ	$\longrightarrow$	mathematical	Adjective

The Fragment

2. Phrase structure rules:

Label	Rule			Traditional name
(PS 1)	S	$\longrightarrow$	NP VP	Sentence
(PS 2)	NP	$\longrightarrow$	DET N	Noun Phrase
(PS 3)	CP	$\longrightarrow$	C S	Complement Phrase
(PS 4)	VP	$\longrightarrow$	IV	Verb Phrase
(PS 5)	VP	$\longrightarrow$	TV NP	Verb Phrase
(PS 6)	VP	$\longrightarrow$	SCV CP	Verb Phrase
(PS 7)	N	$\longrightarrow$	ADJ N	Common Noun
...		...		...

The Language $\mathcal{L}$

Constant	TY_1 type	Variable	TY_1 type
mary, *penny*	e	$x, x_1, \ldots, x_n, y, z$	e
talk, *philosopher*	$\langle e, t\rangle$	p, q, r	t
meet	$\langle e, \langle e, t\rangle\rangle$	$P, P_1, \ldots, P_n$	$\langle e, t\rangle$
know	$\langle t, \langle e, t\rangle\rangle$	$Q, Q_1, \ldots, Q_n$	$\langle\langle e, t\rangle, t\rangle$
mathematical	$\langle\langle e, t\rangle, \langle e, t\rangle\rangle$	$R, R_1, \ldots, R_n$	$\langle\alpha_1, \langle\ldots\langle\alpha_n, t\rangle\rangle\rangle$
	$\vec{X}$	a sequence of variables of the types	$\alpha_1, \ldots, \alpha_n$

- The frame $\mathcal{F}$ is very large.
- $\mathcal{I}_{\mathcal{F}} : \mathcal{L} \to \mathcal{F}$ respects the conventional rel's bw. content words:

$$\mathcal{I}_{\mathcal{F}}(talk) \subseteq \mathcal{I}_{\mathcal{F}}(\lambda x \exists y.\, y = x)$$

NL-to-TY_1 Translation

NL expressions (LFs) are translated into TY_1 terms via type-driven translation (Klein and Sag, 1985):

Definition (Type-driven translation)

The smallest relation, $\rightsquigarrow$, between LFs and TY_1 terms such that

- (T0) $X \rightsquigarrow A$ if X is a word and A its translation. (Base Rule)
- (T1) If $X \rightsquigarrow A$, then $[X] \rightsquigarrow A$. (Copying)
- (T2) If $X \rightsquigarrow A$ and $Y \rightsquigarrow B$, then $[XY] \rightsquigarrow A(B)$ if $A(B)$ is well-formed, $[YX] \rightsquigarrow B(A)$ ow; (Application)
- (T3) If $X \rightsquigarrow A$, $Y \rightsquigarrow B$, then, if $A(\lambda v_n. B)$ is well-formed, $[X^n Y] \rightsquigarrow A(\lambda v_n. B)$. (Quantifying In)
- (T4) If $X \rightsquigarrow A$, A reduces to B, then $X \rightsquigarrow B$. (Reduction)

Basic Translations

(T0) governs the translation of lexical elements/words:

Basic TY_1 translations

Mary	$\rightsquigarrow$	mary;	Penny	$\rightsquigarrow$	penny;
t_n	$\rightsquigarrow$	x_n f. each n;	who(m)	$\rightsquigarrow$	$\lambda P.\, P$;
t_n/she$_n$	$\rightsquigarrow$	x_n f. each n;	that	$\rightsquigarrow$	$\lambda p.\, p$;
philosopher	$\rightsquigarrow$	$\mathit{philosopher}$;	talk	$\rightsquigarrow$	talk;
exist	$\rightsquigarrow$	$\lambda x \exists y.\, y = x$;	meet	$\rightsquigarrow$	$\lambda y \lambda x.\, \mathit{meet}(y)(x)$;
be	$\rightsquigarrow$	$\lambda y \lambda x.\, x = y$;	know	$\rightsquigarrow$	$\lambda p s \lambda x.\, \mathit{know}(p)(x) \wedge p$;
mathemat'l	$\rightsquigarrow$	$\lambda P \lambda x.\, (\mathit{mathematical}(P))(x) \wedge P(x)$;			
a	$\rightsquigarrow$	$\lambda P_1 \lambda P \exists x.\, P_1(x) \wedge P(x)$;			
every	$\rightsquigarrow$	$\lambda P_1 \lambda P \forall x.\, P_1(x) \rightarrow P(x)$;			
the	$\rightsquigarrow$	$\lambda P_1 \lambda P \exists x \forall y.(P_1(y) \leftrightarrow x = y) \wedge P(x)$			
and	$\rightsquigarrow$	$\lambda R_1 \lambda R \lambda \vec{X}.\, R(\vec{X}) \wedge R_1(\vec{X})$;			
not	$\rightsquigarrow$	$\lambda R \lambda \vec{X}.\, \neg R(\vec{X})$;			

