Pronouns and Variable Assignments

1. Putting this Unit in Context

(1) **What We've Done So Far This Unit**

- Expanded our semantic theory so that it includes (the beginnings of) a theory of how the *presuppositions* of a sentence are derived.

- Expanded our theory so that it includes (the beginnings of) a theory of how the *implicatures* of a sentence are derived.

(2) **Implicatures and Context**

- We've seen that implicatures are crucially related to *context.*

  Implicatures are inferences drawn from (i) the asserted content of the sentence, and (ii) features of the context, particularly the assumption that the speaker is following the 'Gricean Conversational Maxims'

- Thus, context clearly has an effect on the informational content of a sentence, through its affect on the *implicatures* of the sentence.

But, it turns out, implicatures aren't the only thing affected by context…

(3) **Truth-Conditions and Context**

In this part of the unit, we will see that context also affects the *truth-conditions* (asserted content) of a sentence…

a. Key Example: *Pronouns*

(i) In a context where I am pointing at Barack Obama, "He smokes" is T *iff* Barack Obama smokes.

(ii) In a context where I am pointing at Joe Biden, "He smokes" is T *iff* Joe Biden smokes.

b. Conclusion: The T-conditions of "He smokes" varies from context to context.

c. Our Goal for This Unit:
Augment our extensional semantics so that it captures the way in which the T-conditions of certain sentences (*e.g.* those containing pronouns) depend upon features of the context.

2. Pronouns and the Basic Problem of Their Context Dependency

(4) **Overarching Question**

- How should / could our semantic system model the meanings of *pronouns*?
- What should our interpretation function "[[.]]" yield as the extension of a pronoun?

<u>Our Methodology (Again):</u>
Let's consider the T-conditions of sentences that contain pronouns *and lexical items whose extensions we already know*, and from those try to deduce a lexical entry for the pronoun that will derive those T-conditions…

(5) **First Core Fact**
The extension of a pronoun seems to be an entity.

- *In a context where the speaker is pointing at Barack:*

  [[He smokes]] = T *iff* Barack smokes.

- CONCLUSION: [[He]] = Barack (*in this context*)

(6) **Second Core Fact**
The extension of a pronoun *can vary across contexts*.

- *In a context where the speaker is pointing at Joe:*

  [[He doesn't drink]] = T *iff* Joe doesn't drink.

- CONCLUSION: [[He]] = Joe (*in this context*)

(7) **The Challenge**

The property in (6) poses a serious problem for our account (as it is currently structured).

- How do we write a lexical entry for [[he]] which reflects the fact that its value varies depending on the context?

- We can't just write [[he]] = Barack *or* Joe *or* Barney *or* John… for every x in D_e

- Not only does such an entry just look ugly, it doesn't do the crucial work of relating the extension of the pronoun to *the specific context it's used in*…

  … all such an entry says is that "he" can refer to those entities… it doesn't tell us *when* it can refer to those entities….

(8) **A Syntactic Side-Note**

Following decades of work in syntactic theory, we will assume that pronouns are DPs. Thus, their structure is as follows:

```
DP
 |
 D
 |
he.
```

---

3. Variable Assignments: The Basic Idea

(9) **The Challenge (Restated)**

How can we augment our semantic interpretation function "[[.]]" so that it somehow represents properties of the context?

(10) **The Solution (First Pass)**

- We will represent contextual information via special superscripts on "[[.]]"
- We will start with that contextual information which concerns the interpretation of pronouns.
- We will represent that information with a special superscript, which we will call the 'variable assignment'.
- For now, we take variable assignments to be *single specific entities* taken from D_e

(11) **Some 'Lingo'**

$[[\text{ XP }]]^g$ = 'the extension of XP **relative to the variable assignment *g***'
('the extension of XP **in a context where we're talking about *g***')

(12) **Pronoun Rule, First Pass [PR]**

If X is a pronoun (*he, she, it, him, her, his,* etc.), then $[[X]]^g = g$

<u>Illustration</u>

a. $[[\text{ he }]]^{\text{Barack}}$ = Barack
b. $[[\text{ he }]]^{\text{Joe}}$ = Joe

