Quantificational DPs, Part 2:
Quantificational DPs in Non-Subject Position and Pronominal Binding [1]

1. Introduction

(1) **Our Current System**

a. The Ds *no, some*, and *every* are type <et <et, t>> (Quantificational Determiners)

(i) [[no]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: there is no x such that **g(x) = T** and f(x) = T]

(ii) [[a/some]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: there is an x s.t. **g(x) = T** and f(x) = T]

(iii) [[every]] = [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: for all x, if **g(x) = T**, then f(x) = T]

b. Quantificational DPs are of type <et,t> (Generalized Quantifiers)

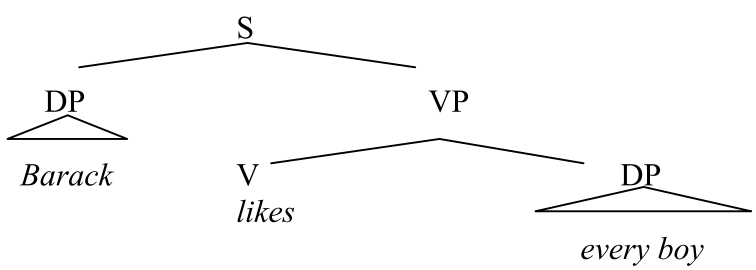
(i) [[no man]] = [$\lambda f_{<et>}$: there is no x such that **x is a man** and f(x) = T]

(ii) [[a/some man]] = [$\lambda f_{<et>}$: there is an x such that **x is a man** and f(x) = T]

(iii) [[every man]] = [$\lambda f_{<et>}$: for all x, if **x is a man**, then f(x) = T]

(2) **Key Empirical Problem**

Our system is not able to interpret sentences like the following:



The Problem:

- The V "likes" is of type <e<et>>, while the DP "every boy" is of type <et,t>
- Thus, our system is not able to assign an interpretation to the VP!

So, how can we fix this???

[1] These notes are based upon the material in Heim & Kratzer (1998: 178-189).

Let's begin by considering the T-conditions that sentence (2) seems to have.

(3) **Truth Conditions of Sentence (2)**

"Barack likes every boy" is T *iff* for all x, if x is a boy, then Barack likes x

(4) **Key Observation 1**

We could derive the T-conditions in (3) for sentence (2) if we could *somehow* combine the meaning of the DP object "every boy" with the function [λy_e : Barack likes y]

a. [[every boy]]([λy_e : Barack likes y]) = T *iff*

b. [$\lambda f_{<et>}$: for all x, if **x is a boy**, then f(x) = T] ([λy_e : Barack likes y]) = T *iff*

c. For all x, if x is a boy, then [λy_e : Barack likes y](x) = T *iff*

d. For all x, if x is a boy, then Barack likes x.

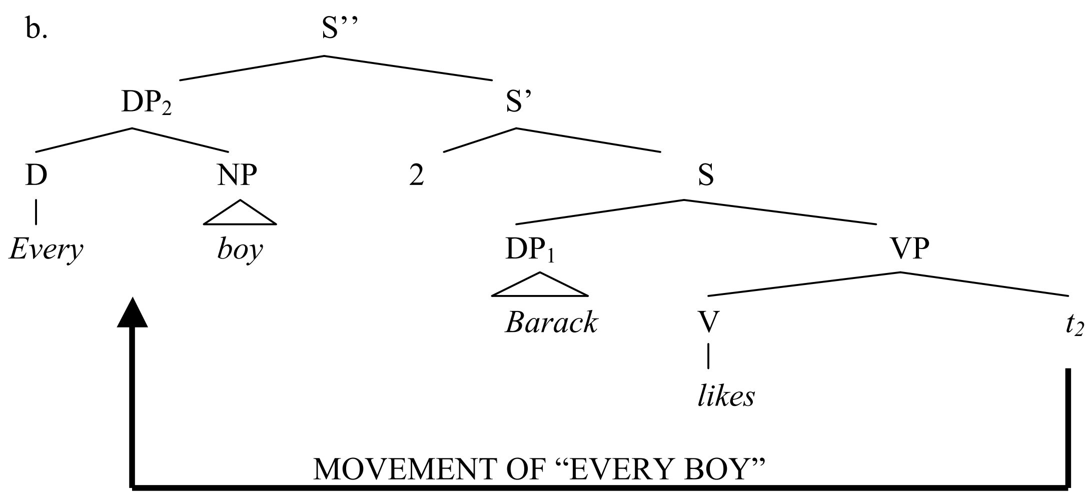
(5) **Key Observation 2**

- The sentence in (6) – which is nearly identical to sentence (2) – *is* interpretable by our semantics.
- **Moreover, (6) is predicted (correctly) to *also* have the T-conditions in (3).**

(6) **Movement of the Quantificational DP Object**

a. Every boy, Barack likes.

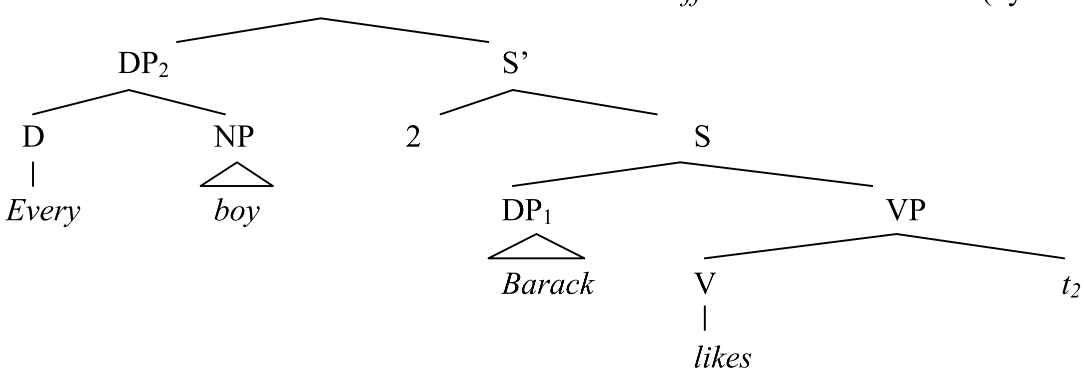
b.



