The Semantics of Phi: Introduction

Yasutada Sudo
Institut Jean-Nicod
yasutada.sudo@gmail.com

ACTL Day 1
June 24, 2013
UCL

Overview of the Course

- Phi-features appear on many items, most notably **pronouns**, **verbs**, **nouns**, **adjectives**, etc.
- In this course, we'll mostly focus on phi-features encoded in **pronouns**. But on the last day, we'll discuss consequences of our analysis of pronouns on phi-features appearing on other elements.
- Let's take a very simple sentence:

(1) **She** is Russian.

- *She* has a gender feature, [feminine] (among other features)
- It's unquestionable that [feminine] has a semantic effect:

(2) (Pointing to Lisa)
She is Russian.

(3) (Pointing to Igor)
#**She** is Russian.

> - **Q1**: How are phi-features on pronouns interpreted? How do we capture the above contrast?

- It has been suggested that some occurrences of phi-features are not semantically interpreted, but rather are morphological reflexes of syntactic agreement.
  - Phi-features on bound pronouns

(4) a. Only Jesse likes **herself**.
b. No student brought lunch with **her** today.

  - Grammatical gender

(5) *French*
Pierre a une chaise Louis XV. **Elle** est belle.
Pierre has a.FEM chair Louis XV. she.FEM is beautiful.FEM
'Pierre has a Louis the 15th chair. It is beautiful.'
(adapted from Wechsler & Zlatić 2003:198)

(6) a. John bought a boat. He named **her** 'Elizabeth'.
b. At one time, according to Sir George H. Darwin, the Moon was very close to the Earth. Then the tides gradually pushed **her** far away: the tides that the Moon **herself** causes in the Earth's waters, where the Earth slowly loses energy.
('The Distance of the Moon' by Italo Calvino)

(7) *German*
Jedes Kind definierte **seine** Ziele.
every.NEU child defined its.NEU goals
'Every child defined his or her goals.' (Angelika Kratzer, p.c.)

- **Q2**:Are these occurrences of phi-features really not semantically interpreted? If so, how do we know which occurrences are semantically interpreted and which are not?

- After discussing these questions for pronouns, we'll look at phi-features on other things.

- **Q3**: Are phi-features on other items semantically interpreted at all? Do we have to say that phi-features on pronouns and phi-features on other items are semantically interpreted in different ways?

  – **Phi-Features on Predicates:** Agreement appearing on predicates are usually considered to have no semantic content.

(8) a. John walk**s** to school.
b. I **am** hungry.

(9) *Mayali*
a. Na-meke bininj **na**-kimuk.
MASC-that man.MASC MASC-big
'That man is big.'
b. Ngal-eke daluk **ngal**-kimuk.
FEM-that woman.FEM FEM-big
'That woman is big.' (Baker 2008:16)

  ∗ That phi-features on predicates are not semantically relevant is an old idea. Although has occasionally been questioned, it's the predominant view today.
  ∗ But should we really be convinced that the phi-features on predicates have no semantic consequences?

  – There are phi-features on predicates that seem to have some semantic consequences (% indicates dialectal variation).

(10) *English Collective Nouns*
a. The committee **is** old.
(i) The committee was founded a long time ago.
(ii) The committee members are old.
b. %The committee **are** old.
(i) The committee members are old.
(adapted from Pearson 2011)

(11) *Bulgarian 1pl Agreement with Quantifiers*
a. Povecheto visoki zheni **imame** hubavi drehi.
most tall women have.1pl nice clothes
'Most tall women (including the speaker) have nice clothes.'
b. Nikoi visoki zheni **njamame** hubavi drehi.
no tall women not.have.1pl nice clothes
'No tall women (including the speaker) have nice clothes.'
(adapted from Schlenker 2006:520, fn.16)

  ∗ Should we conclude that at least in some cases phi-features on predicates are interpreted? If so, why not in all cases?
  ∗ Or should we pursue a syntactic account of the above data (e.g. there are morphologically invisible occurrences of agreeing features in the subjects)?

- **Nominal Concord:** Which phi-features within a single DP are interpreted?

(12) *Italian*

Abbiamo	**una**	**cuoca**	**italiana**.
have.3PL	a.FEM.SG	chef.FEM.SG	italian.FEM.SG

'We have a female Italian chef.' (adapted from Percus 2011:168)

More exotic examples from an Australian language with four genders: masculine, feminine, neuter and vegetable.