The introduction of variable assignments to our interpretation function "[[.]]" requires that we make a few minor adjustments to our semantic interpretation rules…

(13) **General Convention for Non-Pronominals**

If X is a terminal node and X is not a pronoun, then $[[X]]^g = [[X]]$

Illustration

$[[\text{ smokes }]]^{Barack} = [[\text{ smokes }]] = [\ \lambda x_e : \underline{\text{x smokes}}\ ]$

(14) **General Convention for Semantic Rules**

Our existing rules of semantic composition are not yet sensitive to variable assignments.

a. (New) Function Application

Let *g* be any variable assignment. If X is a branching node that has two daughters – Y and Z – and if $[[Y]]^g$ is a function whose domain contains $[[Z]]^g$, then $[[X]]^g = [[Y]]^g(\ [[Z]]^g\ )$

b. (New) Non-Terminal Nodes

Let *g* by any variable assignment. If X is a non-branching node, and Y is its sole daughter, then $[[X]]^g = [[Y]]^g$

c. (New) Terminal Nodes

Let *g* be any variable assignment. If X is a terminal node *and is not a pronoun*, then $[[X]]^g$ (= $[[X]]$) is specified in the lexicon

d. (New) Predicate Modification

Let *g* be any variable assignment. If X is a branching node that has two daughters – Y and Z – and if both $[[Y]]^g$ and $[[Z]]^g$ are in $D_{<et>}$, then $[[X]]^g$ =
$[\ \lambda x : x \in D_e\ .\ \underline{[[Y]]^g(x) = T \text{ and } [[Z]]^g(x) = T}\ ]$

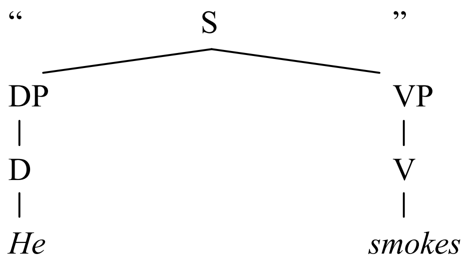
These notational additions allow our system to:

(i) calculate T-conditions for sentences containing pronouns (relative to a given context)

(ii) capture the fact that the T-conditions of such sentences can vary with the context.

(15) **Sample Derivation 1**

Context: We are talking about Barack

a. “ 

” is T *iff* (by notation)

b. $[[\text{ S }]]^{\text{Barack}} = \text{T}$

c. Subproof

(i) $[[\text{ VP }]]^{\text{Barack}}$ = (by NNx2, TN)

(ii) [λx_e : x smokes]

d. Subproof

(i) $[[\text{ DP }]]^{\text{Barack}}$ = (by NNx 2)

(ii) $[[\text{ he }]]^{\text{Barack}}$ = **(by PR)**

(ii) Barack

e. $[[\text{ S }]]^{\text{Barack}} = \text{T}$ *iff* (by FA, c, d)

f. $[[\text{VP}]]^{\text{Barack}}$ ($[[\text{DP}]]^{\text{Barack}}$) = T *iff* (by c)

g. [λx_e : x smokes] ($[[\text{DP}]]^{\text{Barack}}$) = T *iff* (by d)

h. [λx_e : x smokes] (Barack) = T *iff* (by LC)

i. Barack smokes

(16) **Sample Derivation 2**

Context: We are talking about Joe

a. " [S [DP [D *He*]] [VP [V *smokes*]]] " is T *iff* (by notation)

b. $[[\text{ S }]]^{\text{Joe}} = \text{T}$

c. Subproof

(i) $[[\text{ VP }]]^{\text{Joe}} =$ (by NNx2, TN)

(ii) [λx_e : x smokes]

d. Subproof

(i) $[[\text{ DP }]]^{\text{Joe}} =$ (by NNx 2)

(ii) $[[\text{ he }]]^{\text{Joe}} =$ **(by PR)**

(ii) Joe

e. $[[\text{ S }]]^{\text{Joe}} = \text{T}$ *iff* (by FA, c, d)

f. $[[\text{VP}]]^{\text{Joe}} ([[\text{DP}]]^{\text{Joe}}) = \text{T}$ *iff* (by c)

g. [λx_e : x smokes] ($[[\text{DP}]]^{\text{Joe}}$) = T *iff* (by d)

h. [λx_e : x smokes] (Joe) = T *iff* (by LC)

i. Joe smokes

(17) **The Upshot of All This**

Our semantic interpretation function "[[.]]" now comes paired with a superscript, *g*.