(7) **T-Conditional Derivation for (6)**

Let g be any variable assignment.

a. “



” is T *iff* (by notation)

b. $[[\text{ S'' }]]^g = T$

c. **Subproof:**

(i) $[[\text{D}]]^g$ = (by NN, TN)
(ii) [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: <u>for all x, if g(x) = T, then f(x) = T</u>]

d. **Subproof:**

(i) $[[\text{NP}]]^g$ = (by NNx2, TN)
(ii) [λy_e : <u>y is a boy</u>]

e. **Subproof:**

(i) $[[\text{DP}_2]]^g$ = (by FA, c, d)

(ii) $[[\text{D}]]^g$ ($[[\text{NP}]]^g$) = (by c)

(iii) [$\lambda g_{<et>}$: [$\lambda f_{<et>}$: <u>for all x, if g(x) = T, then f(x) = T</u>] ($[[\text{NP}]]^g$) = (by LC)

(iv) [$\lambda f_{<et>}$: <u>for all x, if $[[\text{NP}]]^g$ (x) = T, then f(x) = T</u>] = (by d)

(v) [$\lambda f_{<et>}$: <u>for all x, if [λy_e : y is a boy] (x) = T, then f(x) = T</u>] = (by LC)

(vi) [$\lambda f_{<et>}$: <u>for all x, if x is a boy, then f(x) = T</u>]

f. **Subproof:**

(i) $[[\text{ S' }]]^g$ = (by PA)

(ii) [λy_e : <u>$[[\text{ S }]]^{g(2/y)}$ = T</u>]

g. **Subproof:**

(i) $[[\ DP_1\ ]]^{g(2/y)}$ = (by NNx3, TNx2, FA)
(ii) Barack

h. **Subproof:**

(i) $[[\ V\ ]]^{g(2/y)}$ = (by NN, TN)
(ii) [λx_e : [λz_e : z likes x]]

i. **Subproof:**

(i) $[[\ t_2\ ]]^{g(2/y)}$ = (by PR)

(ii) g(2/y) (2) = (by notation)

(iii) y

j. **Subproof:**

(i) $[[\ VP\ ]]^{g(2/y)}$ = (by FA, h, i)

(ii) $[[\ V\ ]]^{g(2/x)}$ ($[[\ t_2\ ]]^{g(2/y)}$) = (by h, i)

(iii) [λx_e : [λz_e : z likes x]] (y) = (by LC)

(iv) [λz_e : z likes y]

k. **Subproof:**

(i) $[[\ S\ ]]^{g(2/y)}$ = T *iff* (by FA, g, j)

(ii) $[[\ VP\ ]]^{g(2/y)}$ ($[[\ DP_1\ ]]^{g(2/y)}$) = T *iff* (by g, j)

(iii) [λz_e : z likes y] (Barack) = T *iff* (by LC)

(iv) Barack likes y

l. **Subproof:**

(i) $[[\ S'\ ]]^{g}$ = (by PA)

(ii) [λy_e : $[[\ S\ ]]^{g(2/y)}$ = T] = (by notation, k)

(iii) [λy_e : Barack likes y]

m. $[[S'']]^g = T$ *iff* (by FA, e, l)

n. $[[DP_2]]^g ([[S']]^g) = T$ *iff* (by e)

o. $[\lambda f_{<et>} : \underline{\text{for all x, if x is a boy, then f(x) = T}}] ([[S']]^g) = T$ *iff* (by LC)

p. For all x, if x is a boy, then $[[S']]^g (x) = T$ *iff* (by l)

q. For all x, if x is a boy, then $[\lambda y_e : \underline{\text{Barack likes y}}] (x) = T$ *iff* (by LC)

r. For all x, if x is a boy, then Barack likes x.

(8) **Key Observation, Reiterated**

- The sentence in (6) – which is nearly identical to sentence (2) – *is* interpretable by our semantics.
- **Moreover, (6) is predicted (correctly) to *also* have the T-conditions in (3).**

(9) **Burning Question**

- Could there be some sense in which the movement seen in (6) – which renders the sentence interpretable – could *also* be taking place in the sentence in (2)?
- Is there some sense in which the direct object DP in (2) *also* undergoes movement to the front of the sentence, *but we just don't pronounce such movement.*

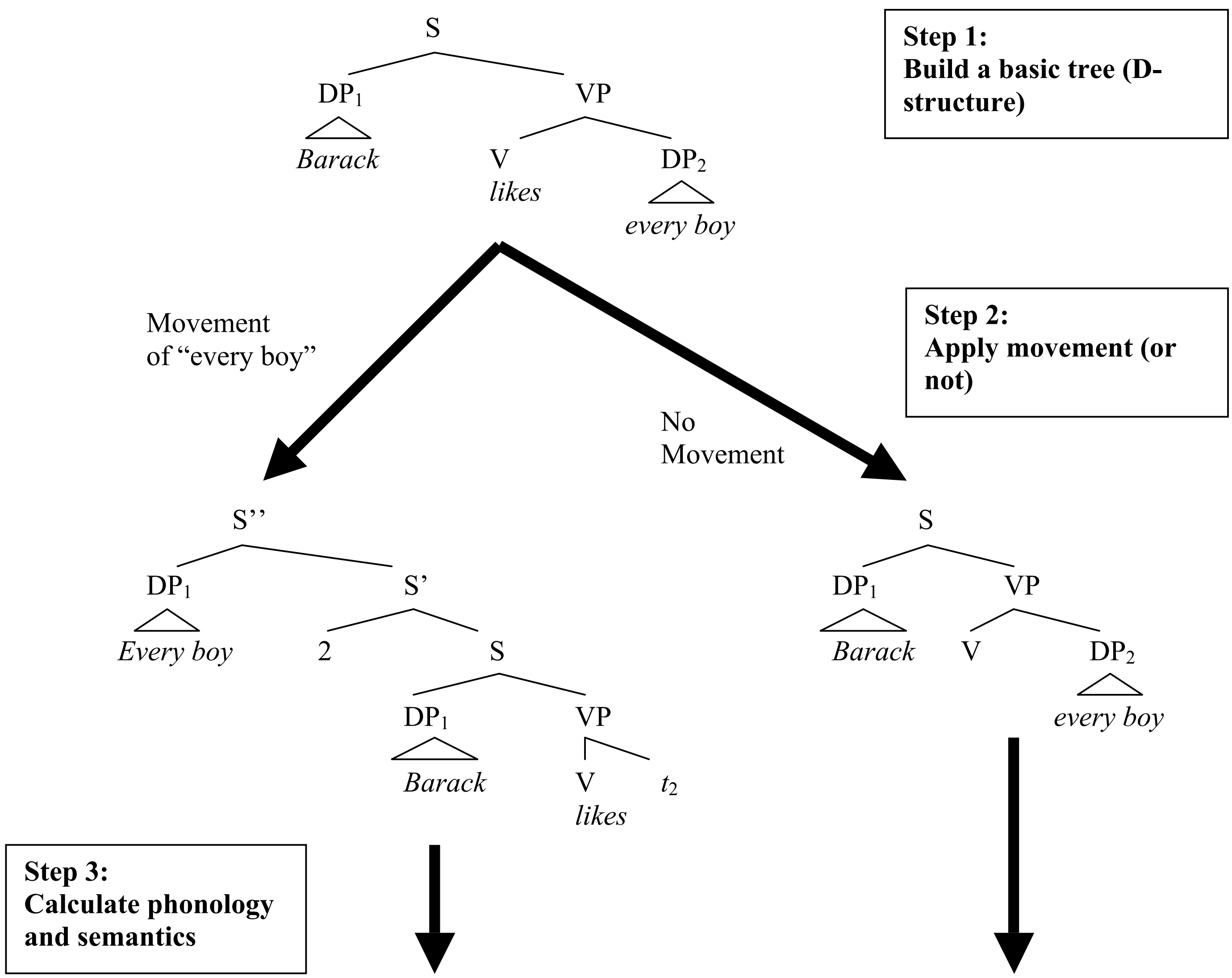
2. The Theory of Covert Movement

The following is a rather intuitive model of how a syntactic derivation proceeds (in a grammar where you have movement rules). In fact, it's kind of what we've been implicitly assuming…

