Semantics Bootcamp Part IV: Variables

Elizabeth Coppock

NASSLLI 2012, Austin, Texas

Contents

1 Predicate calculus: now with variables!

Yesterday, I ignored formulas of predicate calculus with quantifiers and variables like these:

$\forall x\ \textsc{happy}(x)$ ‘for all x, x is happy’
$\exists x\ \neg\textsc{happy}(x)$ ‘there exists an x such that it is not the case that x is happy’

In order to interpret formulas with variables, we need to make interpretation relative to a model *and an assignment function*:

$$[\![\phi]\!]^{M,g}$$

An assignment function assigns individuals to variables. Examples:

$$g_1 = \begin{bmatrix} x & \rightarrow & \text{Maggie} \\ y & \rightarrow & \text{Bart} \\ z & \rightarrow & \text{Bart} \end{bmatrix}$$

$$g_2 = \begin{bmatrix} x & \rightarrow & \text{Bart} \\ y & \rightarrow & \text{Bart} \\ z & \rightarrow & \text{Bart} \end{bmatrix}$$

Informally, $\forall x\ \textsc{happy}(x)$ is true iff: no matter which individual we assign to x, $\textsc{happy}(x)$ is true. In other words, for all elements in the domain d, $d \in [\![\textsc{happy}]\!]$.

Informally, $\exists x\ \textsc{happy}(x)$ is true iff: we can find some individual to assign to x such that $\textsc{happy}(x)$ is true. In other words, there is some element in the domain d such that $d \in [\![\textsc{happy}]\!]$.

The assignment function determines what x is assigned to. Formally:

$$[\![x]\!]^{M,g} = g(x)$$

This in turn influences the value of a formula containing x in which x is not bound by a quantifier (a formula in which x is *free*).

Let's interpret $\textsc{happy}(x)$ using the reality model M_r and the assignment functions g_1 and g_2.

$[\![\textsc{happy}(x)]\!]^{M_r,g_1} = 1$
iff $[\![x]\!]^{M_r,g_1} \in [\![\textsc{happy}]\!]^{M_r,g_1}$
iff $g_1(x) \in I_r(\textsc{happy})$
iff Maggie $\in \{$Bart$\}$.

Maggie $\notin \{$Bart$\}$ so $[\![\textsc{happy}(x)]\!]^{M_r,g_1} = 0$.

$[\![\textsc{happy}(x)]\!]^{M_r,g_2} = 1$
iff $[\![x]\!]^{M_r,g_2} \in [\![\textsc{happy}]\!]^{M_r,g_2}$
iff $g_2(x) \in I_r(\textsc{happy})$
iff Bart $\in I_r(\textsc{happy})$.

Bart $\in \{$Bart$\}$ so $[\![\textsc{happy}(x)]\!]^{M_r,g_2} = 1$.

Intuitively, this means that $\exists x\ \textsc{happy}(x)$ is true, but $\forall x\ \textsc{happy}(x)$ is false in M_r.

New interpretation rules:

- **Constants**
  If α is a constant, then $[\![\alpha]\!]^{M,g}$ = I(α).

- **Variables – all new!**
  If α is a variable, then $[\![\alpha]\!]^{M,g} = g(\alpha)$.

- **Atomic formulae**
  If π is an n-ary predicate and $\alpha_1, ...\alpha_n$ are terms, then $[\![\pi(\alpha_1, ..., \alpha_n)]\!]^{M,g} = 1$ iff
  $$\langle [\![\alpha_1]\!]^{M,g}, ..., [\![\alpha_n]\!]^{M,g} \rangle \in [\![\pi]\!]^{M,g}$$
  If π is a unary predicate and α is a term, then $[\![\pi(\alpha)]\!]^{M,g} = 1$ iff $[\![\alpha]\!]^{M,g} \in [\![\pi]\!]^{M,g}$.

- **Negation**
  $[\![\neg\phi]\!]^{M,g} = 1$ if $[\![\phi]\!]^{M,g} = 0$; otherwise $[\![\neg\phi]\!]^{M,g} = 0$.

- **Connectives**
  $[\![\phi \wedge \psi]\!]^{M,g} = 1$ if $[\![\phi]\!]^{M,g} = 1$ and $[\![\psi]\!]^{M,g} = 1$; 0 otherise. Similarly for $[\![\phi \vee \psi]\!]^{M,g}$, $[\![\phi \rightarrow \psi]\!]^{M,g}$, and $[\![\phi \leftrightarrow \psi]\!]^{M,g}$.

- **Universal quantification – all new!**
  $[\![\forall v \phi]\!]^{M,g} = 1$ iff for all $d \in D$, $[\![\phi]\!]^{M,g'} = 1$, where g' is an assignment function exactly like g except that $g'(v) = d$.