(13) *Wambaya*

a.

Ngankiyaga	**bungmanyani**	ngiya-ngajbi	**yaniyaga**	**darranggu**.
that.FEM.ERG	woman.FEM.ERG	she-saw	that.NEU.ACC	tree.NEU.ACC

'That woman saw that tree.'

b.

Ninkiyaga	**bungmanyini**	gina-ngajbi	**mamiyaga**	**jigama**.
that.MASC.ERG	man.MASC.ERG	he-saw	that.VEG.ACC	yam.VEG.ACC

'That man saw that yam.' (Sag & Wasow 1999:100)

It's clear that at least one occurrence of these phi-features should be interpreted, but which one?

- **Tentative plan (very ambitious):**
  - **Day 1**: Theoretical Background
    - Very brief introduction to truth-conditional semantics
    - Pronouns and quantifiers in truth-conditional semantics
  - **Day 2**: Phi-Features on Pronouns and Presuppositions
    - Phi-Features as Presupposition Triggers
    - Unmarked Features and Anti-Presupposition
    - Grammatical gender
  - **Day 3**: Problem of Bound Pronouns
    - Minimal Pronouns and Feature Transmission
    - Semantic Account
  - **Day 4**: Number Features on Pronouns
  - **Day 5**: Phi-features on Verbs, Adjectives and Nouns
- We'll discuss the semantics of phi-features from the viewpoint of truth-conditional semantics.

1 Quick Introduction to Truth-Conditional Semantics

1.1 Semantic Intuitions

- We have several kinds of semantic intuitions. The core intuitions for truth-conditional semantics are the following:
  - **Truth-Conditional Judgment** (about sentences)

    If you are given sufficient information about a situation s, you can tell if a given declarative sentence ϕ is true or false in s.

    (14) SITUATION: We know that John lives in Paris with his wife and three kids.
    a. 'John lives in France' is true.
    b. 'John lives in Tokyo' is false.
    c. 'John has a wife' is true.

    Or to put it differently, you know what to check in s in order to evaluate the truth of ϕ.
  - **Referential Judgment** (about referring expressions)

    You can tell who proper names and definite descriptions should refer to in situation s, provided that they have referents in s.

    (15) SITUATION: John Smith is the only person around us whose name is John. He is dating a girl named Mary.
    a. 'John' refers to John Smith.
    b. 'The man who is dating Mary' refers to John Smith.
    c. 'The girl who Mary is dating' refers to Mary.
- These are part of our linguistic knowledge, and our main empirical data in truth-conditional semantics. We'll construct a theory of these semantic intuitions in relation to syntax and morphology.
- They are arguably not everything there is to the 'meaning' of the sentences. But for the purposes of this course, we'll unfortunately have to ignore many other aspects of meaning (One exception is presupposition, which we'll discuss tomorrow). Of course this doesn't mean that these other kinds of meaning are unimportant.
- We'll also only discuss declarative sentences in this course, for the sake of brevity.

1.2 Sentence Meanings as Truth-Conditions

- How do we model the truth-conditional meaning of a declarative sentence ϕ?
  - If we are given a situation s, we can tell if ϕ is true or false in s.
  - Let's write $[\![\phi]\!]^s$ for **the truth-conditional meaning of ϕ evaluated in situation s**.
  - $[\![\phi]\!]^s$ is true or false. We'll use truth-values: 1 (TRUE) and 0 (FALSE).
  - We say ϕ **denotes** 1 (or 0) in s.

(16) For any situation s,

$$[\![\textbf{John smokes Marlboros}]\!]^s = \begin{cases} 1 & \text{if John smokes Marlboros in } s \\ 0 & \text{if John doesn't smoke Marlboros in } s \end{cases}$$
$$= 1 \text{ iff John smokes Marlboros in } s$$

('iff' stands for 'if and only if')

- Some more examples:

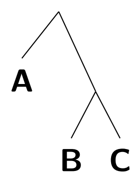
(17) $[\![$**It is raining in London**$]\!]^{s'}$ $= 1$ iff it is raining in London in s'.

(18) $[\![$**Yasu bought a laptop**$]\!]^{s}$ $= 1$ iff Yasu obtained a laptop in exchange for money in s.