(10) **A Very Intuitive Derivational Syntactic Architecture**

- You create a 'basic tree'
- You apply movement (optional)
- Then, from the post-movement structure, you derive (compute) the phonological form of the structure, and its semantics…

(11) **The Very Intuitive Derivational Architecture, in a Picture**



PHONOLOGY:
"Every boy, Barack likes:

SEMANTICS:
T *iff* for all x, if x is a boy, Barack likes x

PHONOLOGY:
"Barack likes every boy"

SEMANTICS:
undefined

(12) **Key Property of This Model**

The *pronounced form* (surface structure) of a sentence S is the same as the structure that the semantics interprets in order to compute a meaning for S.

(13) **Interesting Fact**

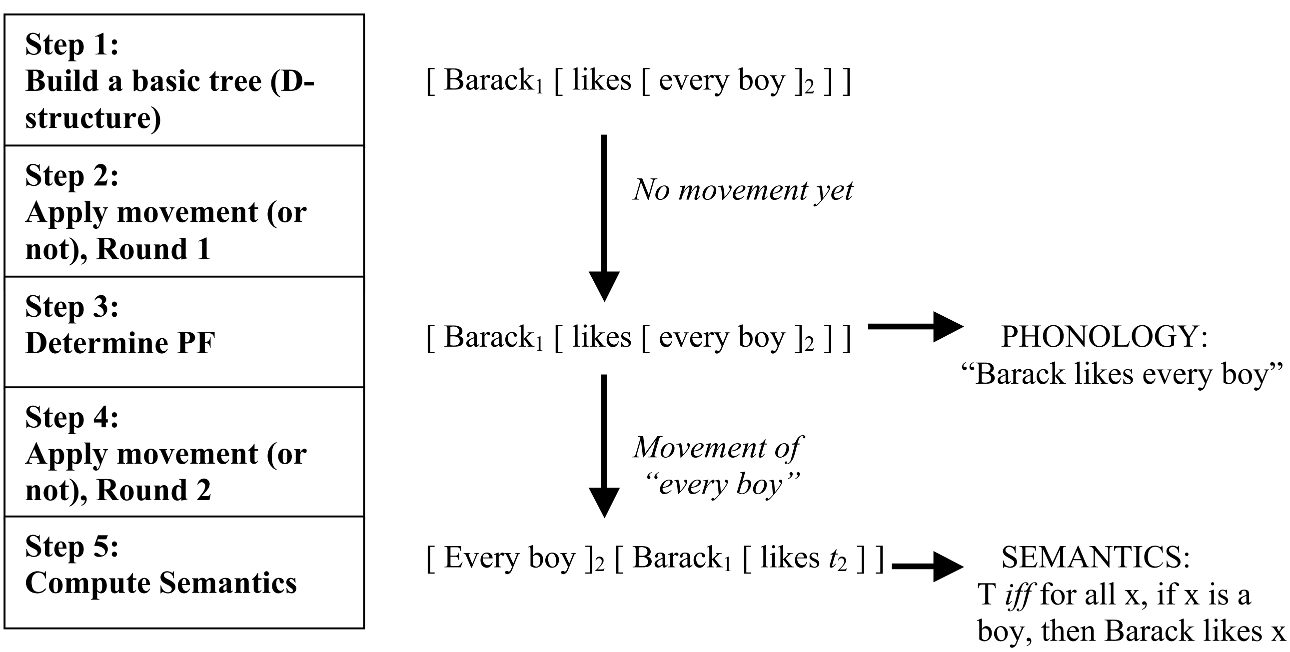
- Over time, linguists have learned that a model like (10)/(11) is not adequate for natural language (if you assume movement rules)

- **Rather, *if you assume that movement rules exist*, then it seems that language allows such rules to apply *after* the phonological form of the sentence has been computed…**

(14) **A More Empirically Adequate Derivational Syntactic Architecture**

- You create a 'basic tree' — D-Structure
- You apply movement (optional) — Movement, Round 1
- You determine the phonological form (PF) of the sentence — Determine PF
- **You apply more movement rules (optional) — Movement, Round 2**
- **You determine the semantics of the structure — Compute semantics**

(15) **The More Empirically Adequate Derivational Architecture, In a Picture**



(16) **Terminology: 'Overt / Covert Movement'**

- Movement which applies to a sentence S *before* the phonological form of S has been computed is *overt movement*.

- Movement which applies to a sentence S *after* the phonological form of S has been computed is *covert movement*.

(17) **Terminology: Surface Structure (SS)**

The syntactic representation of a sentence S that is used to compute the phonological form of the sentence.

(18) **Terminology: Logical Form (LF)**

The syntactic representation of sentence S that is used to compute the semantics of the sentence.

(19) **Key Property of Our New Architecture**

- The pronounced form (surface structure) of a sentence S does not necessarily reflect the form that S has when its interpretation is computed!

- So, while the surface form of S might show DP in position X, the semantics of S might be computed from a *different* structure…
  … one *derived* from the surface form of S via movement of DP!

(20) **Illustration (see (15))**

a. Sentence: "Barack likes every boy"

b. Surface Structure (SS): [Barack_1 [likes [every boy $]_2$]]

c. Possible Logical Form (LF): [every boy $]_2$ [2 [Barack_1 likes t_2]]

(21) **Illustration**

a. Sentence: "Joe saw no girl"

b. Surface Structure (SS): [Joe_1 [saw [no girl $]_2$]]

c. Possible Logical Form (LF): [no girl $]_2$ [2 [Joe_1 saw t_2]]

(22) **A Very Natural Question (Which Always Comes Up)**

So, is this grammatical model somehow proposing that I first say some sentence and then only later out figure out what I meant by it?

