Lecture 1: Introduction to Formal Semantics and Compositionality

Read for next time: (1) R. Larson (1995) Semantics. Chapter 12 in L. Gleitman and M.Liberman, eds. *An Invitation to Cognitive Science, Vol I: Language*. (D. Osherson, general editor), pp 361-380. https://udrive.oit.umass.edu/partee/Semantics_Readings/Larson_chpt.12.pdf

(2) Partee, Barbara H. 1999. "Semantics" in R.A. Wilson and F.C. Keil, eds., *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge, MA: The MIT Press. 739-742.
https://udrive.oit.umass.edu/partee/Semantics_Readings/Partee%20MITECS%20Semantics.pdf

(3) Partee, Barbara H. In press. Formal semantics. In *The Cambridge Encyclopedia of the Language Sciences*, ed. Patrick Colm Hogan. Cambridge: Cambridge University Press. https://udrive.oit.umass.edu/partee/Partee_Formal_Semantics_Camb_Encyc.pdf

LOOK AT THE WEBSITE FOR THE COURSE:
http://people.umass.edu/partee/RGGU_2008/RGGU08_formal_semantics.htm

– become familiar with it, and with the "Links to Readings 2008" document linked there:
https://udrive.oit.umass.edu/partee/Semantics_Readings/Links%20to%20Readings%202008.doc

-- the "Links to Readings 2008" contains clickable links to all downloadable readings, and more supplementary semantic resources as well. Both the website and the Links to Readings will be updated during the semester.

Bring next time or send by e-mail: "Anketa": see page 10. (I'll send an electronic version to everyone whose e-mail address I have. If you don't get one, please send me an e-mail.)

1. Compositional Semantics

1.1. The Principle of Compositionality.

A basic starting point of generative grammar: there are infinitely many sentences in any natural language, and the brain is finite, so linguistic competence must involve some finitely describable means for specifying an infinite class of sentences. That is a central task of **syntax**.

Semantics: A speaker of a language knows the meanings of those infinitely many sentences, is able to understand a sentence he/she has never heard before or to express a meaning he/she has never expressed before. So for semantics also there must be a finite way to specify the meanings of the infinite set of sentences of any natural language.

A central principle of formal semantics is that the relation between syntax and semantics is compositional.

The Principle of Compositionality: The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.

Each of the key terms in the principle of compositionality is a "theory-dependent" term, and there are as many different versions of the principle as there are ways of specifying those terms. (*meaning, function, parts (syntax)*)

Some of the different kinds of things meanings could be in a compositional framework:

(a) (early Katz and Fodor) Representations in terms of semantic features. bachelor: [+HUMAN, +MALE, +ADULT, +NEVER-MARRIED (?!)]. Semantic composition: adding feature sets together. Problems: insufficient structure for the representations of transitive verbs, quantifiers, and many other expressions; unclear status of uninterpreted features.

(b) Representations in a "language of thought" or "conceptual representation" (Jackendoff, Jerry Fodor); if semantics is treated in terms of representations, then semantic composition becomes a matter of compositional translation from a syntactic representation to a semantic representation.

(c) The logic tradition: Frege, Tarski, Carnap, Montague. The basic meaning of a sentence is its truth-conditions: to know the meaning of a sentence is to know what the world must be like if the sentence is true. Knowing the meaning of a sentence does not require knowing whether the sentence is *in fact* true; it only requires being able to discriminate between situations in which the sentence is true and situations in which the sentence is false.

Starting from the idea that the meaning of a sentence consists of its truth-conditions, meanings of other kinds of expressions are analyzed in terms of their contribution to the truth-conditions of the sentences in which they occur.

1.2. Model-theoretic Semantics.

In formal semantics, truth-conditions are expressed in terms of truth relative to various parameters — a formula may be true at a given time, in a given possible world, relative to a certain context that fixes speaker, addressee, etc., and relative to a certain assignment of meanings to its atomic "lexical" expressions and of particular values to its variables. For simple formal languages, all of the relevant variation except for assignment of values to variables is incorporated in the notion of truth relative to a *model*. Semantics which is based on truth-conditions is called *model-theoretic.*

Compositionality in the Montague Grammar tradition:
The task of a **semantics** for language L is to provide truth conditions for every well-formed sentence of L, and to do so in a compositional way. This task requires providing appropriate model-theoretic interpretations for the *parts* of the sentence, including the lexical items.