Derived Translations: 'Penny talks'

1. $[_{\mathrm{NP}}\text{Penny}] \rightsquigarrow \mathit{penny}$ (by (T0))
2. $[_{\mathrm{IV}}\text{talks}] \rightsquigarrow \mathit{talk} =_{\eta} \lambda x.\,\mathit{talk}(x)$ (by (T0), (T4))
3. $[_{\mathrm{VP}}[_{\mathrm{IV}}\text{talks}]] \rightsquigarrow \lambda x.\,\mathit{talk}(x)$ (by (T1))
4. $[_{\mathrm{S}}[_{\mathrm{NP}}\text{Penny}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{talks}]]] \rightsquigarrow \lambda x.\,\mathit{talk}(x)\ \mathit{penny}$ (by (T2))
   $=_{\beta}\ \mathit{talk}(\mathit{penny})$

Analogous: Mary meets Penny

Derived Translations: 'Mary meets Penny'

1. $[_{\mathrm{NP}}\text{Penny}] \rightsquigarrow \mathit{penny}$ (by (T0))
2. $[_{\mathrm{VP}}[_{\mathrm{IV}}\text{talks}]] \rightsquigarrow \lambda x.\, \mathit{talk}(x)$ ((T0), (T1), (T4))
4. $[_{\mathrm{S}}[_{\mathrm{NP}}\text{Penny}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{talks}]]] \rightsquigarrow \lambda x.\, \mathit{talk}(x)\ \mathit{penny}$ (by (T2))
   $=_\beta\ \mathit{talk}(\mathit{penny})$

Analogous:

1. $[_{\mathrm{NP}}\text{Mary}] \rightsquigarrow \mathit{mary}$ 2. $[_{\mathrm{NP}}\text{Penny}] \rightsquigarrow \mathit{penny}$ (by (T0))
3. $[_{\mathrm{IV}}\text{meets}] \rightsquigarrow \mathit{meet} =_\eta \lambda y \lambda x.\, \mathit{meet}(y)(x)$ ((T0), (T4))
4. $[_{\mathrm{VP}}[_{\mathrm{IV}}\text{meets}][_{\mathrm{NP}}\text{Penny}]] \rightsquigarrow \lambda y \lambda x.\, \mathit{meet}(y)(x)\ \mathit{penny}$ (by (T2))
   $=_\beta\ \lambda x.\, \mathit{meet}(\mathit{penny})(x)$
5. $[_{\mathrm{S}}[_{\mathrm{NP}}\text{Mary}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{meets}][_{\mathrm{NP}}\text{Penny}]]]$
   $\rightsquigarrow\ \lambda x.\, \mathit{meet}(\mathit{penny})(x)\ \mathit{mary}$ (by (T2))
   $=_\beta\ \lambda x.\, \mathit{meet}(\mathit{penny})(\mathit{mary})$

Other Derived Translations

$[_S[_{NP}\text{Penny}][_{VP}[_{TV}\text{is}][_{NP}[_{DET}\text{a}][_N\text{philosopher}]]]]$

$\rightsquigarrow\quad \exists x.\, philosopher(x) \wedge penny = x$

$=\quad philosopher(penny)$

$[_S[_{NP}\text{Penny}][_{VP}[_{TV}\text{is}][_{NP}[_{DET}\text{a}][_N[_{ADJ}\text{mathematical}][_N\text{philosopher}]]]]]$