(23) **The Answer, in Brief**
No! The model here is competence model. The steps in the syntactic derivations are not intended to correlate with steps in the production task…

(24) **The Answer, in Full**

- This formal system is a model of *competence*, of our *knowledge* of which phonological strings PHON in English are paired with which meanings SEM.
  The system offers a *characterization* or *definition* of the licit pairs <PHON,SEM>

- In this system, a particular phonological form PHON is paired with a particular meaning SEM *iff* the system allows a derivation where

  (i) The phonological form computed *in the course* of the derivation is PHON
  (ii) The semantic value computed *in the course* of the derivation is SEM

- In actually speaking/comprehending English, we determine whether a particular PHON is paired with a particular SEM by asking the following question:

  Does my grammar allow a derivation where:
  (i) The phonological form computed *in the course* of the derivation is PHON
  (ii) The semantic value computed *in the course* of the derivation is SEM

- That is, in this model of grammar, the production task for speakers is as follows:

  'Given some particular SEM* I wish to express, find a PHON such that my grammar allows the pairing <PHON, SEM*>… '

  'That is, find a PHON such that my grammar allows a derivation where:
  (i) The phonological form computed *in the course* of the derivation is PHON
  (ii) The semantic value computed *in the course* of the derivation is SEM*

- Similarly, in this model, the production task for hearers is as follows:

  'Given some particular PHON* I've heard, find a SEM such that my grammar allows the pairing <PHON*,SEM>…'

  'That is, find a SEM such that my grammar allows a derivation where:
  (i) The phon. form computed *in the course* of the derivation is PHON*
  (ii) The semantic value computed *in the course* of the derivation is SEM

(25) **The Big Upshot**

- If we assume the syntactic architecture in (14)/(15), then it is now possible for movement to apply *without there being an overt phonological effect*.
  (we can have 'invisible' movement of DPs)

- Thus, a sentence with the phonological appearance of (20a) or (21a) might have as its (relevant) structural analysis the ones in (20c) or (21c).

- **Thus, our system is able to interpret English sentences where quantificational DPs are in object position by assuming that…**

  ... our semantics actually interprets a derived structure where the DP is no longer in object position!

(26) **The Game Being Played Here**

a. If the pronounced form of a sentence S is not interpretable by our semantic rules…

…we now allow ourselves some (limited) freedom to alter the structure of S so that we get a structure that our rules *can* interpret.

b. But, *we don't have unlimited freedom to alter S*…

… our alterations have to be ones that can be obtained by the rule of 'movement' (given the ways that 'movement' is independently known to be constrained)

The point in (26b) is one that we will return to, since it will (allegedly) provide some evidence for this treatment of quantificational DPs…

---

3. Interpreting Quantificational DPs Without Covert Movement

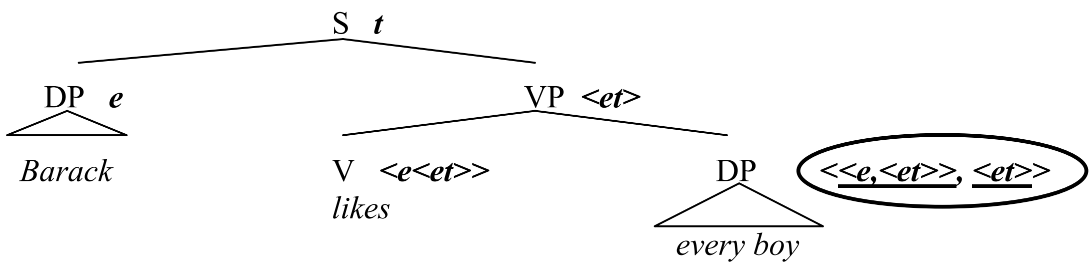
In the preceding section, we developed an analysis of (2) whose key idea is that our syntax permits the existence of 'movement that we don't hear', or 'invisible movement'.

Just as with 'invisible structure' more generally, the existence of invisible movement can complicate the task of syntactic parsing (see (24))…

… thus, it's worth considering analyses that do not appeal to 'covert movement'…

(27) **General Approach to Consider**

- The structure in (2) is not interpretable if the DP object is of type <et,t>.
- So, maybe the DP here is not of type <et,t>…
- ***Maybe the DP is of type <eet,et>!***



(28) **The Idea More Concretely**

- Perhaps *all* quantificational DPs are systematically ambiguous:
  - They have one reading in which they are expressions of type <et,t>
  - They have another reading in which they are expressions of type <eet, et>
- What disambiguates the reading of the DP is what syntactic position it is in
  - For reasons of semantics, only type <et,t> DPs could ever occupy subject position. (A type <eet,et> DP here would create an uninterpretable structure)
  - For reasons of semantics, only type <eet, et> DPs could ever occupy object position. (A type <et,t> DP here would create an uninterpretable structure)

OK, this idea seems to make sense in outline…
… but what is the meaning of the DP 'every boy' in a sentence like (2)?

(29) **Deducing the Meaning of "Every Boy" from the T-Conditions**

Targeted T-Conditions
"Barack likes every boy" is T *iff* for all x, if x is a boy, then Barack likes x

Deduction:

a. [[**likes** every boy]] = [λy_e : for all x, if x is a boy, then y **likes** x]

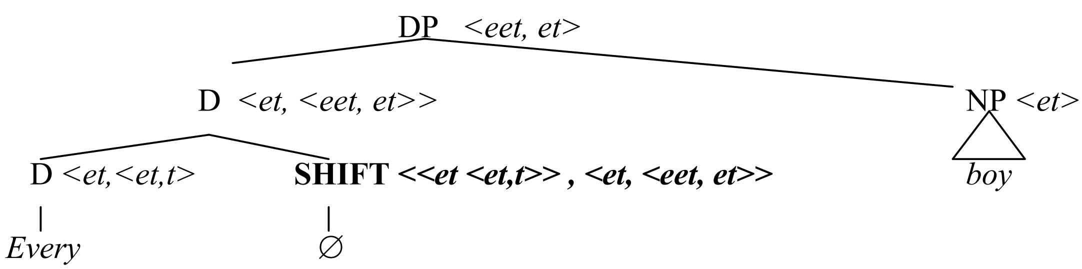
b. [[every boy]]([[**likes**]]) = [λy_e : for all x, if x is a boy, then [[**likes**]](x)(y) = T]

c. [[every boy]] = [$\lambda f_{<eet>}$ [λy_e : for all x, if x is a boy, then f(x)(y) = T]]

But how does "every boy" come to have this alternate extension?...

(30) **Key Hypothesis: Type Shifting of the Determiner**

Perhaps there exists one of those phonologically null 'type shifting' operators in the structure of "every boy" in (27)/(29)! (*See Assignment 4*)



Ok… but what is the extension of this type-shifting operator?