The task of a **syntax** for language L is (a) to specify the set of well-formed expressions of L (of every category, not only sentences), and (b) to do so in a way which supports a compositional semantics. The syntactic part-whole structure must provide a basis for semantic rules that specify the meaning of a whole as a function of the meanings of its parts.

Basic structure in classic Montague grammar:

(1) **Syntactic categories and semantic "types":** For each syntactic category there must be a uniform semantic type. For example, one could hypothesize that sentences express propositions, nouns and adjectives express properties of entities, verbs express properties of events.

(2) **Basic (lexical) expressions and their interpretation.** Some syntactic categories include basic expressions; for each such expression, the semantics must assign an

interpretation of the appropriate type. Within the tradition of formal semantics, most lexical meanings are left unanalyzed and treated as if primitive; Montague regarded most aspects of the analysis of lexical meaning as an empirical rather than formal matter; formal semantics is concerned with the *types* of lexical meanings and with certain aspects of lexical meaning that interact directly with compositional semantics, such as verbal aspect.

(3) **Syntactic and semantic rules.** Syntactic and semantic rules come in pairs: **<Syntactic Rule n, Semantic Rule n>:** in this sense compositional semantics concerns "the semantics of syntax".

Syntactic Rule n: If α is an expression of category A and β is an expression of category B, then $F_i(\alpha,\beta)$ is an expression of category C. [where F_i is some syntactic operation on expressions]

Semantic Rule n: If α is interpreted as α' and β is interpreted as β', then $F_i(\alpha,\beta)$ is interpreted as $G_k(\alpha',\beta')$. [where G_k is some semantic operation on semantic interpretations]

Illustration: See syntax and semantics of predicate calculus in Section 3.

2. Linguistic Examples.

(See also the Larson chapter)

These are examples of some of the kinds of problems that we will be able to solve after we have developed some of the tools of formal semantics. Some of these, and other, linguistic problems will be discussed in future lectures.

2.1. The structure of NPs with restrictive relative clauses.

Consider NPs such as "the boy who loves Mary", "every student who dances", "the doctor who treated Mary", "no computer which uses Windows". Each of these NPs has 3 parts: a determiner (DET), a common noun (CN), and a relative clause (RC). The question is: Are there semantic reasons for choosing among three different possible syntactic structures for these NPs?

a. Flat structure:

```
              NP
         /    |    \
      DET    CN     RC
       |      |       \
      the    boy    who loves Mary
```

b. "NP - RC" structure: The relative clause combines with a complete NP to form a new NP.

```
              NP
           /      \
         NP        RC
       /    \        \
     DET     CN       \
      |       |        \
     the     boy    who loves Mary
```


c. "CNP - RC" structure: (CNP: common noun phrase: common noun plus modifiers)

```
           NP
         /    \
      DET      CNP
       |      /    \
       |    CNP      RC
       |     |        \
       |     CN        \
       |     |          \
      the   boy     who loves Mary
```

Argument: we can argue that compositionality requires the third structure: that "boy who loves Mary" forms a semantic constituent with which the meaning of the DET combines. We can show that the first structure does not allow for recursivity, and that the second structure cannot be interpreted compositionally. (The second structure is a good structure to provide a basis for a compositional interpretation for *non-restrictive* relative clauses.)

2.2. Anaphora puzzle #1: "Strict and sloppy identity".

The sentence *John loves his wife* has one obvious ambiguity – "his" can mean "John's", or it can have a referent outside the sentence – someone else that we have been talking about, for instance Max. Such an ambiguity is sometimes notated as follows:

(1) John_i loves $\text{his}_{i/j}$ wife. (I.e. *his* can have the same "referential index" as John or a different one.)

But the ambiguity of the (2) raises a further puzzle. (2) involves "VP (Verb Phrase) anaphora": "so does" is anaphoric to the VP of the first sentence, "loves his wife".

(2) John loves his wife and so does Bill. Possible interpretations:
  (i) John loves Max's wife, and Bill loves Max's wife. ("loves his_j wife", he_j = Max)
  (ii) John loves John's wife, and Bill loves John's wife. ("loves his_i wife", he_i = John)
  (iii) John loves John's wife, and Bill loves Bill's wife. (??? "loves self's wife"?)