$\rightsquigarrow\quad \exists x.((math'l(philosopher))(x) \wedge philosopher(x)) \wedge penny(x)$

$[_S[_{NP}\text{mary}][_{VP}[_{SCV}\text{knows}][_{CP}[_C\text{that}]$
$[_S[_{NP}\text{Penny}][_{VP}[_{TV}\text{is}][_{NP}[_{DET}\text{a}][_N[_{ADJ}\text{mathematical}][_N\text{philosopher}]]]]]]]]]$

$\rightsquigarrow\quad K(\exists x.((math'l(philosopher))(x) \wedge philos.(x)) \wedge penny(x), mary) \wedge$
$(\exists x.((math'l(philosopher))(x) \wedge philos.(x)) \wedge penny(x))$

$[_S[_{NP}[_{DET}\text{a}][_N\text{philosopher}]][_{VP}[_{IV}\text{exists}]]] \quad\rightsquigarrow\quad \exists x.\, philosopher(x)$

$[_S[_{NP}[_{DET}\text{a}][_N[_{ADJ}\text{mathematical}][_N\text{philosopher}]]][_{VP}[_{IV}\text{exists}]]]$

$\rightsquigarrow\quad \exists x.\,(mathematical(philosopher))(x)$

TY_1-Based NL Entailment

Assume $\Xi = \{X \mid X \rightsquigarrow \gamma\}$ and $\Upsilon = \{Y \mid Y \rightsquigarrow \delta\}$ are sets of NL sentences which are translated into the sets of TY_1 terms $\Gamma = \{\gamma \mid \gamma \in T_t\}$ and $\Delta = \{\delta \mid \delta \in T_t\}$.

NL entailment

Ξ entails Υ w.r.t. $M_{\mathcal{F}}, g_{\mathcal{F}}$ if $\models_{M_{\mathcal{F}}, g_{\mathcal{F}}} \Gamma \Rightarrow \Delta$.

Mary knows that Penny is a mathematical philosopher.
entails Mary knows that Penny is a philosopher.
entails Mary knows that philosophers exist.

Mary knows that Penny is a mathematical philosopher.
entails Penny is a mathematical philosopher.
entails Penny is a philosopher.
entails Philosophers exist.

TY_1-Based NL Equivalence and Consistency

NL entailment

Ξ entails Υ w.r.t. $M_{\mathcal{F}}, g_{\mathcal{F}}$ if $\models_{M_{\mathcal{F}}, g_{\mathcal{F}}} \Gamma \Rightarrow \Delta$.

NL equivalence

Ξ is equivalent to Υ w.r.t. $M_{\mathcal{F}}, g_{\mathcal{F}}$ if $\models_{M_{\mathcal{F}}, g_{\mathcal{F}}} \Gamma \Leftrightarrow \Delta$.

Mary knows that Penny is a mathematical philosopher.

is equiv. to Mary knows that Penny is a mathematical philosopher, and Penny is a mathematical philosopher.

is equiv. to Mary knows that Penny is a mathematical philosopher and that Penny is a philosopher, and Penny is a mathematical philosopher and is a philosopher.

Explain why the last 2 sentences appear redundant.

Another Merit: Disambiguation

1. Quantifier interaction — **Ex.:** Every man loves a woman.

∃-narrow scope: $[_S[_{NP}[_D \text{every}][_N \text{man}]][_{VP}[_{TV}\text{loves}][_{NP}[_D \text{a}][_N \text{woman}]]]]$

$\rightsquigarrow \forall x.\, man(x) \to (\exists y.\, woman(y) \wedge love(y,x))$

∃-wide scope: $[[_{NP}[_D \text{a}][_N \text{woman}]]^1[_S[_{NP}[_D \text{every}][_N \text{man}]][_{VP}[_{TV}\text{loves}]\ t_1]]]$

$\rightsquigarrow \exists y.\, woman(y) \wedge (\forall x.\, man(x) \to love(y,x))$