This one's actually pretty hard to deduce form first principles, so let me just give you the answer:

(31) **The Semantics of the Type-Shifting Operator**

$[[\text{ SHIFT }]] \;=\; [\, \lambda d_{\langle et, \langle et,t\rangle\rangle} : [\, \lambda g_{\langle et\rangle} [\, \lambda f_{\langle eet\rangle} [\, \lambda y_e : \mathbf{\underline{d}(\underline{g})(\; \underline{[\, \lambda z_e : f(z)(y) = T]}\;)} \,]\,]\,]\,]$

(32) **Illustration**

a. $[[\ [_{DP}\, [_{D}\text{ every SHIFT }]\ [_{NP}\text{ boy }]\ ]] =$ (by NN x4, FA x3)

b. $[[\text{SHIFT}]]([[\text{every}]])([[\text{boy}]]) =$ (by TN)

c. $[\lambda d_{\langle et, \langle et,t\rangle\rangle} : [\lambda g_{\langle et\rangle} [\lambda f_{\langle eet\rangle}$
$[\lambda y_e : \underline{d}(g)(\underline{[\lambda z_e : f(z)(y) = T]})]]]]([[\text{every}]])([[\text{boy}]]) =$ (by LC x2)

d. $[\lambda f_{\langle eet\rangle} [\lambda y_e : \underline{[[\text{every}]]([[\text{boy}]])}(\underline{[\lambda z_e : f(z)(y) = T]})]] =$ (by TN x2, LC)

e. $[\lambda f_{\langle eet\rangle} [\lambda y_e : \underline{[\lambda g_{\langle et\rangle} : \text{for all x, if x is a boy, then } g(x) = T]([\lambda z_e : f(z)(y) = T])}]]$
$=$ (by LC)

f. $[\lambda f_{\langle eet\rangle} [\lambda y_e : \underline{\text{for all x, if x is a boy, then } [\lambda z_e : f(z)(y) = T](x) = T}\ ]]$
$=$ (by LC)

g. $\mathbf{[\lambda f_{\langle eet\rangle} [\lambda y_e : \underline{\text{for all x, if x is a boy, then } f(x)(y) = T}\ ]]}$

(33) **Summary**

Our system could interpret structures like (2) if we assume that English contains a phonologically null, 'type-shifting' operator for determiners (31), which serves to 'shift' the type of the determiner to <et, <eet, et>>.

(34) **Exercise for the Reader**

Use the operator in (31) to derive a type <et <eet,et>> meaning for the other determiners we've seen (*e.g.* 'no', 'some', 'most', 'three', *etc.*).

In the next part of this unit (Quantificational DPs, Part 3), we will compare the 'covert movement' account of Section 2 to the 'type-shifting account' of Section 3.

---

4. Pronominal Binding

(35) **A Curious Feature of Our Semantics for Movement**

Our rules for interpreting movement structures, PA (36) and PR (37) – make crucial use of the variable assignment *g*, the same device used to interpret pronouns.

- It is through the variable assignment *g* that the trace created by movement is interpreted as the variable bound by the lambda operator

(36) **The Rule of Predicate Abstraction (PA) [Heim & Kratzer (1998: 186)]**
If X is a branching node whose daughters are Y and Z, and if Y is the index *n*, then for any **variable assignment g**, $[[\ X\ ]]^{g} = [\ \lambda x_e : \underline{[[\ Z\ ]]^{g(n/x)} = T}\ ]$

(37) **The Pronouns-and-Traces Rule (PR) [Heim & Kratzer (1998: 111)]**
If X is a pronoun *or a trace*, and *g* is a **variable assignment**, and n is an index in the domain of *g*, then $[[X_n]]^{g} = \mathbf{g}(n)$.

It is no accident that the 'contextual parameter' we use to interpret pronouns is the same one that is 'manipulated' in the interpretation of movement structures…

This very assumption allows our system to generate 'bound readings' of pronouns…

(38) **Bound Readings of Pronouns**

Certain structures containing pronouns allow interpretations where the pronoun appears to function as *a bound variable.*

(39) **Illustration of Bound Readings of Pronouns**

a. *Sentence:* Every boy likes the girl who likes him.

b. *Reading 1 (The 'Referential Reading')*
For all x, if x is a boy, then x likes the girl who likes (*e.g.*) Bill.

Verifying Scenario: Tom likes the girl who likes Bill (*i.e.,* Sue)
John likes the girl who likes Bill (*i.e.,* Sue)
Bill likes the girl who likes Bill (*i.e.,* Sue)

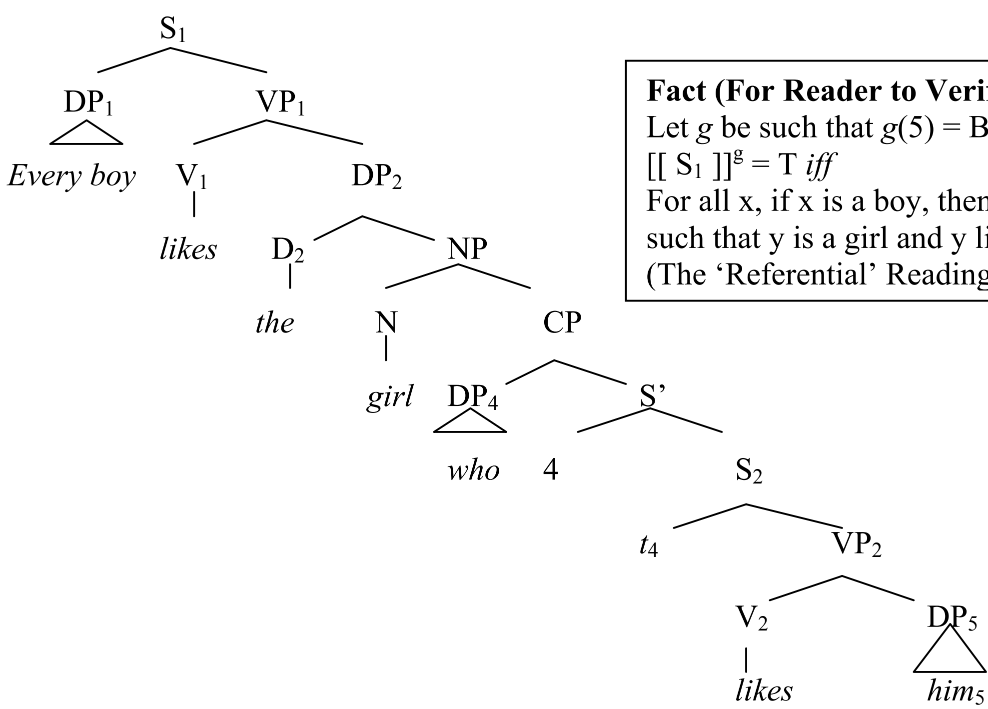
c. *Reading 2 (The 'Bound Reading')*
For all **x**, if **x** is a boy, then **x** likes the girl who likes **x**

Verifying Scenario: Tom likes the girl who likes Tom (*i.e.,* Fran)
John likes the girl who likes John (*i.e.,* Jenny)
Bill likes the girl who likes Bill (*i.e.,* Sue)

*As we will now see, our system is able to predict the existence of **both** these readings.*
For simplicity, we will temporarily ignore the presence of gender on the pronoun…

(40) **An LF for (39a) That Will Be Assigned the Referential Reading**

Key Properties: The quantificational DP undergoes no (covert) movement.
The pronoun *him* bears a different index from the quant. DP