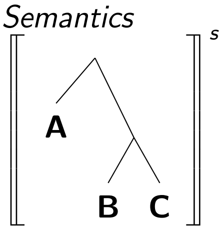
1.3 Sub-Sentential Meanings and the Compositionality Principle

- Syntax-semantics interface:
  - The syntactic component of grammar generates a hierarchically organized representation, called a **Logical Form (LF)**.
  - The semantic component takes an LF representation, and computes the truth-conditional meaning of it.
  - $[\![\ ]\!]^{s}$ is a mapping (more precisely a function) from LF representations to truth-conditional meanings.

Syntax



Semantics

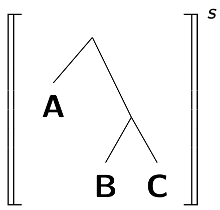


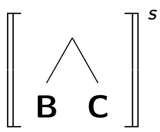
= blah blah blah

  - In this sense, semantics is dependent on syntax. And because of this dependency, our semantic theory often has something to say about syntax.
- The interpretation in the semantic component proceeds in a compositional fashion.

(19) **Compositionality Principle:**
The meaning of a complex phrase is determined by the syntax and semantics of its parts.

Schematically,

we will know what 

is, once we know what $[\![$**A**$]\!]^{s}$ and 

are.

(20) $[\![$**John smokes**$]\!]^{s}$ is determined by:
a. $[\![$**John**$]\!]^{s}$,
b. $[\![$**smokes**$]\!]^{s}$, and
c. how 'John' and 'smokes' are syntactically combined at LF.

- The semantic component has of two major parts:

1. **Lexicon**: A list of the meanings of individual terminal nodes (morphemes, words, etc.).

(21) a. $[\![$**B**$]\!]^{s}$ = blah blah blah
b. $[\![$**C**$]\!]^{s}$ = yadda yadda yadda
etc.

2. **Compositional Rules**: Instructions as to how to combine the meanings of two components according to the way in which they are syntactically combined.

(22) $\left[\!\left[\begin{array}{c} \wedge \\ \textbf{B} \quad \textbf{C} \end{array} \right]\!\right]^{s}$ = the result of combining $[\![\textbf{B}]\!]^{s}$ and $[\![\textbf{C}]\!]^{s}$ in such-and-such ways

- The goal is to assign truth-conditional meanings to any well-formed phrases and sentences, so we will be able to deal with an infinite number of complex linguistic objects (in fact, we have semantic intuitions about all of them).
- Let's consider the following simple LF representation for the sentence 'John smokes'.

(23)

John **smokes**

- To simplify the discussion, we'll ignore functional projections like T, *v*, D, etc.
- We'll also assume that branching is always binary (for the sake of simplicity).
- By assumption, syntactic labels are semantically irrelevant, and often often omitted.

- What is the meaning of '**John**'?
  - '**John**' should refer to the individual John.
  - To make things easy, let's just assume that John exists in all situations.

(24) $[\![\textbf{John}]\!]^{s}$ = John (for any situation s).

- What should be the meaning of '**smokes**'?
  - Recall that we know the meaning of the entire sentence '**John smokes**':

(25) $[\![\textbf{John smokes}]\!]^{s} = 1$ iff John smokes in s

  - Proposal: $[\![\textbf{smokes}]\!]^{s}$ is a **function** that takes $[\![\textbf{John}]\!]^{s}$ and yields the meaning of $[\![\textbf{John smokes}]\!]^{s}$.

FUNCTION AND LAMBDA NOTATION

- A **function** takes an input and returns a unique output.

f is a *function* from set A to set B iff for any $a \in A$, f applied to a returns a unique $b \in B$ (written $f(a) = b$).