- **Existential quantification – all new!**
  $[\![\exists v \phi]\!]^{M,g} = 1$ iff there is a $d \in D$ such that $[\![\phi]\!]^{M,g'}$, where g' is an assignment function exactly like g except that $g'(v) = d$.

$[\![\forall x\ \textsc{happy}(x)]\!]^{M_r,g_1} = 1$
iff for all $d \in D$, $[\![\textsc{happy}(x)]\!]^{M_r,g'} = 1$, where g' is an assignment function exactly like g_1 except that $g'(x) = d$

This can be falsified by setting d equal to Maggie, so $g'(x)$ =Maggie.
$[\![\textsc{happy}(x)]\!]^{M_r,g'} = 0$ in this case.

But there is a $d \in D$ such that $[\![\textsc{happy}(x)]\!]^{M_r,g'}$, where g' is an assignment function exactly like g_1 except that $g'(x) = d$.

As shown above, there is such a d: Bart.

2 Relative clauses

Heim and Kratzer use assignment functions for the interpretation of relative clauses such as the following:

(1) The car **that Joe bought** is very fancy.

(2) The woman **who admires Joe** is very lovely.

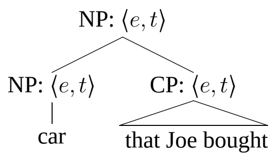
Semantically, relative clauses are just like adjectives:

(3) The **red** car is very fancy.

(4) The **Swedish** woman is very lovely.

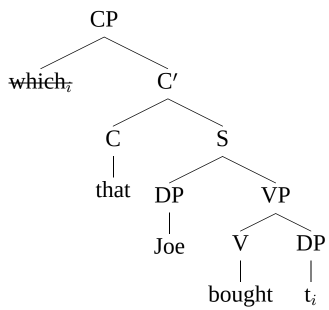
They are type $\langle e, t\rangle$ and combine via Predicate Modification.

(5)

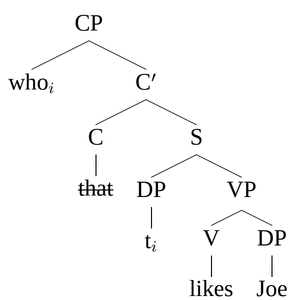


CP stands for "Complementizer Phrase" and Heim and Kratzer assume the following syntax for relative clause CPs:

(6)



(7)



The text that is struck out like ~~so~~ is *deleted*. Heim and Kratzer assume that either the relative pronoun *which* or *who* or the complementizer *that* is deleted.

Interpretation of variables

(8) **Traces Rule (TR)**
If α_i is a trace and g is an assignment, $[\![\alpha_i]\!]^g = g(i)$

$$g_1 = \begin{bmatrix} 1 & \rightarrow & \text{Maggie} \\ 2 & \rightarrow & \text{Bart} \\ 3 & \rightarrow & \text{Maggie} \end{bmatrix} \qquad g_2 = \begin{bmatrix} 1 & \rightarrow & \text{Lisa} \\ 2 & \rightarrow & \text{Bart} \\ 3 & \rightarrow & \text{Maggie} \end{bmatrix}$$

$[\![\mathbf{t}_1]\!]^{g_1}$ = Maggie

$[\![\mathbf{t}_1]\!]^{g_2}$ = Lisa

So now we interpret everything with respect to an assignment.

(9) $[\![\ [_{\text{VP}}\ [_{\text{V}}\ \text{abandoned}\ ]\ [_{\text{DP}}\ \text{t}_1\ ]\ ]]\!]^g = \lambda x\ .\ x\ \text{abandoned}\ g(1)$

But there are *assignment-independent* denotations too.

(10) **Bridge to assignment-independence (BI)**
For any tree α, α is in the domain of $[\![\,]\!]$ iff for all assignments g and g', $[\![\alpha]\!]^g = [\![\alpha]\!]^{g'}$.
If α is in the domain of $[\![\,]\!]$, then for all assignments g, $[\![\alpha]\!] = [\![\alpha]\!]^g$.

So we can still have assignment-independent lexical entries like:

(11) $[\![\textbf{laugh}]\!] = \lambda x \in D_e\ .\ x$ laughs

and then by (10), we have:

(12) $[\![\textbf{laugh}]\!]^{g_1} = \lambda x \in D_e\ .\ x$ laughs

(13) $[\![\textbf{laugh}]\!]^{g_2} = \lambda x \in D_e\ .\ x$ laughs

We need to redo the composition rules now too:

(14) **Lexical Terminals (LT)**
If α is a terminal node occupied by a lexical item, then $[\![\alpha]\!]$ is specified in the lexicon.