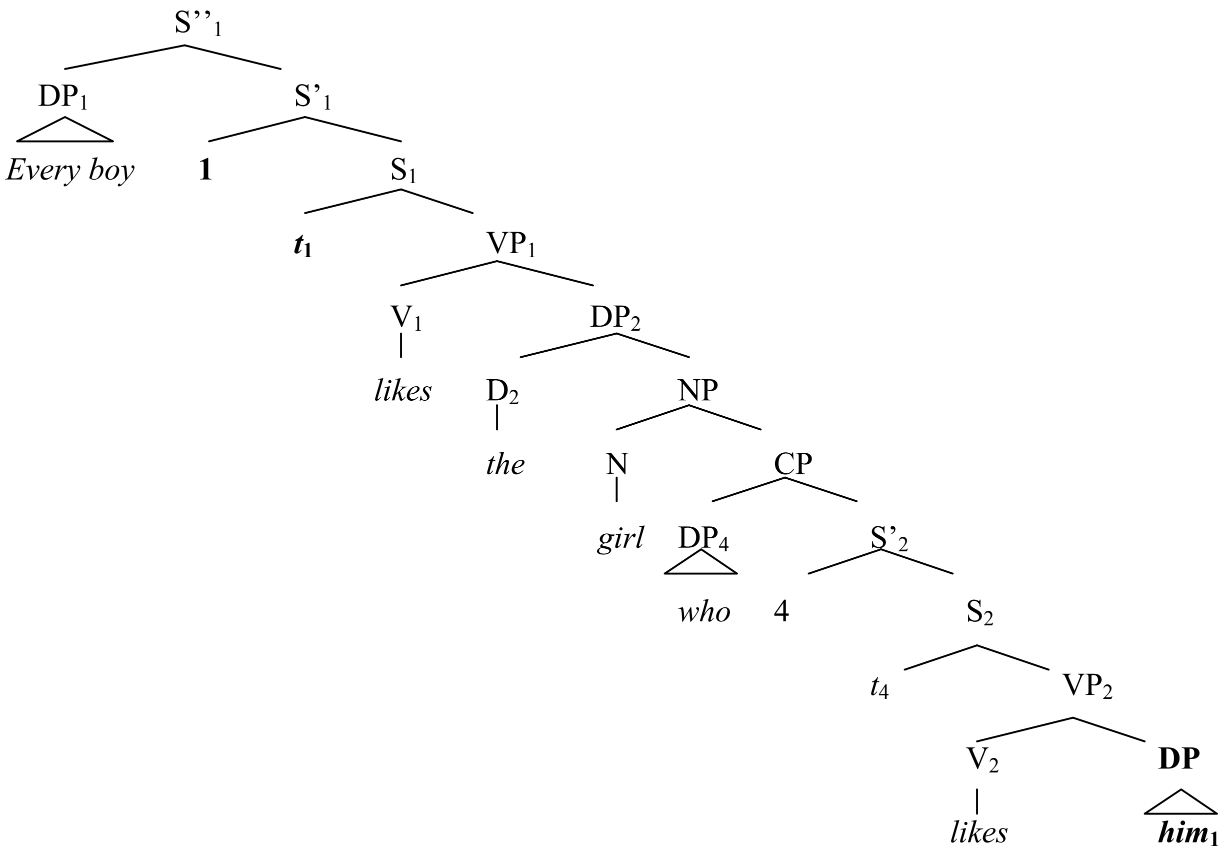


Fact (For Reader to Verify):
Let g be such that $g(5)$ = Bill. Then…
$[[S_1]]^g = T$ *iff*
For all x, if x is a boy, then x likes the unique y such that y is a girl and y likes **Bill**.
(The 'Referential' Reading (39b))

Of course, our ability to generate the ‘referential reading’ (39b) is obvious…
What’s truly important is the fact detailed below…

(41) **An LF for (39a) That Will Be Assigned the Bound Reading**

Key Properties: The quantificational DP undergoes (covert) movement.
The pronoun *him* bears the **same** index as the quantificational DP



The reader is strongly encouraged to closely follow the truth-conditional derivation below.

(42) **Derivation of the Truth-Conditions for LF (41)**

Let g be any variable assignment.

a. **Subproof**
(i) $[[\text{ DP}_1]]^g$ = (by FA, NNx3, TNx2, LC)
(ii) $[\lambda g_{<et>} : \underline{\text{for all z, if z is a boy, then g(z) = T}}]$

b. **Subproof**
(i) $[[\text{ S'}_1]]^g$ = (by PA)
(ii) $[\lambda x_e : \underline{[[\text{ S}_1]]^{g(1/x)} = T}]$

c. **Subproof**
(i) $[[\ t_1\ ]]^{g(1/x)}$ = (by PR, notation)
(ii) x

d. **Subproof**
(i) $[[\ V_1\ ]]^{g(1/x)}$ = (by NN, TN)
(ii) [λz_e : [λy_e : y likes z]]

e. **Subproof**
(i) $[[\ D_2\ ]]^{g(1/x)}$ = (by NN, TN)
(ii) [λf : f ∈ $D_{<et>}$ and there is exactly one s such that f(s) = T .
the unique s such that f(s) = T]

f. **Subproof**
(i) $[[\ N\ ]]^{g(1/x)}$ = (by NN, TN)
(ii) [λu_e : u is a girl]

g. **Subproof**
(i) $[[\ DP_4\ ]]^{g(1/x)}$ = (by NNx2, TN)
(ii) [$\lambda f_{<et>}$: f]

h. **Subproof**
(i) $[[\ S'_2\ ]]^{g(1/x)}$ = (by PA)
(ii) [λv_e : $[[\ S_2\ ]]^{g(1/x)(4/v)} = T$]

i. **Subproof**
(i) $[[\ t_4\ ]]^{g(1/x)(4/v)}$ = (by PR, notation)
(ii) v

j **Subproof**
(i) $[[\ V_2\ ]]^{g(1/x)(4/v)}$ = (by NN, TN)
(ii) [λw_e : [λr_e : r likes w]]

k. **Subproof**
(i) $[[\ DP\ ]]^{g(1/x)(4/v)}$ = (by NNx2)
(ii) $[[\ him_1\ ]]^{g(1/x)(4/v)}$ = (by PR)
(iii) g(1/x)(4/v)(1) = (by notation)
(iv) x

Key part of the proof!

l. **Subproof**
(i) $[[\ VP_2\ ]]^{g(1/x)(4/v)}$ = (by FA, j, k)
(ii) [λw_e : [λr_e : r likes w]](x) = (by LC)
(iii) [λr_e : r likes x]

m. **Subproof**

(i) $[[\ S_2\ ]]^{g(1/x)(4/v)} = T$ *iff* (by FA, i, l)

(ii) $[[VP_2]]^{g(1/x)(4/v)}([[\ t_4\ ]]^{g(1/x)(4/v)}) = T$ *iff* (by i, l)

(iii) $[\ \lambda r_e : \underline{r \text{ likes } x}\ ](v) = T$ *iff* (by LC)

(iv) v likes x

n. **Subproof**

(i) $[[\ S'_2\ ]]^{g(1/x)}$ = (by PA)

(ii) $[\ \lambda v_e : \underline{[[\ S_2\ ]]^{g(1/x)(4/v)} = T}\ ]$ = (by m, notation)