The contrast between (ii) and (iii) arises even when the first clause seems to unambiguously say that John loves John's wife. Is that first clause actually ambiguous? In what way?

The readings in (i) and (ii) are called "strict identity" readings, and (iii) "sloppy identity". Why "sloppy"? (It's J.R. Ross's term; he was the first to discuss the phenomenon, in Ross (1967)) Because there isn't always exact morpho-syntactic identity; cf. (3).

(3) John can stand on his head, and Mary can too. (= "can stand on *her* head too")

Keenan, Partee, and others argued that so-called "sloppy identity" is strict semantic identity involving *bound variable* readings of pronouns. We'll study this in Lecture 5 and beyond.

3. Formal Semantics in Logic and Linguistics

3.1. English as a Formal Language.

R. Montague 1970, "English as a Formal Language" argued that the syntax and semantics of natural languages could be treated by the same kinds of techniques used by logicians to specify the syntax and model theoretic semantics of formal languages such as the predicate calculus[1].

This is the basic thesis of formal semantics. In these lectures we will clarify its principal points. In the process, we will try to answer the following questions:

- What is a formal language?
- What features of formal languages are most important for formal semantics?
- What are the main differences between "artificial" formal languages and natural language?
- For what parts of "real" natural language semantics can the framework of (existing) formal semantics offer useful tools for linguistic research? For what parts are different tools needed?

3.2. Example. Syntax and semantics of the predicate calculus (PC).

Predicate Calculus is the most well known and in a sense the prototypical example of a formal language. We use it to demonstrate features of formal languages which are most important for us: the notions of model and model-theoretic semantics, and the Principle of Compositionality.
We limit ourselves here to some examples and remarks. More exact definitions are given in **Appendix 1**.

The sentences *John loves Mary* and *Everyone whom Mary loves is happy* can be represented as formulas of PC:

John loves Mary	**love (John, Mary)**
Everyone whom Mary loves is happy	$\forall x$**(love(Mary,** x**)** $\rightarrow$ **happy(**x**))**

Formulas and other *expressions* of PC are built from individual constants (or simply "constants"), (individual) variables, predicate constants (or predicate symbols), logical connectives and quantifiers. Each expression belongs to a certain *type*. The type structure of PC is very simple: individuals, relations of different arities (unary, binary, etc.), and truth-values.

In our examples we use the following expressions:

Expressions	**Syntactic categories**	**Semantic Types**
John, Mary	(individual) constant	individuals
x	variable	individuals
happy	unary predicate constant	unary relations
love	binary predicate constant	binary relations
love (John, Mary)		
love(**Mary,** x)	formulas	truth-values
happy(x)		
$\forall x$(**love**(**Mary,** x) $\rightarrow$ **happy**(x))		

[1] "I reject the contention that an important theoretical difference exists between formal and natural languages. ... In the present paper I shall accordingly present a precise treatment, culminating in a theory of truth, of a formal language that I believe may reasonably be regarded as a fragment of ordinary English. ... The treatment given here will be found to resemble the usual syntax and model theory (or semantics) [due to Tarski] of the predicate calculus, but leans rather heavily on the intuitive aspects of certain recent developments in intensional logic [due to Montague himself]." (Montague 1970b, p.188 in Montague 1974)

Expressions are *interpreted* in *models*. The structure common to all of the models in which a given language is interpreted (the *model structure* for the model-theoretic interpretation of the given language) reflects certain basic presuppositions about the "structure of the world" that are implicit in the language. For PC, any given model structure consists of the set of truth-values {0,1}, a *domain* D which is some set of objects (or entities), and some n-ary relations on this set.

A *model,* or *interpreted model,* consists of a model structure plus a ("lexical", or "basic") interpretation function I which assigns semantic values to all constants.