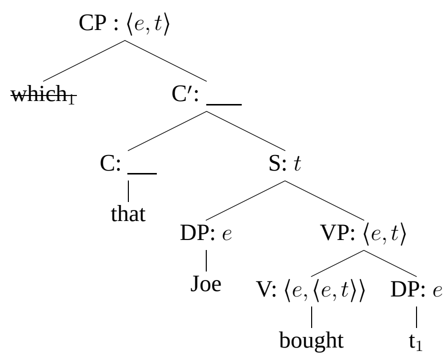
(15) **Non-branching Nodes (NN)**
If α is a non-branching node and β its daughter, then, for any assignment g, $[\![\alpha]\!]^g = [\![\beta]\!]^g$.

(16) **Functional Application (FA)**
If α is a branching node and $\{\beta, \gamma\}$ the set of its daughters, then, for any assignment g, if $[\![\beta]\!]^g$ is a function whose domain contains $[\![\gamma]\!]^g$, then $[\![\alpha]\!]^g = [\![\beta]\!]^g([\![\gamma]\!]^g)$.

(17) **Predicate Modification (PM)**
If α is a branching node and $\{\beta, \gamma\}$ the set of its daughters, then, for any assignment g, if $\llbracket\beta\rrbracket^g$ and $\llbracket\gamma\rrbracket^g$ are both functions of type $\langle e, t\rangle$, then $\llbracket\alpha\rrbracket^g$ = λx ∈ D . $\llbracket\beta\rrbracket^g$(x) = $\llbracket\gamma\rrbracket^g$(x) = 1.

Predicate abstraction. The S in a relative clause is type t. How do we get the CP to have type $\langle e, t\rangle$?

(18)



Heim and Kratzer:

- The complementizer *that* is vacuous; *that S = S*
  or $\llbracket$that$\rrbracket$ = λp ∈ D_t . p
- The relative pronoun is vacuous too, but it triggers a special rule called Predicate Abstraction

(19) **Predicate Abstraction (PA)**
If α is a branching node whose daughters are a relative pronoun indexed i and β, then $\llbracket\alpha\rrbracket^g$ = λx ∈ D_e . $\llbracket\beta\rrbracket^{g^{x/i}}$

$g^{x/i}$ is an assignment that is just like g except that x is assigned to i.

Note that x is a variable that is part of the *meta-language*, bound by the *meta-language* operator λ, ranging over objects in the domain.

So $\llbracket(18)\rrbracket$ = λx ∈ D_e . Joe bought x.

In case you don't believe me:

$\llbracket[_{CP}$ ~~which~~$_1$ $[_{C'}$ $[_C$ that] $[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]]]$\rrbracket$

= $\llbracket[_{CP}$ ~~which~~$_1$ $[_{C'}$ $[_C$ that] $[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]]]$\rrbracket^g$, for all g — BI

= λx . $\llbracket[_{C'}$ $[_C$ that] $[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]]$\rrbracket^{g^{x/1}}$ — PA

= λx . $\llbracket[_C$ that]$\rrbracket^{g^{x/1}}$($\llbracket[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]$\rrbracket^{g^{x/1}}$) — FA

= λx . $\llbracket[_C$ that]$\rrbracket$($\llbracket[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]$\rrbracket^{g^{x/1}}$) — BI

= λx . $\boxed{\lambda p \in D_t .\ p}$($\llbracket[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]$\rrbracket^{g^{x/1}}$) — LT

= λx . $\llbracket[_S$ $[_{DP}$ Joe] $[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]]$\rrbracket^{g^{x/1}}$ — β-reduction

= λx . $\llbracket[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]$\rrbracket^{g^{x/1}}$($\llbracket[_{DP}$ Joe]$\rrbracket^{g^{x/1}}$) — FA

= λx . $\llbracket[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]$\rrbracket^{g^{x/1}}$($\llbracket[_{DP}$ Joe]$\rrbracket$) — BI

= λx . $\llbracket[_{VP}$ $[_V$ bought] $[_{DP}$ t_1]]$\rrbracket^{g^{x/1}}$(Joe) — NN, LT

= λx . $\boxed{\llbracket[_V \text{ bought }]\rrbracket^{g^{x/1}}(\llbracket[_{DP}\ t_1\ ]\rrbracket^{g^{x/1}})}$(Joe) — FA

= λx . $\boxed{\llbracket[_V \text{ bought }]\rrbracket(\llbracket[_{DP}\ t_1\ ]\rrbracket^{g^{x/1}})}$(Joe) — BI

= λx . $\boxed{\boxed{\lambda z .\ \lambda y .\ y \text{ bought } z}(\llbracket[_{DP}\ t_1\ ]\rrbracket^{g^{x/1}})}$(Joe) — LT, NN