(iii) $[\ \lambda v_e : \underline{v \text{ likes } x}\ ]$

o. **Subproof**

(i) $[[\ CP\ ]]^{g(1/x)}$ = (by FA, g, n)

(ii) $[[\ DP_4\ ]]^{g(1/x)}(\ [[\ S'_2\ ]]^{g(1/x)}\ )$ = (by g, n)

(iii) $[\ \lambda f_{<et>} : f\ ](\ [\ \lambda v_e : \underline{v \text{ likes } x}\ ]\ )$ = (by LC)

(iv) $[\ \lambda v_e : \underline{v \text{ likes } x}\ ]$

p. **Subproof**

(i) $[[\ NP\ ]]^{g(1/x)}$ = (by PM, f, o)

(ii) $[\ \lambda o_e : \underline{[[N]]^{g(1/x)}(o) = T \text{ and } [[\ CP\ ]]^{g(1/x)}(o) = T}\ ]$ = (by f, o)

(iii) $[\ \lambda o_e : \underline{[\ \lambda u_e : u \text{ is a girl}\ ](o) = T \text{ and}}$
$\underline{[\ \lambda v_e : v \text{ likes } x\ ]\ (o) = T}\ ]$ = (by LC x2)

(iv) $[\ \lambda o_e : \underline{o \text{ is a girl and } o \text{ likes } x}\ ]$

q. **Subproof**

(i) $[[\ DP_2\ ]]^{g(1/x)}$ = (by FA, e, p)

(ii) $[[\ D_2\ ]]^{g(1/x)}([[\ NP\ ]]^{g(1/x)})$ = (by e)

(iii) $[\ \lambda f : f \in D_{<et>}$ and there is exactly one s such that $f(s) = T$.
the unique s such that $f(s) = T\ ]\ ([[\ NP\ ]]^{g(1/x)})$ = (by LC)

(iv) the unique s such that $[[\ NP\ ]]^{g(1/x)}(s) = T$ = (by p)

(v) the unique s such
that $[\ \lambda o_e : \underline{o \text{ is a girl and } o \text{ likes } x}\ ](s) = T$ = (by notation)

(vi) the unique s such that s is a girl and s likes x

r. **Subproof**

(i) $[[\ VP_1\ ]]^{g(1/x)}$ = (by FA, d, q)

(ii) $[[\ V_1\ ]]^{g(1/x)}([[\ DP_2\ ]]^{g(1/x)})$ = (by d)

(iii) $[\ \lambda z_e : [\ \lambda y_e : \underline{y \text{ likes } z}\ ]\ ]([[\ DP_2\ ]]^{g(1/x)})$ = (by LC)

(iv) $[\ \lambda y_e : \underline{y \text{ likes } [[\ DP_2\ ]]^{g(1/x)}}\ ]$ = (by q)

(v) $[\ \lambda y_e : \underline{y \text{ likes the unique s such that s is a girl and s likes } x}\ ]$

s. **Subproof**
(i) $[[\ S_1\ ]]^{g(1/x)} = T$ *iff* (by FA, c, r)
(ii) $[[\ VP_1\ ]]^{g(1/x)}([[\ t_1\ ]]^{g(1/x)}) = T$ *iff* (by c, r)
(iii) [λy_e : y likes the unique s such that s is a girl and s likes x](x) = T
iff (by LC)
(iv) x likes the unique s such that s is a girl and s likes x

t. **Subproof**
(i) $[[\ S'_1\ ]]^{g}$ = (by PA)
(ii) [λx_e : $[[\ S_1\ ]]^{g(1/x)} = T$] = (by notation, s)
(iii) [λx_e : x likes the unique s such that s is a girl and s likes x]

u. $[[\ S''_1\ ]]^{g} = T$ *iff* (by FA, a, t)

v. $[[\ DP_1\ ]]^{g}([[\ S'_1\ ]]^{g}) = T$ *iff* (by a, t)

w. [$\lambda g_{<et>}$: for all z, if z is a boy, then g(z) = T]
([λx_e : x likes the unique s such that s is a girl and s likes x]) = T *iff*
(by LC)

x. for all z, if z is a boy, then
[λx_e : x likes the unique s such that s is a girl and s likes x](z) = T *iff*
(by LC)

y. **for all z, if z is a boy, z likes the unique s such that s is a girl and s likes z**

(43) **What Just Happened**

Because pronouns and traces are interpreted via the same 'parameter' – the variable assignment *g* – it follows that…

- If a pronoun bears the same index *i* as some trace t_i …
- … *and* if that pronoun is c-commanded by the copied index *i* …
- … *then* that pronoun will (like the trace) be interpreted as the variable bound by the lambda operator introduced by *i*…

In this way, our system is able to generate 'bound readings' of pronouns!

(44) **Special Note**

- As stated above, in order for a pronoun X to receive a bound reading, it must be c-commanded by a 'copied index' identical to the index born by X

- Thus, for better or worse, in our system a quantificational DP can only bind a pronoun X if that DP undergoes (covert) movement.

- The reader can confirm this by computing the T-conditions for an LF that looks just like (40) except that the pronoun *him* bears the index '1' shared with the quantificational DP.

    - The reader will find that this LF *still* receives the 'referential reading', not the 'bound reading'.

- This dependency of binding upon movement is not commonly assumed in syntactic theory, but it is (perhaps wrongly) a core feature of our system.

---

5. Pronominal Binding and Gender

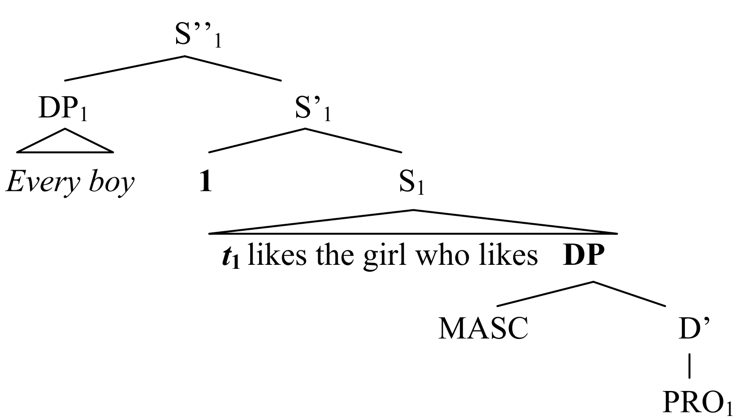
In the previous section, our LF in (41) and our computation in (42) leaves out a key feature of the bound pronoun: *its gender!*

(45) **Important Fact**
When we combine our semantics for pronominal gender (46) with our semantic treatment of pronominal binding, some interesting predictions emerge…

(46) **Semantic Assumptions Regarding Gender**

a. [[MASC]] = [$\lambda x : x \in D_e$ and x is male . x]
b. [[FEM]] = [$\lambda x : x \in D_e$ and x is female . x]

(47) **LF Yielding Bound Reading of (39a)**



(48) **Key Observation, Part 1**

- For any variable assignment *g*, [[S'$_1$]]^g = [λx_e : [[S$_1$]]$^{g(1/x)}$ = T]

- By definition, the function '[λx_e : [[S$_1$]]$^{g(1/x)}$ = T]' takes an entity α as argument, and yields T *iff* [[S$_1$]]$^{g(1/\alpha)}$ = T

- **Clearly, then this function only returns a value for some entity α if [[S$_1$]]$^{g(1/\alpha)}$ is defined.**
  - After all, if [[S$_1$]]$^{g(1/\alpha)}$ cannot be computed, then the meta-language expression '[[S$_1$]]$^{g(1/\alpha)}$ = T' is meaningless.