2. Quantifier/intensional context-interaction — **Ex.:** John seeks a unicorn.

∃-narrow scope: $[_S[_{NP}\text{John}][_{VP}[_{TV}\text{seeks}][_{NP}[_{DET}\text{a}][_N \text{unicorn}]]]]$

$\rightsquigarrow seek([\lambda P \exists x.\, unicorn(x) \wedge P(x)], john)$

∃-wide scope: $[_S[_{NP}[_{DET}\text{a}][_N \text{unicorn}]]^0\ [_S[_{NP}\text{John}][_{VP}[_{TV}\text{seeks}]\ t_0]]]$

$\rightsquigarrow \exists x.\, unicorn(x) \wedge seek([\lambda P.\, P(x)], john)$

↩ The unicorn has become the 'mascot' of Montague semantics.

Problem 1: Coarse-grained sentence-interpretations

Fact: TY_1 interprets sentences and CPs as truth-values.

- All true sentences are logically equivalent.
- We may substitute true sentences in all contexts, including propositional attitude contexts.
- But this warrants counterintuitive inferences:

Mary knows that Penny is a philosopher.	**T**
Penny is a philosopher $\Leftrightarrow 1^3 + 12^3 = 9^3 + 10^3$.	**T**
Mary knows that $1^3 + 12^3 = 9^3 + 10^3$.	**F**

Problem 2: Partee's 'temperature puzzle' (Montague, 1973)

Fact: TY_1 interprets all common nouns and intransitive verbs as sets of individuals (type $\langle e, t \rangle$).

- Intensional Ns (*temperature*) and IVs (*rise*) are interpreted in the type $\langle e, t \rangle$.
- But this warrants counterintuitive inferences:

The temperature is ninety.	**T**
The temperature rises.	**T**
Ninety rises.	**?**

Montague's Solution

(Montague, 1973), cf. (Gallin, 1975)

1. Extend the TY_1 type system via a type s for **indices** (possible worlds/world-time pairs): ➥ TY_2

TY_2 Types (➥ '2Type')

(i) Basic types: e (for individuals), s (for indices), t (for truth-values)

(ii) Complex types: $\alpha \rightarrow \beta$ (written $\langle \alpha, \beta \rangle$), where α, β are types

2Type $\ni \{\langle s, e\rangle$, $\langle s, t\rangle$, $\langle e, \langle s, t\rangle\rangle$, $\langle\langle s, e\rangle, \langle s, t\rangle\rangle$, $\langle e, \langle e, \langle s, t\rangle\rangle\rangle\}$

2. Interpret sentences and CPs as propositions, $\langle s, t\rangle$, cf. (Kripke, 1963; Stalnaker, 1976);

Montague's Solution

(Montague, 1973), cf. (Gallin, 1975)

2. Interpret sentences and CPs as propositions, $\langle s, t \rangle$, cf. (Kripke, 1963; Stalnaker, 1976);

   Interpret intensional Ns and IVs as p'ties of individual concepts, $\langle\langle s, e\rangle, \langle s, t\rangle\rangle$ (in (Montague, 1973), type $\langle\langle s, e\rangle, t\rangle$):

Mary knows that Penny is a philosopher.	**T**
Penny is a philosopher $\not\Leftrightarrow$ $1^3 + 12^3 = 9^3 + 10^3$.	**T**
~~Mary knows that $1^3 + 12^3 = 9^3 + 10^3$.~~	**F**

The temperature **at @** is ninety.	**T**
The temperature rises.	**T**
~~Ninety rises.~~	**?**

A Remaining Problem: Logical omniscience (Hintikka, 1975)

Fact: TY_2 interprets sentences and CPs as sets of indices.

- All true sentences across all worlds are logically equivalent.
- We may substitute such sentences in all contexts, including propositional attitude contexts.
- But this again warrants counterintuitive inferences:

Mary knows that everything is self-identical.	**T**
Everything is self-identical $\Leftrightarrow 1^3 + 12^3 = 9^3 + 10^3$.	**T**
Mary knows that $1^3 + 12^3 = 9^3 + 10^3$.	**F**

- The problem of logical omniscience is observed in (Carnap, 1988; Lewis, 1970; Cresswell, 1973; Barwise and Perry, 1983).