M = <D, I>

An *interpretation* $\| \|^M$, built up recursively on the basis of the basic interpretation function I, assigns to every expression α its *semantic value* $\|\alpha\|^M$ in a given model **M.** (More precisely, $\|\alpha\|^{M,g}$.) These semantic values must correspond to the types of the expressions. Thus, in our examples to the individual constants **John** and **Mary** are assigned certain objects, individual variables take their values in the set of objects (entities), to the predicate constant **love** is assigned a binary relation $\|\mathbf{love}\|^M$, and to the predicate constant **happy**, a unary relation (property) $\|\mathbf{happy}\|^M$. Formulas receive truth values. The formula **love (John, Mary)** is true in the model **M** if the pair of objects corresponding to the constants **John** and **Mary** belongs to the relation $\|\mathbf{love}\|^M$.

The formula $\forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$ is true in **M** iff:
for every object d in the domain,
$d \in \|\mathbf{happy}\|^M$ if $<\|\mathbf{Mary}\|^M, d> \in \|\mathbf{love}\|^M$.

Restating the last statement more carefully and more generally requires talking about semantic values relative to a model *and an assignment g of values to variables.*
The notation g[d/*x*] means: The variable assignment which is identical to g except for the (possible) difference that g[d/*x*] assigns the individual d to the variable *x*.

> The complication of needing to talk about g[d/*x*] comes from formulas with more than one variable, like:
> $\forall x \exists y(\mathbf{love}(y, x) \rightarrow \mathbf{happy}(x))$ and
> $\exists y \forall x(\mathbf{love}(y, x) \rightarrow \mathbf{happy}(x))$. – See the practice exercises at the end.

So let us restate more carefully, according to the semantics given in **Appendix 1**, the truth conditions for the formula: $\forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$:

$\|\forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))\|^{M,g} = 1$ iff :

for each d in D,
if $<\|\mathbf{Mary}\|^{M,g[d/x]}, \|x\|^{M,g[d/x]}> \in \|\mathbf{love}\|^{M,g[d/x]}$, then $\|x\|^{M,g[d/x]} \in \|\mathbf{happy}\|^{M,g[d/x]}$.

For each constant α, $\|\alpha\|^{M,g[d/x]} = I(\alpha)$.

And for any variable ***x***, $\|x\|^{M,g[d/x]} = g[d/x](x) = d$. So the condition above is equivalent to:

iff: for each d in D,
if $<I(\mathbf{Mary}), d> \in I(\mathbf{love})$, then $d \in I(\mathbf{happy})$.

Example

Let us consider a very simple PC language which has (as in the formulas above) only two constants **John** and **Mary** and two predicate symbols **love** (binary) and **happy** (unary).
Let us consider two models, **M1** and **M2**:

M1 = $<D, I_1>$, D = {j,m},
I_1(**John**) = j, I_1(**Mary**) = m,
I_1(**love**) = {$<j,j>,<j,m>,<m,m>,<m,j>$}, I_1 (**happy**) = {j,m},

M2 = $<D, I_2>$, D = {j,m},
I_2(**John**) = j, I_2(**Mary**) = m,
I_2(**love**) = {$<j,j>,<m,j>$}, I_2 (**happy**) = {m}.

It is easy to see that both formulas **love** (**John, Mary**) and **love** (**Mary, John**) are true in **M1** but only the second one is true in **M2**.

The formula $\forall \boldsymbol{x}(\mathbf{love}(\mathbf{Mary}, \boldsymbol{x}) \rightarrow \mathbf{happy}(\boldsymbol{x}))$ is true in **M1.** But it is false in **M2,** since for the evaluation g such that g($\boldsymbol{x}$) = j we have $\|\mathbf{love}(\mathbf{Mary}, \boldsymbol{x})\|^{M2,g} = 1$ and $\|\mathbf{happy}(\boldsymbol{x})\|^{M2,g} = 0$.

The semantics of PC illustrates the Principle of Compositionality.
As we know the infinite set of formulas of PC are built from terms (individual variables and constants) and predicate symbols by recursive syntactic rules (rules R1—R8 in Appendix 1). The semantics of these formulas – their interpretation in every given model -- is defined by semantic rules S1 – S8, which correspond in a direct way to the syntactic rules. The semantics of the whole is based on the semantics of parts by means of this pairing of semantic interpretation rules with syntactic formation rules. **See trees 1 and 2 in the "practice exercise" in APPENDIX 2.** This is a very important feature of every formal language -- *The Principle of Compositionality* – and it is natural to think that this principle holds also for natural language.

3.3. "Logical form", or semantically relevant syntax.