This superscript – which we call a 'variable assignment' – represents (certain features of) the context that a given phrase is being interpreted relative to.

4. Variable Assignments as Functions

(18) **Critical Problem for Our 'First Pass' Account**

It is possible to have multiple instances of a given pronoun *he* in a single sentence, where *each* instance of the pronoun refers to a *different* entity.

a. Illustration:

Context: *The speaker points first at Barack and then at Joe*

T-Conditions: "He likes him" is T *iff* Barack likes Joe

But, if a variable assignment is only a *single* entity, then all pronouns in a sentence will have to be interpreted as referring to that same single entity.

b. Illustration: (the reader is invited to confirm the following)

$[[$ He likes him $]]^{Barack}$ = T *iff* Barack likes Barack

$[[$ He likes him $]]^{Joe}$ = T *iff* Joe likes Joe.

There's actually a solution to this problem that employs an idea originally proposed by *syntacticians* for dealing with exactly this problem…

(19) **Solution: Indices**

There are two parts to this solution: one syntactic and one semantic.

a. Syntactic Component
Every instance of a pronoun in a sentence is assigned an *index*. We might represent these indices simply as numeric subscripts.

(i) he_1 likes him_2. (ii) he_{23} likes him_{456}. (iii) he_2 likes him_2.

b. Semantic Component (Vague; Taken from Syntax)

(i) Pronouns that receive the same index must refer to the same entity.

$[[He_1$ smokes and he_1 dances$]]^g$ = T *iff* Barack smokes and Barack dances.

(ii) Pronouns that receive different indices can refer to different entities.

$[[He_1$ smokes and he_2 dances$]]^g$ = T *iff* Barack smokes and Joe dances.

But, how do we formalize this vaguely stated 'semantics' for the notation of indices in (19b)?

(20) **Crucial Augmentation of Variable Assignments**

A variable assignment *g* is now no longer a *single* individual… rather it is a (partial) *function* from natural numbers (indices) to the set of individuals.

Illustration of Possible Variable Assignments

s = { < 1, Barack > }

h = { <1,Barack>, <2,Joe> }

j = { <5, Bill>, <78, Barack>, <2098, Joe> … }

Side-Note:

- Our earlier variable assignments were taken to represent (in a direct fashion) *the entity that we are talking about.*
- Our new, 'functional' variable assignments can still be thought of doing essentially the same thing: they are functions *into* the set of *entities that we are talking about*.
- You can (for now) think of the index associated with an entity *x* as something like 'the memory slot' that is assigned to *x* in the course of the conversation.

With our new variable assignments, we need a new rule for interpreting pronouns…

(21) **Pronouns Rule (PR) [Heim & Kratzer (1998: 111)]**

If X is a pronoun bearing index *n*, and *g* is a variable assignment, and *n* is an index in the domain of *g*, then $[[X_n]]^g = g(n)$.

Side-Note:
The conventions laid out earlier in (13) and (14) may remain exactly as before.