- We often use the **lambda notation** for convenience. A description of a function has three parts:

1. 'λ' followed by a variable, which stands for the input of the function
2. A period
3. A description of the output

$$\lambda \underbrace{\alpha}_{\text{INPUT}} . \underbrace{\beta}_{\text{OUTPUT}}$$

(26) Some examples
a. Identity function: $\lambda x.\ x$

the identity function takes an input and returns the same thing.
e.g. $[\lambda x.x](a) = a$

b. Square function: $\lambda n.\ n^2$
e.g. $[\lambda n.\ n^2](3) = 3^2 = 9$

c. Celsius-Fahrenheit conversion function: $\lambda C.\ \frac{9}{5}C + 32$

- Functions can return functions as outputs:

(27) $\lambda x.\lambda y.x + y$
(this function takes an input and returns another function $[\lambda y.x + y]$)
a. $[\lambda x.\lambda y.x + y](5) = \lambda y.x + 5$
b. $[\lambda x.\lambda y.x + y](17) = \lambda y.x + 17$

– According to our proposal,

(28) $\llbracket \textbf{smokes} \rrbracket^s(\llbracket \textbf{John} \rrbracket^s) = \llbracket \textbf{John smokes} \rrbracket^s = 1$ iff John smokes in s

So, $\llbracket \textbf{smokes} \rrbracket^s$ takes any individual x and returns 1 just in case x smokes in s.

(29) $\llbracket \textbf{smokes} \rrbracket^s = \lambda x.$ 1 iff x smokes in s

- For the moment, we assume that for any syntactically well-formed complex phrase, one of its daughters denotes a function, and the other one denotes its argument. We state this as a compositional rule, and call it **Functional Application**:

(30) *Functional Application*
$$\left\llbracket \; [\ \textbf{A}\ \ \textbf{B}\ ] \; \right\rrbracket^s = \llbracket \textbf{A} \rrbracket^s(\llbracket \textbf{B} \rrbracket^s) \text{ or } \llbracket \textbf{B} \rrbracket^s(\llbracket \textbf{A} \rrbracket^s), \text{ whichever is appropriate.}$$

- It is often useful to talk about the **semantic types** of the inputs and outputs.
  – Individuals like John Smith, my laptop, etc. are of type e.
  – Truth-values (1 and 0) are of type t.
  – Functions from individuals to truth-values, e.g. $\llbracket \textbf{smokes} \rrbracket^s$, is of type $\langle e, t\rangle$. (i.e. its input is of type e and its output is of type t).
- We sometimes annotate the types of variables with subscripts:

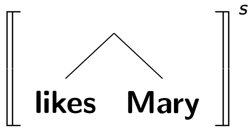
(31) $\llbracket \textbf{smokes} \rrbracket^s = \lambda x_e.$ 1 iff x smokes in s

- An example with a transitive predicate:

(32) $\llbracket \textbf{likes} \rrbracket^s = \lambda y_e.[\lambda x_e.$ 1 iff x likes y in $s]$

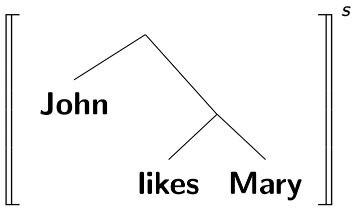
This is a function of type $\langle e, \langle e, t\rangle\rangle$, i.e. a function that takes an individual and returns a function of type $\langle e, t\rangle$.

The meaning of '**likes Mary**' is, therefore, going to be a function of type $\langle e, t\rangle$:

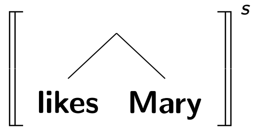


$$= [\![\textbf{likes}]\!]^s([\![\textbf{Mary}]\!]^s)$$
$$= [\lambda y_e.[\lambda x_e.\ 1 \text{ iff } x \text{ likes } y \text{ in } s]](\text{Mary})$$
$$= \lambda x_e.\ 1 \text{ iff } x \text{ likes Mary in } s$$

From this, we can compute the meaning of '**John likes Mary**':



$$=$$



$$([\![\textbf{John}]\!]^s)$$
$$= [\lambda x_e.\ 1 \text{ iff } x \text{ likes Mary in } s](\text{John})$$
$$= 1 \text{ iff John likes Mary in } s$$

INTERIM SUMMARY

- Declarative sentences denote truth-values, 1 (TRUE) or 0 (TRUE), in a given situation s.