What is the interpretation of "every student"? There is no appropriate syntactic category or semantic type in predicate logic. Inadequacy of 1st-order predicate logic for representing the semantic structure of natural language.

<u>**Categories of PC**</u>:		<u>**Categories of NL**</u>:
Formula	-	Sentence
Predicate	-	Verb, Common Noun, Adjective
Term		
Constant	-	Proper Noun
Variable	-	Pronoun (he, she, it)
==========		
(no more)	-	Verb Phrase, Noun Phrase, Common Noun Phrase, Adjective Phrase, Determiner, Preposition, Prepositional Phrase, Adverb, …

In the next lectures, we will see how a logic built on a richer type theory including the tools of the lambda-calculus can provide a richer formal semantics that can more adequately represent the structure of natural language semantics in a compositional way.
==

APPENDIX 1. Syntax and semantics of the predicate calculus (PC).

<u>**SYNTAX**</u>**.**
<u>**Syntactic Categories**</u>**:** terms (Term), 1-place predicates (Pred-1), 2-place predicates (Pred-2), ..., n-place predicates (Pred-n), formulas (Form).

<u>**Basic Expressions**</u>**:**
Basic Term(s): (i) (individual) variables: $\boldsymbol{x, y, z, x_1, y_1, z_1, x_2}$, ...
(ii) (individual) constants: **a,b,c, $\mathbf{a_1}$, .John, Mary, ...**
Basic Pred-1: **run, walk, happy, calm,** ...
Basic Pred-2: **love, kiss, like, see, ...**
...
Basic Form(ulas): — (none)

<u>**Syntactic Rules**</u>**:**
R1: If $P \in$ Pred-1 and $T \in$ Term, then $P(T) \in$ Form.
R2: If $R \in$ Pred-2 and $T_1, T_2 \in$ Term, then $R(T_1, T_2) \in$ Form.
More general rule: If $R \in$ Pred-n and $T_1, ...,T_n \in$ Term, then $R(T_1, ...,T_n) \in$ Form
R3: If $\varphi \in$ Form, then $\neg\varphi \in$ Form.
R4: If $\varphi \in$ Form and $\psi \in$ Form, then $(\varphi \,\&\, \psi) \in$ Form.
R5: If $\varphi \in$ Form and $\psi \in$ Form, then $(\varphi \vee \psi) \in$ Form.
R6: If $\varphi \in$ Form and $\psi \in$ Form, then $(\varphi \rightarrow \psi) \in$ Form.
R7: If v is a variable and $\varphi \in$ Form, then $\forall v\varphi \in$ Form.
R8: If v is a variable and $\varphi \in$ Form, then $\exists v\varphi \in$ Form.

<u>**SEMANTICS**</u>**.**
<u>**Model structure**</u>**:**
Domain D of entities (individuals)
Truth values {True, False} or {1,0}
I: Interpretation function which assigns semantic values to all constants (in Term and in Pred-1, Pred-2, ... Pred-n)
M = $<D, I>$
Set G of assignment functions *g*, functions from variables to D.

<u>**Semantic Types assigned to Syntactic Categories**</u>**:**
Term: entities, individuals. The semantic values of this type are the members of D.
Pred-1: sets (of entities). Semantic values of this type are members of $\wp$(D).
($\wp$(D) is the power set (the set of all subsets) of D).
Pred-2: relations between entities (sets of pairs). Values: members of $\wp(D\times D)$.
Pred-n: n-place relations; sets of n-tuples of entities.Values: members of $\wp(D\times ...\times D)$.
Form: Truth values. Values: members of {0,1}.

<u>**Semantic interpretation relative to M, *g***</u>**:**
We use the notation $\|\varphi\|^{M,g}$ for the semantic value of an expression φ relative to **M**, *g*.

<u>**Basic Expressions ("lexical semantics")**</u>**:**
A. If α is a variable, then $\|\alpha\|^{M,g} = g(\alpha)$.
B. If α is a constant, then $\|\alpha\|^{M,g} = I(\alpha)$.

Semantic Rules ("semantics of syntax"):

S1: If $P \in$ Pred-1 and $T \in$ Term, then $\|P(T)\|^{M,g} = 1$ iff $\|T\|^{M,g} \in \|P\|^{M,g}$.