Solution Strategies

overview in (Fox and Lappin, 2005)

Master strategy: Introduce 'more' semantic values for sentences:
Define a more fine-grained notion of proposition, which does not identify equivalent expressions.

1. 'Structured Meanings' (Carnap, 1988; Lewis, 1970; Cresswell, 1985) also consider the compositional structure of Montagovian propositions;
2. Partialization (Hintikka, 1975; Rantala, 1982; Muskens, 1995) extend the s, t-domains to situations (or impossible worlds) and truth-combinations;
3. Property Theory (Thomason, 1980; Chierchia and Turner, 1988; Pollard, 2008) add a new domain of intensional objects (e.g. primitive propositions).

Wrap-up

Formal semantics . . .

. . . is an active research area in the intersection of linguistics, logic (or comp. sci.), and philosophy.

. . . explains and predicts many properties of NL semantics.

. . . obtains the domain of application for formal techniques.

➥ enables the formal treatment of philosophical problems.

Pointers to the Literature

($\star$:= my favorite)

1. Formal methods in NL semantics:
   - (Partee, ter Meulen, and Wall, 1990): overlaps with today's introductions by Gil & Florian, and Catrin & Sebastian
   - (Gamut, 1991, vol. 1): overlaps with Gil & Florian's lecture
   - ⋆ (Landman, 1991): more algebraic/lattice-theoretic, with detailed linguistic applications

2. Montague semantics (general introductions):
   - (Dowty, Wall, and Peters, 1981): the classic textbook
   - ⋆ (Gamut, 1991, vol. 2): the Amsterdam textbook by van Benthem, Groenendijk, Stokhof, de Jongh, Verkuyl
   - (Heim and Kratzer, 1998): probably the most popular introduction to (general) formal semantics

Pointers to the Literature

($\star$:= my favorite)

3. Montague semantics (historical/overview):
   - ⋆ (Partee and Hendriks, 1997): the authoritative introduction (linguistics-oriented)
   - (Janssen, 2012): a more philosophically-oriented introduction

4. Alternative frameworks
   - Situation semantics (Barwise and Perry, 1983; Kratzer, 1989; Muskens, 1995): replaces possible worlds by partial situations
   - Dynamic semantics (Groenendijk and Stokhof, 1991; Kamp, 1981; Heim, 1982): models intersentential anaphoric relations
   - ⋆ Data semantics (Veltman, 1981; Landman, 1985; 1984; Muskens, 2013): models information growth in discourse
   - Proof-theoretic semantics (Schröder-Heister, 1991; Francez and Dyckhoff, 2011): focuses on proofs, not (model-th.) truth

Pointers to the Literature

($\star$:= my favorite)

5. Up and coming: Montague-style semantics in modern type-theories (MTT semantics)
   - (Ranta, 1994): the earliest monograph on MTT semantics
   - $\star$ (Luo, 2014): accessible lecture notes
6. The lambda calculus
   - (Barendregt and Barendsen, 2000): a classic
   - (Hindley and Seldin, 2008): the standard textbook
   - $\star$ (Barker, 2014): an interactive tutorial
7. The typed lambda calculus
   - $\star$ (Muskens, 2011): applied to Montague-style NL semantics

Thank you!

Bach, Emmon. 1986. *Natural language metaphysics*, Logic, Methodology and Philosophy of Science (Ruth Barcan Marcus, George J.W. Dorn, and Paul Weingartner, eds.), Vol. VII, Elsevier Science Publishers, Amsterdam, 1986.

Barendregt, Henk and Erik Barendsen. 2000. *Introduction to the lambda calculus*: *Revised edition*, Technical report, University of Nijmegen.

Barker, Chris. 2014. *Lambda Tutorial*, `https://files.nyu.edu/cb125/public/Lambda/`.

Barwise, Jon and John Perry. 1983. *Situations and Attitudes*, MIT Press, Cambridge, Mass.