(33) $[\![\textbf{John smokes}]\!]^s = 1$ iff John smokes in s

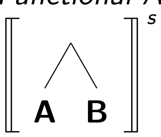
- **Compositionality Principle:**
  The meaning of a complex phrase is determined by the syntax and semantics of its parts.
- Two components:

1. **Lexicon**

(34) a. $[\![\textbf{John}]\!]^s = \text{John}$ (type e)
b. $[\![\textbf{smokes}]\!]^s = \lambda x_e.\ 1 \text{ iff } x \text{ smokes}$ (type $\langle e, t\rangle$)
c. $[\![\textbf{likes}]\!]^s = \lambda y_e.[\lambda x_e.\ 1 \text{ iff } x \text{ likes } y \text{ in } s]$ (type $\langle e, \langle e, t\rangle\rangle$)

2. **Compositional Rule**

(35) *Functional Application*



$$= [\![\textbf{A}]\!]^s([\![\textbf{B}]\!]^s) \text{ or } [\![\textbf{B}]\!]^s([\![\textbf{A}]\!]^s), \text{ whichever is appropriate.}$$

2 Pronouns and Quantifiers

- Personal pronouns in English

(36) *3rd Person Pronouns in English*
a. he, him, his
b. she, her
c. it, its

d. they, them, their

(37) *1st and 2nd Person Pronouns in English*
a. I, me, my
b. you, your

- We'll analyze 3rd person pronouns first.
- There are two main uses of (3rd person) pronouns.
  - **Referential/Free use:**

    (38) John came in. **He** was drunk.

    'He' refers to John, a particular person.
  - **Bound use:**

    (39) No boy brought lunch with **him** today.

    'Him' does not refer to a particular person, but is dependent on the meaning of 'no boy'.
- We want a semantics of pronouns that captures both uses. We'll enrich our semantics introduced above to achieve this.
- Today, we'll ignore the semantic effects of phi-features. We'll discuss them in detail tomorrow.

2.1 Referential/Free Pronouns

- An analysis of referential pronouns needs to capture the **flexibility of reference**:
  - A referential pronoun denotes/refers to a specific individual (it's of type e).
  - E.g., in (38), we want to say that the meaning of 'he' is the individual John.
  - But simply stating $[\![\textbf{he}]\!]^s$ = John will not be good, because in a different context, the same word 'he' refers to some other person.

    (40) Bill came in. **He** looked happy.
  - Furthermore, even in the same context, the referent of *he* is not rigidly determined.

    (41) John told Bill that **he** won the election.

    Here, 'he' can be John or Bill.
- We will analyze pronouns as **variables**.[1]
- Indices and assignment functions
  - We assume that pronouns have **indices**, indicated by a subscript (e.g. he_5).
  - Indices are natural numbers (i.e. members of $\mathbb{N}$).
  - An indexed pronoun (e.g. he_5) is interpreted via an **assignment function**.
  - An assignment function g is a function that takes a natural number and returns an individual.
  - From now on, we will relativize the meaning to g in addition to s, so we always write $[\![\alpha]\!]^{s,g}$ (the meaning of α with respect to s and g).

    (42) $[\![\textbf{he}_5]\!]^{s,g} = g(5)$
    (for any situation s and for any assignment function g)

[1]This is not the only theory of pronouns. See Pauline Jacobson's papers (e.g. Jacobson 1999, 2012) for the so-called *variable-free semantics* where pronouns denote identity functions.

- Everything else stays the same.

(43) a. $[\![\textbf{John}]\!]^{s,g}$ = John
b. $[\![\textbf{smokes}]\!]^{s,g} = \lambda x_e$. 1 iff x smokes in s

(44) $[\![\textbf{John smokes}]\!]^{s,g}$ = 1 iff John smokes in s

(45) $[\![\textbf{He}_{16}\textbf{ smokes}]\!]^{s,g}$ = 1 iff $g(16)$ smokes in s

- This allows us to account for the flexibility of reference.
  - The same pronoun with the same index may have a different referent depending on the assignment function.

    (46) a. $[\![\textbf{he}_8]\!]^{s,g_1} = g_1(8)$
    b. $[\![\textbf{he}_8]\!]^{s,g_2} = g_2(8)$

    It's possible that $g_1(8) \neq g_2(8)$.
  - We assume that which assignment function to use is determined pragmatically, but we stay silent about how exactly this is done.
  - The referent of a pronoun also depends on the index.

    (47) a. $[\![\textbf{John told Bill that he}_7\textbf{ won the election}]\!]^{s,g}$
    b. $[\![\textbf{John told Bill that he}_{12}\textbf{ won the election}]\!]^{s,g}$

    These two sentences can denote different truth-values.
- We haven't incorporated the phi-features yet. That's for Day 2.