S2: More general rule: If $R \in$ Pred-n and $T_1, ..., T_n \in$ Term, then $\|R(T_1, ..., T_n)\|^{M,g} = 1$ iff $<\|T_1\|^{M,g}, ..., \|T_n\|^{M,g}> \in \|R\|^{M,g}$.

S3: If $\varphi \in$ Form, then $\|\neg\varphi\|^{M,g} = 1$ iff $\|\varphi\|^{M,g} = 0$.

S4: If $\varphi, \psi \in$ Form, then $\|(\varphi \,\&\, \psi)\|^{M,g} = 1$ iff $\|\varphi\|^{M,g} = 1$ and $\|\psi\|^{M,g} = 1$.

S5: If $\varphi, \psi \in$ Form, then $\|(\varphi \vee \psi)\|^{M,g} = 1$ iff $\|\varphi\|^{M,g} = 1$ or $\|\psi\|^{M,g} = 1$.

S6: If $\varphi, \psi \in$ Form, then $\|(\varphi \rightarrow \psi)\|^{M,g} = 1$ iff $\|\varphi\|^{M,g} = 0$ or $\|\psi\|^{M,g} = 1$.

S7: If v is a variable and $\varphi \in$ Form, then $\|\forall v\varphi\|^{M,g} = 1$ iff for all $d \in D$, $\|\varphi\|^{M,g[d/v]} = 1$.

S8: If v is a variable and $\varphi \in$ Form, then $\|\exists v\varphi\|^{M,g} = 1$ iff there is a $d \in D$ such that $\|\varphi\|^{M,g[d/v]} = 1$.

[**The notation g[d/*x*] means**: The variable assignment which is identical to g except for the (possible) difference that g[d/*x*] assigns the individual d to the variable *x*.]

Truth: Some formulas are true independent of the choice of assignment; those can be called true relative to just **M**, i.e. simply true on the given interpretation.

If $\varphi \in$ Form, then: $\|\varphi\|^{M} = 1$ iff for all assignments g, $\|\varphi\|^{M,g} = 1$.
$\|\varphi\|^{M} = 0$ iff for all assignments g, $\|\varphi\|^{M,g} = 0$.
Otherwise $\|\varphi\|^{M}$ is undefined.

REFERENCES.

(Some of these will be referred to in later lectures.)

Bach, Emmon (1986) "Natural language metaphysics", in Barcan Marcus et al (eds), *Logic, Methodology and Philosophy of Science VII*, Elsevier Publ., 573-595.
Bach, Emmon (1989), *Informal Lectures on Formal Semantics*. State University of New York Press, Albany.
Dowty, David (1979) *Word Meaning and Montague Grammar*, Dordrecht:Reidel.
Dowty, David, R. Wall and S. Peters (1981), *Introduction to Montague Semantics*, Dordrecht: Reidel.
Frege, Gottlob (1892) "Ueber Sinn und Bedeutung", Zeitschrift fuer Philosophie und philosophische Kritik 100, 25-50. Translated as "On sense and reference", in P.T. Geach and M.Black, eds., *Translations form the Philosophical Writings of Gottlob Frege*. Oxford: Blackwell (1952), 56-78.
Gamut, L.T.F. (1991), *Logic, Language, and Meaning. Vol I: Introduction to Logic; Vol II: Intensional Logic and Logical Grammar*. University of Chicago Press, Chicago and London.
R. Larson (1995) Semantics. Chapter 12 in L. Gleitman and M.Liberman, eds. *An Invitation to Cognitive Science, Vol I: Language*. (D. Osherson, general editor), pp 361-380. https://udrive.oit.umass.edu/partee/Semantics_Readings/Larson_chpt.12.pdf
Lewis, David (1970) "General semantics" Synthese 22, 18-67; reprinted in D.Davidson and G.Harman (eds.), *Semantics of Natural Language*. Dordrecht: Reidel (1972), 169-218.
Montague, Richard (1970b) "English as a Formal Language", in B. Visentini et al, eds. *Linguaggi nella Societa e nella Tecnica*. Milan: Edizioni di Comunità; reprinted in Montague (1974) 188-221.
Montague, Richard (1970c) "Universal Grammar", Theoria 36, 373-398; reprinted in Montague (1974) 222-246.
Montague, R. (1973) "The Proper Treatment of Quantification in Ordinary English," in K.J.J. Hintikka, J.M.E. Moravcsik, and P. Suppes, eds., *Approaches to Natural Language*, Reidel, Dordrecht, 221-242, reprinted in Montague (1974) 247-270.