= λx . $\boxed{\boxed{\lambda z .\ \lambda y .\ y \text{ bought } z}(\llbracket t_1\rrbracket^{g^{x/1}})}$(Joe) — NN

= λx . $\boxed{\boxed{\lambda z .\ \lambda y .\ y \text{ bought } z}(x)}$(Joe) — TR

= λx . $\boxed{\lambda y .\ y \text{ bought } x}$(Joe) — β-reduction

= λx . Joe bought x — β-reduction

3 Quantifiers

How do we analyze sentences like the following:

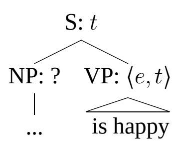
(20) Somebody is happy.

(21) Everybody is happy.

(22) Nobody is happy.

(23) {Some, every, at least one, at most one, no} linguist is happy.

(24) {Few, some, several, many, most, more than two} linguists are happy.



3.1 Type e?

Most of the DPs we have seen so far have been of type e:

- Proper names: Mary, John, Rick Perry, 4, Texas
- Definite descriptions: the governor of Texas, the square root of 4
- Pronouns and traces: it, t

Exception: indefinites like *a Republican* after *is*.

Should words and phrases like *Nobody* and *At least one person* be treated as type e? How can we tell?

Predictions of the type e analysis:

- They should validate subset-to-superset inferences
- They should validate the law of contradiction
- They should validate the law of the excluded middle

Subset-to-superset inferences

(25) John came yesterday morning.
Therefore, John came yesterday.

This is a valid inference if John is type e. Proof: ⟦came yesterday morning⟧ $\subseteq$ ⟦came yesterday⟧ (everything that came yesterday morning came yesterday), and if the subject denotes an individual, then the sentence means that the subject is an element of the set denoted by the VP. If the first sentence is true, then the subject is an element of the set denoted by the VP, which means that the second sentence must be true. QED.

(26) At most one letter came yesterday morning.
Therefore, at most one letter came yesterday.

This inference is <u>not</u> valid, so *at most one letter* must not be type e.

The law of contradiction $(\neg[P \wedge \neg P])$

This sentence is contradictory:

(27) Mount Rainier is on this side of the border, and Mount Rainier is on the other side of the border.

The fact that it is contradictory follows from these assumptions:

- ⟦Mount Rainier⟧ $\in$ D_e
- ⟦is on this side of the border⟧ $\cap$ ⟦is on the other side of the border⟧ = $\varnothing$ (Nothing is both on this side of the border and on the other side of the border)
- When the subject is type e, the sentence means that it is in the set denoted by the VP
- standard analysis of *and*

This sentence is not contradictory:

(28) More than two mountains are on this side of the border, and more than two mountains are on the other side of the border.

So *more than two mountains* must not be type e.

The law of the excluded middle $(P \vee \neg P)$

(29) I am over 30 years old, or I am under 40 years old.

This is a tautology. That follows from the following assumptions:

- $[\![\text{I}]\!] \in D_e$
- $[\![\text{over 30 years old}]\!] \cup [\![\text{under 40 years old}]\!] = D$ (everything is either over 30 years old or under 40 years old)
- When the subject is type e, the sentence means that it is in the set denoted by the VP
- standard analysis of *or*

This sentence is not a tautology:

(30) Every woman in this room is over 30 years old, or every woman in this room is under 40 years old.

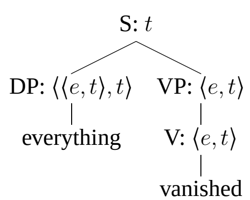
So *every woman* must not be of type e

3.2 Solution: Generalized quantifiers

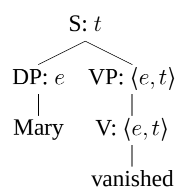
(31) $[\![\text{nothing}]\!] = \lambda f \in D_{\langle e,t \rangle}$. there is no $x \in D_e$ such that $f(x) = 1$

(32) $[\![\text{everything}]\!] = \lambda f \in D_{\langle e,t \rangle}$. for all $x \in D_e$, $f(x) = 1$

(33) $[\![\text{something}]\!] = \lambda f \in D_{\langle e,t \rangle}$. there is some $x \in D_e$ such that $f(x) = 1$

(34)


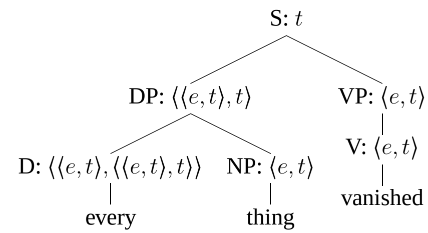
vs.