(49) **Key Observation, Part 2**
If the entity α is *male*, then [[S$_1$]]$^{g(1/\alpha)}$ is defined.

- To compute the value of [[S$_1$]]$^{g(1/\alpha)}$, we will need to compute the value of [[DP]]$^{g(1/\alpha)}$ (the reader is strongly encouraged to confirm this)

- Given our rule of PR, [[PRO$_1$]]$^{g(1/\alpha)}$ = α

- By assumption, α is male, and so [[PRO$_1$]]$^{g(1/\alpha)}$ is in the domain of [[MASC]]$^{g(1/\alpha)}$

- Thus, a value for [[DP]]$^{g(1/\alpha)}$ can be computed. Indeed, it is equal to α. As the reader is invited to confirm, the remainder of the semantic computation for [[S$_1$]]$^{g(1/\alpha)}$ proceeds without problem.

(50) **Key Observation, Part 3**
If the entity α is ***female***, then [[S$_1$]]$^{g(1/\alpha)}$ is **not** defined.

- To compute the value of [[S$_1$]]$^{g(1/\alpha)}$, we will need to compute the value of [[DP]]$^{g(1/\alpha)}$ (the reader is strongly encouraged to confirm this)

- Given our rule of PR, [[PRO$_1$]]$^{g(1/\alpha)}$ = α

- By assumption, α is *female*, and so [[PRO$_1$]]$^{g(1/\alpha)}$ is *not* in the domain of [[MASC]]$^{g(1/\alpha)}$

- Thus, a value for [[DP]]$^{g(1/\alpha)}$ *cannot* be computed. Consequently, a value for [[S$_1$]]$^{g(1/\alpha)}$ cannot be computed. Thus, by definition, [[S$_1$]]$^{g(1/\alpha)}$ is **undefined** when α is female.

(51) **Key, Interim Conclusion**
Putting together the observations in (48)-(50), we come to the following conclusion:

Given the presence of the gender feature MASC in (47), the extension of S' – *i.e.,* the function [λx_e : $\underline{[[\ S_1\ ]]^{g(1/x)} = T}$] – is *only* defined for entities that are *male*.

- After all, if an entity α is female, then $[[\ S_1\ ]]^{g(1/\alpha)}$ is not defined
- Thus, '[λx_e : $\underline{[[\ S_1\ ]]^{g(1/x)} = T}$](α)' is also undefined.
- Thus, the only entities in the *domain* of [λx_e : $\underline{[[\ S_1\ ]]^{g(1/x)} = T}$] are males.
- **Thus, $[[S']]^g$ = [$\lambda x : x \in D_e$ and x is male . $\underline{[[\ S_1\ ]]^{g(1/x)} = T}$]**

(52) **General Conclusion**

Informally speaking, our semantics for pronominal 'phi-features' (*e.g.* gender) predicts that when a pronoun is bound, the presuppositions introduced by the phi-features (*e.g.* gender) are essentially 'inherited' by the lambda operator binding the pronoun.

$$[\ \lambda x_e : [[\ldots [\text{ Feature PRO}_i\ ] \ldots]]^{g(i/x)}\ ] \quad =$$

$$[\ \lambda x : x \in D_e \textbf{ and } \mathbf{x \in Domain([[Feature]])}\ .\ [[\ldots \text{PRO}_i \ldots]]^{g(i/x)}\ ]$$

This prediction in (51)/(52) appears to be accurate in a number of cases…
*We can see this by examining the predicted T-conditions of sentences where "**him**" is bound by a DP of the form "**every NP**".*

(53) **Truth-Conditions Predicted for (47)**

a. $[[\ S''_1\ ]]^g$ = (by FA)

b. $[[\ DP_1\ ]]^g$ ($[[S']]^g$) = (by FA, (51)/(52))

c. [$\lambda g_{<et>}$: $\underline{\text{for all y, if y is a boy, then } g(y) = T}$]
([$\lambda x : x \in D_e$ and x is male . $\underline{[[\ S_1\ ]]^{g(1/x)} = T}$]) = (by other rules)

d. [$\lambda g_{<et>}$: $\underline{\text{for all y, if y is a boy, then } g(y) = T}$]
([$\lambda x : x \in D_e$ and x is male . $\underline{\text{x likes the girl who likes x}}$]) = (by LC)

e. for all y, if y is a boy, then
[$\lambda x : x \in D_e$ and x is male . $\underline{\text{x likes the girl who likes x}}$](y) = T

(54) **Another Key Observation**

- As computed above, the LF in (47) is true *iff* every boy y is such that
  $[\lambda x : x \in D_e$ and x is male . x likes the girl who likes x $](y) = T$

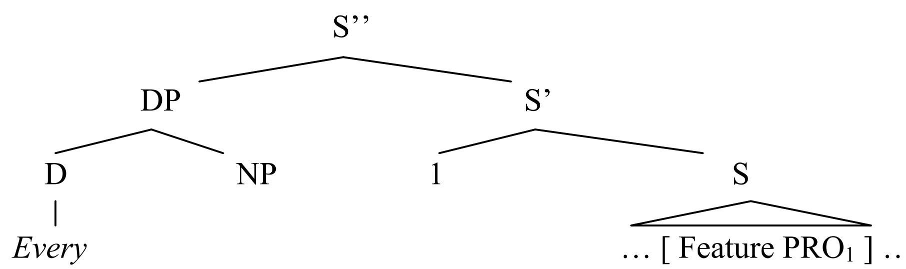
- However, this means that every boy y must be in the domain of the function
  $[\lambda x : x \in D_e$ and x is male . x likes the girl who likes x $]$

- This, of course means that every boy is male.

- Since every boy *is* male, this all works out… but it does establish the following general point.

(55) **The Central Point of All of This**
Given our presuppositional semantics for 'phi-features' (*i.e.,* gender), the following is a general prediction of our system:

If a sentence S has the form below, it can only be true if every entity x such that [[NP]](x) = T is also in the domain of [[Feature]].



(56) **Some Evidence**

Consider the sentence in (a) under the bound interpretation in (b).

a. *Sentence:* Every student likes the girl who likes him.
b. *Interpretation:* For all x, if x is a student, then x likes the girl who likes x.

Under this bound interpretation, sentence (a) does seem to entail that *every student is male*.

The reader is highly encouraged to confirm that our theory predicts the facts in (56).