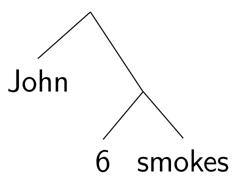
2.2 Bound Pronouns and Quantifiers

- We'll use the same semantics (pronouns as variables) to account for bound uses of pronouns.
- In order to achieve this, we introduce binding indices, Binding Rule and modified assignment function.
- We assume that DPs come with **binding indices** (indicated by superscripts).

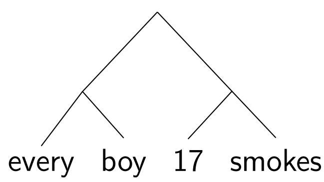
  (48) a. John^{6} smokes.
  b. [Every boy]^{17} smokes.

  Let's assume that they are separate nodes in the LF representations, e.g. (48) are shorthand for (49).

  (49) a.



b.



  (Note that binding indices are actually not part of the DPs themselves)
- Binding indices trigger the **Binding Rule**:

(50) *Binding Rule*

$$\left[\!\left[[\; i \;\; [_{\mathbf{A}} \ldots] \;] \right]\!\right]^{s,g} = \lambda x_e. \left[\left[\!\left[[_{\mathbf{A}} \ldots] \right]\!\right]^{s,g[i \to x]} (x) \right]$$

- The Binding Rule makes use of **assignment modification**: '$g[i \to x]$' denotes the assignment function obtained from g by changing the value of $g(i)$ to x. i.e.
  - $g[i \to x](i) = x$
  - for all $j \in \mathbb{N}$ different from i, $g[i \to x](j) = g(j)$
- The Binding Rule enables variable binding:
  - If **A** in (50) contains a pronoun indexed with i, it will refer to x.
  - When the subject is a quantifier, we'll get a bound pronoun interpretation.
- Example:
  - Some lexical entries (we ignore the DP-internal structure of 'his mother')

(51) For any situation s and for any assignment function g,

a. $[\![\textbf{likes}]\!]^{s,g} = \lambda y_e.[\lambda x_e.\ 1 \text{ iff } x \text{ likes } y \text{ in } s]$ (type $\langle e, \langle e, t \rangle\rangle$)

b. $[\![\textbf{his}_9 \textbf{ mother}]\!]^{s,g} = g(9)\text{'s mother}$ (type e)

  - We can combine these two and yield the following:

$$\left[\!\left[[_{\mathbf{VP}} \textbf{ likes his}_9 \textbf{ mother}] \right]\!\right]^{s,g} = [\![\textbf{likes}]\!]^{s,g}([\![\textbf{his}_9 \textbf{ mother}]\!]^{s,g})$$

$$= [\lambda y_e.\lambda x_e.\ 1 \text{ iff } x \text{ likes } y \text{ in } s](g(9)\text{'s mother})$$

$$= \lambda x_e.\ 1 \text{ iff } x \text{ likes } g(9)\text{'s mother in } s$$

  - What if we had the binding index 9 on top of this?

$$\left[\!\left[[\; 9 \;\; [_{\mathbf{VP}} \textbf{ likes his}_9 \textbf{ mother}] \;] \right]\!\right]^{s,g} = \lambda z_e. \left[\!\left[[_{\mathbf{VP}} \textbf{ likes his}_9 \textbf{ mother}] \right]\!\right]^{s,g[9 \to z]} (z)$$

$$= \lambda z_e.\ [[\lambda x_e.\ 1 \text{ iff } x \text{ likes } g[9 \to z](9)\text{'s mother in } s](z)]$$

$$= \lambda z_e.\ 1 \text{ iff } z \text{ likes } g[9 \to z](9)\text{'s mother}$$

$$= \lambda z_e.\ 1 \text{ iff } z \text{ likes } z\text{'s mother}$$

Notice that the denotation of the pronoun is dependent on the subject.