(35) $[\![\text{every}]\!] = \lambda f \in D_{\langle e,t \rangle}$. $[\lambda g \in D_{\langle e,t \rangle}$. for all $x \in D_e$ such that $f(x) = 1$, $g(x)=1$]

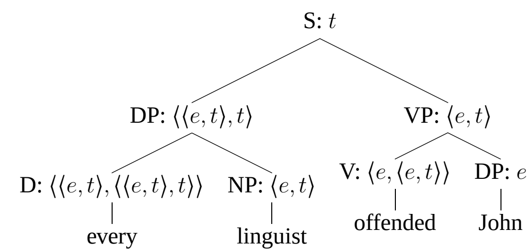
(36) $[\![\text{no}]\!] = \lambda f \in D_{\langle e,t \rangle}$. $[\lambda g \in D_{\langle e,t \rangle}$. there is no $x \in D_e$ such that $f(x) = 1$ and $g(x)=1$]

(37) $[\![\text{some}]\!] = \lambda f \in D_{\langle e,t \rangle}$. $[\lambda g \in D_{\langle e,t \rangle}$. there is some $x \in D_e$ such that $f(x) = 1$ and $g(x)=1$]

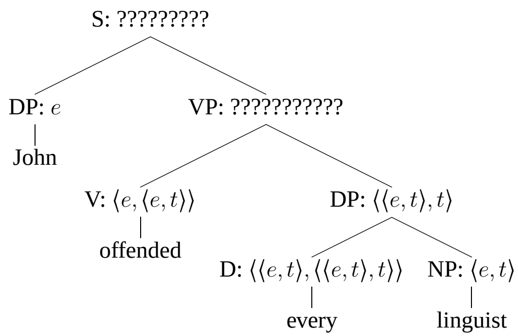
(38)


4 The problem of quantifiers in object position

4.1 The problem

(39)


(40)



Two types of approaches to the problem:

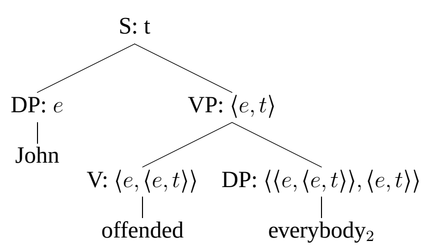
1. Move the quantifier phrase to a higher position in the tree (via Quantifier Raising), leaving a DP trace of type e in object position. (Or simulate movement via Cooper Storage, as in Head-Driven Phrase Structure Grammar.)

2. Interpret the quantifier phrase *in situ*. In this case one can apply a type-shifting operation to change its type.

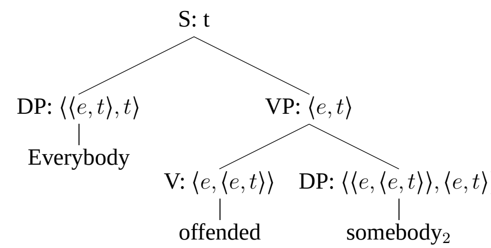
4.2 An *in situ* approach

Multiple versions of lexical items:
$[\![$everybody$_1]\!]$ = λf ∈ $D_{\langle e,t\rangle}$. for all persons x ∈ D, f(x) = 1
$[\![$everybody$_2]\!]$ = λf ∈ $D_{\langle e,\langle e,t\rangle\rangle}$. [λx ∈ D . for all persons y ∈ D, f(y)(x) = 1]
$[\![$somebody$_1]\!]$ = λf ∈ $D_{\langle e,t\rangle}$. there is some person x ∈ D such that f(x) = 1
$[\![$somebody$_2]\!]$ = λf ∈ $D_{\langle e,\langle e,t\rangle\rangle}$. [λx ∈ D . there is some person y ∈ D such that f(y)(x) = 1]

(41)



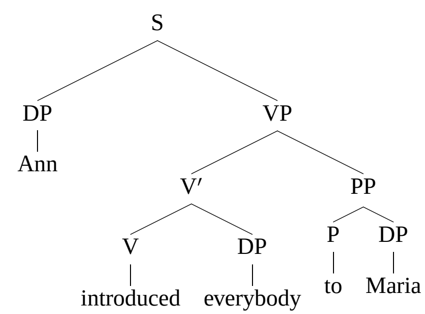
(42)



Note: This only gets one of the readings.

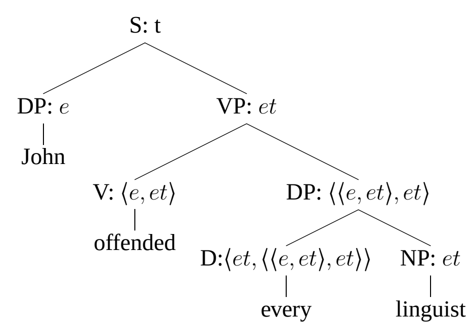
We need a new *everybody* for ternary relations:

(43)



What type are the determiners (note: et = $\langle e,t\rangle$)?