Beeson, Michael. 2005. *Lambda logic*, LPAR-05 Workshop: Empirically Successful Automated Reasoning in Higher-Order Logic (ESHOL) (Christoph Benzmüller, John Harrison, and Carsten Schürmann, eds.), 2005.

Carlson, Gregory N. 1977. *Reference to Kinds in English*, Outstanding Dissertations in Linguistics, Garland Publishing, Inc., New York and London, 1980.

Carnap, Rudolf. 1988. *Meaning and Necessity*: *A study in semantics and modal logic*, University of Chicago Press, Chicago and London.

Chierchia, Gennaro and Raymond Turner. 1988. *Semantics and property theory*, Linguistics and Philosophy **11**, 261–302.

Church, Alonzo. 1940. *A formulation of the Simple Theory of Types*, Journal of Symbolic Logic **5/2**, 56–68.

______. 1985. *The Calculi of Lambda Conversion*, Princeton University Press, Princeton, NJ.

Cresswell, Maxwell J. 1973. *Logics and Languages*, Methuen Young Books, London.

______. 1985. *Structured Meanings*: *The semantics of propositional attitudes*, MIT Press, Cambridge, Mass.

Crouch, Dick and Tracy Holloway King. 2008. *Type-checking in formally non-typed systems*, Software Engineering, Testing, and Quality Assurance for Natural Language Processing, Association for Computational Linguistics, Columbus, Ohio, 2008.

Curry, H.B. and R. Feys. 1958. *Combinatory logic*: *Vol. 1*, Cambridge Textbooks in Linguistics, vol. 1, North-Holland, Cambridge and New York.

Dowty, David R., Robert E. Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*, Studies in Linguistics and Philosophy, vol. 11, Kluwer Academic Publishers, Dordrecht.

Fox, Chris and Shalom Lappin. 2005. *Foundations of Intensional Semantics*, Blackwell, Malden, Mass.

Francez, Nissim and R. Dyckhoff. 2011. *Proof-theoretic semantics for a natural language fragment*, Linguistics and Philosophy **33/6**, 447–477.

Gallin, Daniel. 1975. *Intensional and Higher-Order Modal Logic with Applications to Montague Semantics*, North Holland, Amsterdam.

Gamut, L.T.F. 1991. *Logic, Language, and Meaning*, University of Chicago Press, Chicago and London.

Groenendijk, Jeroen and Martin Stokhof. 1991. *Dynamic predicate logic*, Linguistics and Philosophy **14**, 39–100.

Heim, Irene. 1982. *The Semantics of Definite and Indefinite Noun Phrases*, Dissertation, University Microfilms International, Ann Arbor, MI.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*, Blackwell Textbooks in Linguistics, vol. 13, Blackwell, Malden, Mass. and Oxford.

Henkin, Leon. 1950. *Completeness in the theory of types*, Journal of Symbolic Logic **15**, 81–91.

Hindley, J. Roger and Jonathan P. Seldin. 2008. *Lambda-Calculus and Combinators*: *An introduction*, Cambridge University Press, Cambridge.

Hintikka, Jaakko. 1975. *Impossible possible worlds vindicated*, Journal of Philosophical Logic **4**, 475–484.

Janssen, Theo M.V. 2012. *Montague semantics*, The Stanford Encyclopedia of Philosophy, Winter 2012 edition (Edward N. Zalta, ed.), 2012.

Kamp, Hans. 1981. *A theory of truth and semantic representation*, Formal Methods in the Study of Language (Jeroen Groenendijk, Theo Janssen, and Martin Stokhof, eds.), Mathematical Centre, Amsterdam, 1981.

Klein, Ewan and Ivan Sag. 1985. *Type-driven translation*, Linguistics and Philosophy **8**, 163–201.

Kratzer, Angelika. 1989. *An investigation into the lumps of thought*, Linguistics and Philosophy **12**, 607–653.

Kripke, Saul A. 1963. *Semantical considerations on modal logic*, Acta Philosophica Fennica **16**, 83–94.

Landman, Fred. 1985. *Data semantics*: *An epistemic theory of partial objects*, Towards a Theory of Information: The status of partial objects in semantics (Jeroen Groenendijk, Dick de Jongh, and Martin Stokhof, eds.), Groningen-Amsterdam Studies in Semantics, vol. 8, Foris Publications, Dordrecht, 1986.