(22) **Illustration of the Rule PR**

a. $[[\text{ he}_1\ ]]^s$ = (by PR)
s(1) = (by definition of *s* in (20))
Barack

b. $[[\text{ he}_2\ ]]^h$ = (by PR)
h(2) = (by definition of *h* in (20))
Joe

c. $[[\text{ he}_5\ ]]^j$ = (by PR)
j(5) = (by definition of *j* in (20))
Bill

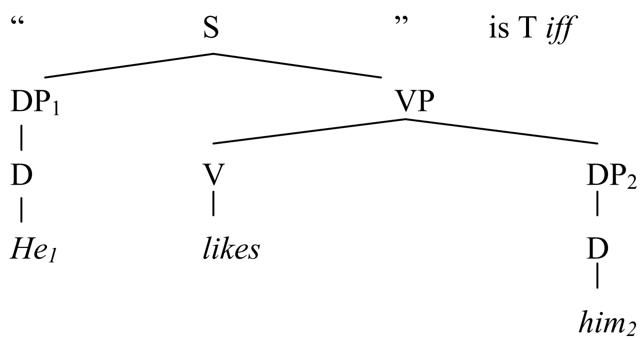
d. $[[\text{ he}_{23}\ ]]^h$ = (by PR)
h(23) = (by definition of *h* in (20))
UNDEFINED!!!

With these additions in place, we can now easily interpret sentences like (18a).

(23) **Sample Derivation**

Context: *Speaker first points at Barack and then at Joe*

Variable Assignment (Reflecting the Context): g = { < 1, Barack > , <2 , Joe > }

a. “  ” is T *iff* (by notation)

b. $[[\text{ S }]]^g = T$

c. Subproof

(i) $[[\text{ DP}_1\ ]]^g$ = (by NNx2)

(ii) $[[\text{ he}_1\ ]]^g$ = **(by PR)**

(iii) g(1) = **(by definition of g)**

(iv) Barack

d. Subproof

(i)	$[[\text{DP}_2]]^g$	=	(by NNx2)
(ii)	$[[\text{him}_2]]^g$	=	**(by PR)**
(iii)	g(2)	=	**(by definition of g)**
(iv)	Joe		

e. Subproof

(i)	$[[\text{V}]]^g$	=	(by NNx 2, TN)
(ii)	$[\lambda x_e : [\lambda y_e : \underline{\text{y likes x}}]]$		

f. Subproof

(i)	$[[\text{VP}]]^g$	=	(by FA, d, e)
(ii)	$[[\text{V}]]^g ([[\text{DP}_2]]^g)$	=	(by e, d)
(iii)	$[\lambda x_e : [\lambda y_e : \underline{\text{y likes x}}]] (\text{Joe})$	=	(by LC)
(iv)	$[\lambda y_e : \underline{\text{y likes Joe}}]$		

g. $[[\text{S}]]^g = \text{T}$ *iff* (by FA, c, f)

h. $[[\text{VP}]]^g ([[\text{DP}_1]]^g) = \text{T}$ *iff* (by c, f)

i $[\lambda y_e : \underline{\text{y likes Joe}}](\text{Barack}) = \text{T}$ *iff* (by LC)

j. Barack likes Joe

(24) **The Upshot**

- Pronouns are represented as bearing an ‘index’ (natural number)
- Variable assignments are functions from indices (numbers) to entities
- Thus, the variable assignment (still) implicitly represents features of the context:

  The entities we are referring to and *which pronouns we are using to refer to which entity.*

- Our rule of PR links the nature of the variable assignment (function from indices to entities) to the extension of the pronoun

The extension of the pronoun depends upon (i) the index it bears, and (ii) the variable assignment (i.e., ‘context’) it is interpreted with

5. The Role of Gender (and Other Features) on Pronominal Meaning

In the system we have constructed thus far, the gender of the pronoun plays no role in limiting its interpretation.

That is, nothing so far rules out a pronoun *he* bearing index *n* from being interpreted relative to a variable assignment that maps *n* to some *woman*.

(25) **Illustration: Vacuity of Gender**

a. Variable Assignment: g = { <1, Sally> … }

b. Computation

(i)	$[[\text{ he}_1]]^g$	=	(by PR)
(ii)	g(1)	=	(by definition of *g*)
(ii)	Sally		

Clearly, we need to fix this!

(26) **Facts to Capture**

a. Male Gender:
A pronoun *he* bearing index *n* can only be interpreted relative to variable assignments that map *n* to some *male entity*.