(44)



How do we get this *every* from our normal $\langle et, \langle et, t\rangle\rangle$ *every*? A lexical rule.

(45) For every lexical item δ_1 with a meaning of type $\langle et, \langle et, t\rangle\rangle$, there is a (homophonous and syntactically identical) item δ_2 with the following meaning of type $\langle et, \langle\langle e, et\rangle, et\rangle\rangle$:
$[\![\delta_2]\!] = \lambda f \in D_{\langle e,t\rangle}$. $[\lambda g \in D_{\langle e,et\rangle}$. $[\lambda x \in D$. $[\![\delta_1]\!](f)(\lambda z \in D$. g(z)(x))]]

4.3 A Quantifier Raising approach

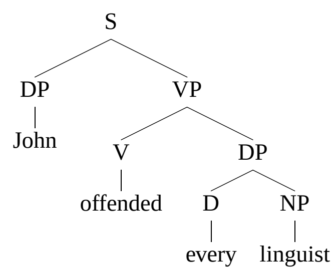
Several levels of representation:

- Deep Structure (DS): Where the derivation begins
- Surface Structure (SS): Where the order of the words is what we see
- Phonological Form (PF): Where the words are realized as sounds
- Logical Form (LF): The input to semantic interpretation

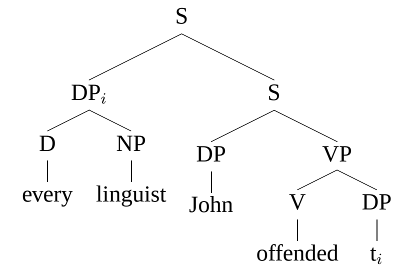
Transformations map from DS to SS, and from SS to PF and LF. (Since the transformations from SS to LF happen "after" the order of the words is determined, we do not see the output of these transformations. These movement operations are in this sense *covert*.)

A transformation called QR (Quantifier Raising) maps the SS structure in (46a) to something like the LF structure in (46b)

(46) a.

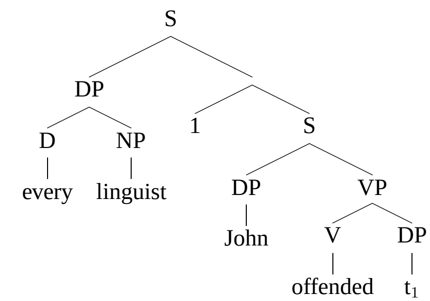


b.



Actually, Heim and Kratzer propose the following, so that they can make it work with Predicate Abstraction:

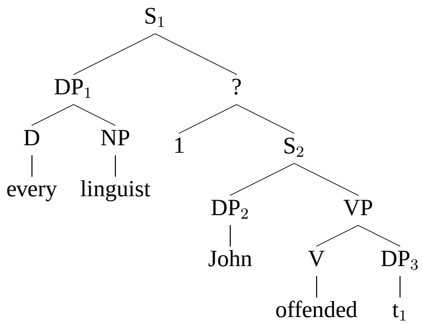
(47)



(48) **Predicate Abstraction (PA) (revised)**
Let α be a branching node with daughters β and γ, where β dominates only a numerical index i. Then for any variable assignment g, $[\![\alpha]\!]^g = \lambda x \in D$. $[\![\gamma]\!]^{g^{x/i}}$.

Example. Let's give every node of the tree a unique category label so we can refer to the denotation of the tree rooted at that node using the category label.

(49)



The task is to analyze the truth conditions of S_1 (or, to be more precise, the tree rooted at the node labelled S_1). The basic idea is straightforward – Predicate Abstraction at the mystery-category node (labelled '?' here), Pronouns and Traces rule at the trace, and Functional Application everywhere else – but it is a bit tricky to go between assignment-dependent and assignment-independent denotations. The trick is to start with the Bridge to Independence.