– We can combine '**John**' with this:

$$[\![\textbf{John}^9 \textbf{ likes his}_9 \textbf{ mother}]\!]^{s,g} = \left[\!\!\left[\; [\, 9 \; [_{\textbf{VP}} \textbf{ likes his}_9 \textbf{ mother}]\,] \; \right]\!\!\right]^{s,g} ([\![\textbf{John}]\!]^{s,g})$$

$$= [\lambda z_e.\ 1 \text{ iff } z \text{ likes } z\text{'s mother}](\text{John})$$

$$= 1 \text{ iff John likes John's mother}$$

- We'll analyze quantifiers as taking predicates (as generalized quantifiers):

(52) a. $[\![\textbf{every boy}]\!]^{s,g} = \lambda P_{\langle e,t\rangle}.\ 1 \text{ iff for every boy } x,\ P(x) = 1$
b. $[\![\textbf{no boy}]\!]^{s,g} = \lambda P_{\langle e,t\rangle}.\ 1 \text{ iff for no boy } x,\ P(x) = 1$

These are functions that take a predicative meaning (of type $\langle e, t\rangle$) and return a truth-value. So they are of type $\langle\langle e, t\rangle, t\rangle$.

- Example (bound vs. free pronouns)

– A quantifier binding a pronoun:

$$[\![\textbf{every boy}^9 \textbf{ likes his}_9 \textbf{ mother}]\!]^{s,g}$$

$$= [\![\textbf{every boy}]\!]^{s,g} \left(\left[\!\!\left[\; [\, 9 \; [_{\textbf{VP}} \textbf{ likes his}_9 \textbf{ mother}]\,] \; \right]\!\!\right]^{s,g} \right)$$

$$= [\![\textbf{every boy}]\!]^{s,g}([\lambda z_e.\ 1 \text{ iff } z \text{ likes } z\text{'s mother}])$$

$$= [\lambda P_{\langle e,t\rangle}.\ 1 \text{ iff for every boy } x,\ P(x) = 1]([\lambda z_e.\ 1 \text{ iff } z \text{ likes } z\text{'s mother}])$$

$$= 1 \text{ iff for every boy } x,\ [\lambda z_e.\ 1 \text{ iff } z \text{ likes } z\text{'s mother}](x) = 1$$

$$= 1 \text{ iff for every boy } x,\ x \text{ likes } x\text{'s mother}$$

– If the pronoun had a different index, we would get a referential/free interpretation.

$$\left[\!\!\left[\; [\, 9 \; [_{\textbf{VP}} \textbf{ likes his}_3 \textbf{ mother}]\,] \; \right]\!\!\right]^{s,g} = \lambda z_e. \left[\!\!\left[\; [_{\textbf{VP}} \textbf{ likes his}_3 \textbf{ mother}] \; \right]\!\!\right]^{s,g[9\to z]} (z)$$

$$= \lambda z_e.\ [[\lambda x_e.\ 1 \text{ iff } x \text{ likes } g[9 \to z](3)\text{'s mother in } s](z)]$$

$$= \lambda z_e.\ 1 \text{ iff } z \text{ likes } g[9 \to z](3)\text{'s mother}$$

$$= \lambda z_e.\ 1 \text{ iff } z \text{ likes } g(3)\text{'s mother}$$

– Consequently, the following sentence has a different truth-condition:

$$[\![\textbf{every boy}^9 \textbf{ likes his}_3 \textbf{ mother}]\!]^{s,g} = [\![\textbf{every boy}]\!]^{s,g}\left([\![\,[\;9\;[_{\textbf{VP}}\ \textbf{likes his}_3\textbf{ mother}]\,]\;]\!]^{s,g}\right)$$

$$= [\![\textbf{every boy}]\!]^{s,g}([\lambda z_e.\ 1 \text{ iff } z \text{ likes } g(3)\text{'s mother}])$$

$$= 1 \text{ iff for every boy } x,\ [\lambda z_e.\ 1 \text{ iff } z \text{ likes } g(3)\text{'s mother}](x) = 1$$

$$= 1 \text{ iff for every boy } x,\ x \text{ likes } g(3)\text{'s mother}$$

Notice that this is about a particular person's mother, namely $g(3)$'s mother.

SUMMARY

- Declarative sentences denote truth-values, 1 (TRUE) or 0 (TRUE), in a given situation s with respect to an assignment function g.