______. 1984. *Pegs and alecs*, Towards a Theory of Information, Foris Publications, Dordrecht, 1986.

______. 1991. *Structures for Semantics* (Gennaro Chierchia, Pauline Jacobson, and Francis J. Pelletier, eds.), Studies in Linguistics and Philosophy, vol. 45, Kluwer Academic Publishers, Dordrecht, Boston, and London.

Lewis, David. 1970. *General semantics*, Synthese **22/1–2**, 18–67.

Luo, Zhaohui. 2014. *Formal Semantics in Modern Type Theories*: *Is It model-theoretic, proof-theoretic, or both?*, Proceedings of LACL'14.

Montague, Richard. 1970a. *English as a formal language*, Formal Philosophy: Selected papers of Richard Montague (Richmond H. Thomason, ed.), Yale University Press, New Haven and London, 1976.

______. 1970b. *Universal grammar*, Formal Philosophy, 1976.

______. 1973. *The proper treatment of quantification in ordinary English*, Formal Philosophy, 1976.

Muskens, Reinhard. 1995. *Meaning and Partiality*, CSLI Lecture Notes, FoLLI, Stanford.

______. 2011. *Type-logical semantics*, Routledge Encyclopedia of Philosophy Online (Edward Craig, ed.), Routledge, New York, 2011.

______. 2013. *Data Semantics and Linguistic Semantics*, The dynamic, inquisitive, and visionary life of ϕ, $?\phi$, and $\diamond\phi$: A festschrift for Jeroen Groenendijk, Martin Stokhof, and Frank Veltman (Maria Aloni, Michael Franke, and Floris Roelofsen, eds.), http://www.illc.uva.nl/Festschrift-JMF/, Amsterdam, 2013.

Partee, Barbara and Mats Rooth. 1983. *Generalized conjunction and type ambiguity*, Meaning, Use and Interpretation of Language (Rainer Bauerle, Christoph Schwarze, and Arnim von Stechow, eds.), Foundations of Communication, Walter De Gruyter, Berlin, 1983.

Partee, Barbara. 1984. *Compositionality*, Varieties of Formal Semantics: Proceedings of the 4th Amsterdam Colloquium (Fred Landman and Frank Veltman, eds.), Groningen-Amsterdam Studies in Semantics, vol. 3, 1984.

Partee, Barbara, Alice ter Meulen, and Robert Wall. 1990. *Mathematical Methods in Linguistics*, Studies in Linguistics and Philosophy, Kluwer Academic Publishers, Dordrecht.

Partee, Barbara and H.L.W. Hendriks. 1997. *Montague grammar*, Handbook of Logic and Language (Johan van Benthem and Alice ter Meulen, eds.), Elsevier Science Publishers, Amsterdam, 1997.

Pollard, Carl. 2008. *Hyperintensions*, Journal of Logic and Computation **18/2**, 257–282.

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

Rantala, Veikko. 1982. *Quantified modal logic*: *Non-normal worlds and propositional attitudes*, Studia Logica **41/1**, 41–65.

Russell, Bertrand. 1905. *On denoting*, Mind **14/56**, 479–493.

———. 1908. *Mathematical logic as based on the theory of types*, American Journal of Mathematics **30/3**, 222–262.

Schröder-Heister, Peter. 1991. *Uniform proof-theoretic semantics for logical csonstants*, Journal of Symbolic Logic **56**, 1142.

Stalnaker, Robert. 1976. *Propositions*, Issues in the Philosophy of Language (A.F. Mackay and D.D. Merrill, eds.), Yale University Press, New Haven, 1976.

Thomason, Richmond H. 1980. *A model theory for the propositional attitudes*, Linguistics and Philosophy **4**, 47–70.

Veltman, Frank. 1981. *Data semantics*, Formal Methods in the Study of Language: Part 2 (Jeroen Groenendijk, Theo Janssen, and Martin Stokhof, eds.), Mathematical Centre Tracts, vol. 136, Mathematisch Centrum, Amsterdam, 1981.