$[\![S_1^g]\!]$	
$=[\![S_1]\!]^g$, for all g	BI
$=[\![DP_1]\!]^g([\![?]\!]^g)$	FA
$=[\![DP_1]\!]^g(\lambda x\ .\ [\![S_2]\!]^{g^{x/1}})$	PA
$=[\![DP_1]\!]^g(\lambda x\ .\ [\![VP]\!]^{g^{x/1}}([\![DP_2]\!]^{g^{x/1}}))$	FA
$=[\![DP_1]\!]^g(\lambda x\ .\ [\![V]\!]^{g^{x/1}}([\![DP_3]\!]^{g^{x/1}})([\![DP_2]\!]^{g^{x/1}}))$	FA
$=[\![DP_1]\!]^g(\lambda x\ .\ [\![\text{offended}]\!]^{g^{x/1}}([\![t_1]\!]^{g^{x/1}})([\![\text{John}]\!]^{g^{x/1}}))$	NN, TN
$=[\![DP_1]\!]^g(\lambda x\ .\ [\![\text{offended}]\!]^{g^{x/1}}(x)([\![\text{John}]\!]^{g^{x/1}}))$	TR
$=[\![DP_1]\!]^g(\lambda x\ .\ [\![\text{offended}]\!](x)([\![\text{John}]\!]))$	BI
$=[\![DP_1]\!]^g(\lambda x\ .\ [\lambda y\ .\ \lambda z\ .\ \text{z offended y}\ ](x)(\text{John}))$	TN
$=[\![DP_1]\!]^g(\lambda x\ .\ \text{John offended x})$	β-R
$=[\![D]\!]^g([\![NP]\!]^g)(\lambda x\ .\ \text{John offended x})$	TN
$=[\![\text{every}]\!]^g([\![\text{linguist}]\!]^g)(\lambda x\ .\ \text{John offended x})$	FA
$=[\![\text{every}]\!]([\![\text{linguist}]\!])(\lambda x\ .\ \text{John offended x})$	BI
$= [\lambda f \in D_{\langle e,t\rangle}\ .\ \lambda g \in D_{\langle e,t\rangle}\ .\ \text{for all y, if f(y) then g(y)}]([\![\text{linguist}]\!])(\lambda x\ .\ \text{John offended x})$	TN
$= [\lambda g \in D_{\langle e,t\rangle}\ .\ \text{for all y, if y is a linguist then Q(y)}](\lambda x\ .\ \text{John offended x})$	TN, β-R
$=1$ iff for all y, if y is a linguist then John offended x	β-R

4.4 Arguments in favor of the movement approach

Argument #1: Scope ambiguities. In order to get both readings of *Everybody loves somebody*, we have to introduce yet even more complicated types. Scope ambiguities are trivially derived under the movement approach.

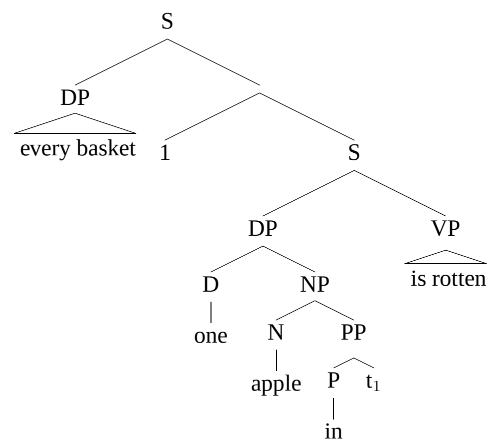
Argument #2: Inverse linking. There is one class of examples that cannot be generated under an *in situ* approach:

(50) One apple in every basket is rotten.

This does not mean: 'One apple that is in every basket is rotten'. That is the only reading that an *in situ* analysis can give us.

QR analysis:

(51)



Argument #3: Antecedent-contained deletion

(52) I read every novel that you did.

Like regular VP ellipsis:

(53) I read *War and Peace* before you did.

except that the antecedent VP is contained in the elided VP!

To create an appropriate antecedent, you have to QR the object.

Argument #4: Quantifiers that bind pronouns

(54) a. Mary blamed herself.
b. Mary blamed Mary.

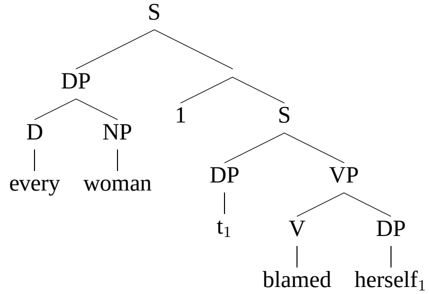
(55) a. Every woman blamed herself.
b. Every woman blamed every woman.

(56) No man noticed the snake next to him.

Treat pronouns as variables and use QR ⇒ no problem.

(57) **Traces and Pronouns Rule (TP)** (p. 116)
If α is a pronoun or trace and g is an assignment and i is in the domain of g,
$[\![\alpha_i]\!]^g = g(i)$

(58)



But how do we get the truth conditions on the in-situ approach?

(59) $[\![\ [_{VP}\ [_{V}\ \text{blamed}\ ]\ [_{DP}\ \text{herself}_1]\ ]\ ]\!]^g = \lambda x\ .\ x \text{ blamed } g(1)$

How do we combine this with *every woman*? We cannot get an assignment-independent denotation.