Montague, Richard (1974) *Formal Philosophy: Selected Papers of Richard Montague*. Edited and with an introduction by Richmond Thomason, New Haven: Yale Univ. Press.
Partee, Barbara H. (1984) "Compositionality", in F. Landman and F. Veltman, eds., *Varieties of Formal Semantics*. Foris Pubs., Dordrecht, 281-311.
Partee, Barbara (1995) "Lexical Semantics and Compositionality", in *Invitation to Cognitive Science*, 2nd edition. Daniel Osherson, general editor; in Part I: Language, Lila Gleitman and Mark Liberman, eds. MIT Press, Cambridge, pp. 311-360.
Partee, Barbara (1996) "The development of formal semantics in linguistic theory", in Shalom Lappin, ed., *The Handbook of Contemporary Semantic Theory*, Blackwell Handbooks in Linguistics Series, Oxford: Blackwell, 11-38.
Partee, Barbara H. 1999. "Semantics" in R.A. Wilson and F.C. Keil, eds., *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge, MA: The MIT Press. 739-742. https://udrive.oit.umass.edu/partee/Semantics_Readings/Partee%20MITECS%20Semantics.pdf
Partee, Barbara H. In press. Formal semantics. In *The Cambridge Encyclopedia of the Language Sciences*, ed. Patrick Colm Hogan. Cambridge: Cambridge University Press. https://udrive.oit.umass.edu/partee/Partee_Formal_Semantics_Camb_Encyc.pdf
Partee, B., A. ter Meulen, and R.E. Wall (1990) *Mathematical Methods in Linguistics*, Dordrecht: Kluwer Academic Publishers.
Portner, Paul, and Partee, Barbara H. eds. 2002. *Formal Semantics: The Essential Readings*. Oxford: Blackwell Publishers.
Ross, John R. 1967 [1986]. Constraints on variables in syntax, MIT: Ph.D. dissertation.

"HOMEWORK" No. 0: Participant Questionnaire ["Anketa"]

Please answer the following questions for me; answers can be in Russian except for question 6. Please write clearly and legibly. Short answers: no more than 2 pages total. **Anketa due at the time of Lecture 2. Bring it to class or e-mail it to me at** partee@linguist.umass.edu .

1. Your name
   -- Your e-mail address:
   -- Website address if you have one:
2. Your "status" — 2nd,3rd, ..., year student in the linguistics program, or other.
3. Do you expect to take this course for credit ("ocenka" or "začet")? (You can change your mind later, that's ok.)
4. (Briefly) How much / what kind of the following have you studied?
   a. Semantics
   b. Syntax
   c. Logic
   d. Mathematics
5. a. Estimate your knowledge of English (poor, fair, good, very good) in
   (i) reading; (ii) writing; (iii) listening; (iv) speaking.
   b. Were you able to understand most of my lecture?
   c. Do you have any suggestions that will make it easier for you to understand me?
6. What other languages do you have some knowledge of? What languages have you done some linguistic work on, or are planning to in the future? Have you been on linguistic expeditions (which languages?)?
7. Write two or three sentences in English about what areas of linguistics you are most interested in and what you might like to do for a future career.
8. Write one or two sentences in Russian about why you are taking this course and what you hope to learn from it.

APPENDIX 2: For Seminar Feb 9: A Practice Homework

(to do together in class, not to turn in)

Background:

1. We will first work on the formula $\forall$**x happy(x)**, and work out its interpretation with respect to the model M2, working compositionally. We'll do it basically the same way as #2 below, but just on the blackboard.

2. Below you will find a syntactic "derivation" tree for the formula $\forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$, which expresses the same proposition as the English sentence *Everyone who Mary loves is happy.* That is followed by a derivation of the truth-conditions of the formula according to the compositional semantic rules of the predicate calculus. Each line is annotated to identify what semantic rule was applied in the derivation of that line, and what node of the syntactic derivation tree it corresponds to. (The problem you are asked to solve is stated after all of that.)