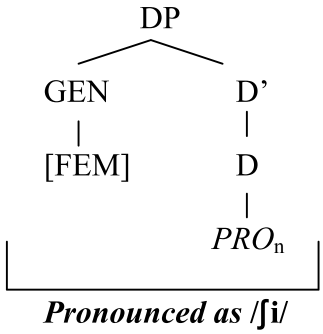
b. Female Gender:
A pronoun *she* bearing index *n* can only be interpreted relative to variable assignments that map *n* to some *female entity.*

(27) **Step 1: The Syntactic Representation of Gender**

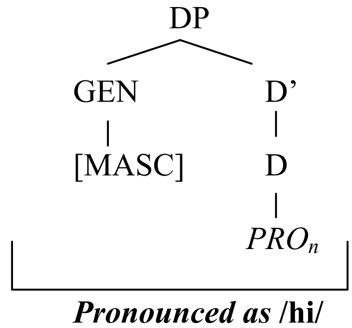
- We want to model the contribution that gender features make to the interpretation of the pronoun.
- Given the set-up of our semantic system, it would be easiest to model this by assuming that pronouns are complex expressions.
- That is, although we pronounce it as a single word, the pronoun *she* actually has the complex structure in (28a) below, whereby it consists of:
  - A gender feature [FEM] modifying
  - A pronominal 'core', which contributes the pronominal index itself

(28) **The Structure of Pronouns**

a.



b.



(29) **Step 2: The Semantic Contribution of Gender**

a. Question: What is the semantic contribution of Gender?

b. Observations:

(i) Adding [FEM] to [PRO$_n$] creates a pronoun that can only receive an interpretation if the variable assignment sends *n* to a **woman**.

(ii) Adding [MASC] to [PRO$_n$] creates a pronoun that can only receive an interpretation if the variable assignment sends *n* to a **man**.

c. Formal Proposal

Let's assume the following:

(i) [[FEM]] = [λx : x ∈ D_e **and x is female** . x]
The identity function on *female entities.*

(ii) [[MASC]] = [λx : x ∈ D_e **and x is male** . x]
The identity function on *male entities*.

(30) **The Key Consequence for *She*, Part 1**

Suppose that "she$_n$" is interpreted relative to a variable assignment *g* that maps *n* to some woman, Sally.

a. $[[\text{ DP }]]^g$ = (by FA)
b. $[[\text{FEM}]]^g([[\text{D'}]]^g)$ = (by NN)
c. $[[\text{FEM}]]^g([[\text{PRO}_n]]^g)$ = (by TN, PR)
d. [λx : x ∈ D_e **and x is female** . x](g(n)) = (by def. of *g*)
e. [λx : x ∈ D_e **and x is female** . x](Sally) = (by LC)
f. Sally

(31) **The Key Consequence for *She*, Part 2**

Suppose that "she_n" is interpreted relative to a variable assignment *g* that maps *n* to some man, Dave.

$[[\text{ DP }]]^g$ = ***UNDEFINED! FA <u>cannot</u> apply!***
$[[\text{ D' }]]^g$ is not in the *domain* of $[[\text{GEN}]]^g$

- $[[\text{GEN}]]^g = [[\text{FEM}]]^g = [\ \lambda x : x \in D_e$ **and x is female** $. x\ ]$
- $[[\text{D'}]]^g = [[\text{PRO}_n]]^g = g(n) = \text{Dave}$
- Dave is *not* in the domain of $[\ \lambda x : x \in D_e$ **and x is female** $. x\ ]$

(32) **General, Key Consequence**

Under the syntax in (28) and the semantics in (29c):

a. *She$_n$* can only receive an interpretation if the variable assignment *g* maps *n* to some female. (If *g* doesn't map *n* to a woman, then *she*$_n$ is just not interpretable).

b. Thus, *she*$_n$ can only be used to refer to females!

(A similar, parallel result clearly holds for the pronoun *he$_n$*).

This general approach can also (in principle) be extended to the other morpho-syntactic features of pronouns (i.e., person, number).

Seeing in detail how to do this, however, requires a bit more machinery than what we have thus far…

But, in general, it seems possible to think of pronominal (phi-)features as introducing 'presuppositions' in the way outlined above…