(53) $[\![\textbf{John smokes}]\!]^{s,g} = 1$ iff John smokes in s

- **Compositionality Principle:**
  The meaning of a complex phrase is determined by the syntax and semantics of its parts.
- Two components:

1. **Lexicon**
   For any situation s, for any assignment function g, and for any $i \in \mathbb{N}$,

(54) a. $[\![\textbf{John}]\!]^{s,g} = \text{John}$ (type e)
b. $[\![\textbf{smokes}]\!]^{s,g} = \lambda x_e.\ 1 \text{ iff } x \text{ smokes}$ (type $\langle e,t\rangle$)
c. $[\![\textbf{likes}]\!]^{s,g} = \lambda y_e.[\lambda x_e.\ 1 \text{ iff } x \text{ likes } y \text{ in } s]$ (type $\langle e,\langle e,t\rangle\rangle$)

(55) *Pronoun*
$[\![\textbf{he}_i]\!]^{s,g} = g(i)$ (type e)

(56) *Quantifiers*
a. $[\![\textbf{every boy}]\!]^{s,g} = \lambda P_{\langle e,t\rangle}.\ 1 \text{ iff for every boy } x,\ P(x) = 1$
(type $\langle\langle e,t\rangle,t\rangle$)
b. $[\![\textbf{no boy}]\!]^{s,g} = \lambda P_{\langle e,t\rangle}.\ 1 \text{ iff for no boy } x,\ P(x) = 1$ (type $\langle\langle e,t\rangle,t\rangle$)

2. **Compositional Rules**

(57) *Functional Application*

$$[\![\,[\ \textbf{A}\ \ \textbf{B}\ ]\,]\!]^{s,g} = [\![\textbf{A}]\!]^{s,g}([\![\textbf{B}]\!]^{s,g}) \text{ or } [\![\textbf{B}]\!]^{s,g}([\![\textbf{A}]\!]^{s,g}), \text{ whichever is appropriate.}$$

(provided that **A** is not a binding index)

(58) *Binding Rule*

$$[\![\,[\ i\ \ \textbf{A}\ ]\,]\!]^{s,g} = \lambda x_e.\ [[\![\textbf{A}]\!]^{s,g[i\rightarrow x]}(x)]$$

Further Background Readings

Introductory Textbooks on Formal Semantics

Semantics in Generative Grammar by Irene Heim and Angelika Kratzer. 1998. Blackwell.
One of the standard textbooks on formal semantics. The theory introduced above is built on Heim & Kratzer's.

Logic, Language and Meaning. by L. T. F. Gamut. 1991. University of Chicago Press.
Vol. 1 is a very good introduction to core concepts in formal semantics and logic. Vol. 2 covers a slightly more advanced topics in natural language semantics, and is also recommended.

Binding Theory by Daniel Büring. Cambridge Textbooks in Linguistics. Cambridge University Press.
Although the focus of this book is on the syntax and semantics of pronouns and Binding Theory, the first few chapters can be read as a general introduction to formal semantics. The rest of the book discusses very interesting topics in the syntax and semantics of pronouns. Recommendable to syntacticians too.

More Advanced Textbooks on the Mathematical Background

Mathematical Methods in Linguistics by Barbara H. Partee, Alice ter Meulen, and Robert E. Wall. 1990. Springer.
An accessible introductory textbook that covers core mathematical concepts essential in linguistics. Highly recommended for all formal linguists.

Lambda-Calculus and Combinators: An Introduction (2nd Ed.) by J. Roger Hindley and Jonathan P. Seldin. 2008. Cambridge University Press.
An accessible introductory textbook on lambda calculus, combinatorial logic, and computability.

References

Baker, Mark. 2008. *The Syntax of Agreement and Concord*. Cambridge: Cambridge University Press.
Jacobson, Pauline. 1999. Towards a variable free semantics. *Linguistics and Philosophy* 22(2). 117–185.
Jacobson, Pauline. 2012. The direct compositionality and "uninterpretability": The case of (sometimes) "uninterpretable" features on pronouns. *Journal of Semantics* 29(3). 305–343.
Pearson, Hazel. 2011. A new semantics for group nouns. In *Proceedings of WCCFL 28*, 160–168.
Percus, Orin. 2011. Gender features and interpretation: a case study. *Morphology* 21. 167–196.
Sag, Ivan & Thomas Wasow. 1999. *Syntactic Theory: A Formal Introduction*. Stanford: CSLI.
Schlenker, Philippe. 2006. Ontological symmetry in language: A brief manifesto. *Mind and Language* 21(4). 504–539.
Wechsler, Stephen & Larisa Zlatić. 2003. *The Many Faces of Agreement*. Stanford: CSLI.