Tree 1.

$\forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$ **, Form, R7**

x $(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$ **, Form, R6**

love(Mary, ***x*****), Form, R2**   **happy(*x*), Form, R1**

love, Pred-2, Basic | ***x*, T, Basic**   **happy, Pred-1, Basic**   ***x*, T, Basic**

Mary, T,Basic

Annotated semantic derivation of truth conditions:

1. $\| \forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x)) \|^{M,g} = 1$ iff for each d in D,

$\| \mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x) \|^{M,g[d/x]} = 1$. **By rule S7 at the "R7" node.**

2. That will hold iff for each d in D,

$\| \mathbf{love}(\mathbf{Mary}, x) \|^{M,g[d/x]} = 0$ or $\| \mathbf{happy}(x) \|^{M,g[d/x]} = 1$. **By rule S6 at the "R6" node.**

3. That will hold iff for each d in D,

if $< \| \mathbf{Mary} \|^{M,g[d/x]}, \| x \|^{M,g[d/x]} > \in \| \mathbf{love} \|^{M,g[d/x]}$, then $\| x \|^{M,g[d/x]} \in \| \mathbf{happy} \|^{M,g[d/x]}$.

By rule S2 at the R2 node and by S1 at the R1 node.

4. And that will hold iff for each d in D,

if $< \| \mathbf{Mary} \|^{M,g[d/x]}, d > \in \| \mathbf{love} \|^{M,g[d/x]}$, then $d \in \| \mathbf{happy} \|^{M,g[d/x]}$.

By rule A (for variables) at the two *x* nodes.

5. I.e., if $<I(\mathbf{Mary}), d> \in I(\mathbf{love})$, then $d \in I(\mathbf{happy})$.

By rule B (for constants) at the nodes for Mary, love, happy.

If we then annotate the syntactic tree above to also show the semantic rule applied at each step, we can see a perfect match between syntactic and semantic rules in the derivation of the form and meaning of the formula.

Tree 2. $\forall x(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$ **, Form, R7, S7**

x $(\mathbf{love}(\mathbf{Mary}, x) \rightarrow \mathbf{happy}(x))$ **, Form, R6, S6**

love(Mary, *x*), Form, R2, S2   **happy(*x*), Form, R1, S1**

love, Pred-2, Basic, B | ***x*, T, Basic, A**   **happy, Pred-1, Basic, B**   ***x*, T, Basic, A**

Mary, T,Basic, B

3. <u>Exercise: (to do in seminar together)</u> This one gives more practice with using *g*.

The predicate logic formula $\forall x(\exists y\,\mathbf{love}(x, y) \rightarrow \mathbf{happy}(x))$ is equivalent to the English sentence *Everyone who loves someone is happy.*

(b) **<u>Draw a syntactic tree</u>** (analogous to Tree 1 above) which shows how that formula is built up from its parts according to the syntactic rules of the predicate calculus (in the Appendix above).

(c) **<u>Give each node a label</u>** that identifies both the syntactic category of the expression it dominates and the number of the syntactic rule by which its immediate constituents were combined (or "Basic", if that node dominates a basic expression.)

(d) **<u>Work out the truth-conditions</u>** of the formula according to the semantic rules of the predicate calculus, analogous to the step-by-step derivation of truth conditions for the example above (see **NOTE** below). **<u>Annotate each line</u>** by identifying the semantic rule that was applied anywhere within that line (show where), and the node of the tree to which it corresponds. (According to the principle of compositionality, there should be a perfect match between syntactic rule and semantic rule applied at each node.)

(e) In addition, **<u>further annotate the syntactic tree</u>** by adding to the label of each non-terminal node the number of the *semantic* rule which was used to combine the meanings of the daughter-node expressions to get the meaning of the whole expression dominated by that node. For nodes dominating basic expressions, indicate whether the semantic rule to use is Rule A or Rule B. (If you've done it right, there should be a perfect correspondence between syntactic rules and semantic rules applied at a given node, as in Tree 2 above.)

NOTE: What happens when you are working with g[d/x] and you need to make a further substitution, e.g. for the variable *y*? Answer: you need to consider another arbitrary element d' of D, and modify the assignment again, resulting in g[d/x][d'/y]: the assignment just like g except it assigns d to x and d' to y.