5 Free and Bound Variable Pronouns

5.1 Toward a unified theory of anaphora

A deictic use of a pronoun:

(60) [after a certain man has left the room:]
I am glad he is gone.

An anaphoric use of a pronoun:

(61) I don't think anybody here is interested in Smith's work. He should not be invited.

"Anaphoric and deictic uses seem to be special cases of the same phenomenon: the pronoun refers to an individual which, for whatever reason, is highly salient at the moment when the pronoun is processed." (Heim and Kratzer 1998, p. 240)

Hypothesis 1: All pronouns refer to whichever individual is most salient at the moment when the pronoun is processed.

It can't be that simple for all pronouns:

(62) the book such$_1$ that Mary reviewed it$_1$

(63) No$_1$ woman blamed herself$_1$.

So not all pronouns are referential.[1]

Hypothesis 2: All pronouns are bound variables.

Then in (61) we would have to QR *Smith* to a position where it QR's *He* in the second sentence somehow.

Plus, the strict/sloppy ambiguity exemplified in (64) can be explained by saying that on one reading, we have a bound pronoun, and on another reading, we have a

[1] Sometimes it is said that *No woman* and *herself* are "coreferential" in (63) but this is strictly speaking a misuse of the term "coreferential", because, as Heim and Kratzer point out, "coreference implies reference."

referential pronoun.

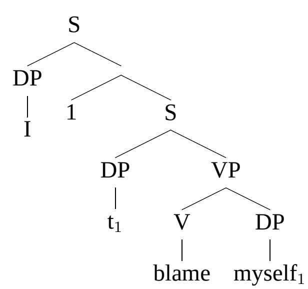
In the movie *Ghostbusters,* there is a scene in which the three Ghostbusters Dr. Peter Venkman, Dr. Raymond Stanz, and Dr. Egon Spengler (played by Bill Murray, Dan Akroyd, and Harold Ramis, respectively), are in an elevator. They have just started up their Ghostbusters business and received their very first call, from a fancy hotel in which a ghost has been making disturbances. They have their proton packs on their back and they realize that they have never been tested.

(64) Dr Ray Stantz: You know, it just occurred to me that we really haven't had a successful test of this equipment.
Dr. Egon Spengler: I blame myself.
Dr. Peter Venkman: So do I.

Strict reading: Peter blames himself.
Sloppy reading: Peter blames Egon.

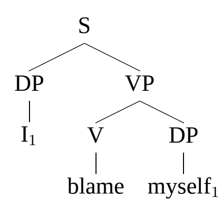
LF of antecedent for sloppy reading:

(65)



LF of antecedent for strict reading:

(66)



Heim and Kratzer's hypothesis: All pronouns are variables, and bound pronouns are interpreted as bound variables, and referential pronouns are interpreted as free variables.

What does it mean for a variable to be bound or free?

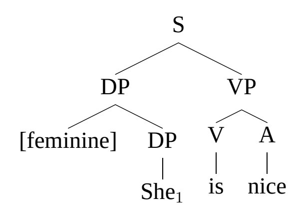
- The formal definition (p. 118): Let α^n be an occurrence of a variable α in a tree β. Then α^n is free in β if no subtree γ of β meets the following two conditions: (i) γ contains α^n, and (ii) there are assignments g such that α is not in the domain of $[\![ ]\!]^g$, but γ is.

- More intuitively: A variable is free in a tree β if the value of $[\![\beta]\!]^g$ depends on what g assigns to the variable's index.

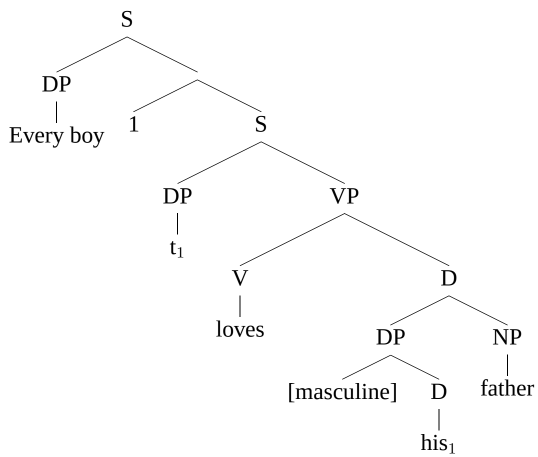
- With the Predicate Abstraction rule, we make semantic values independent of assignments, so we can use the following shortcut to determine whether a variable is bound or free: A variable is bound if there is a node that meets the structural description for Predicate Abstraction dominating it and its index; otherwise it is free.

Examples:

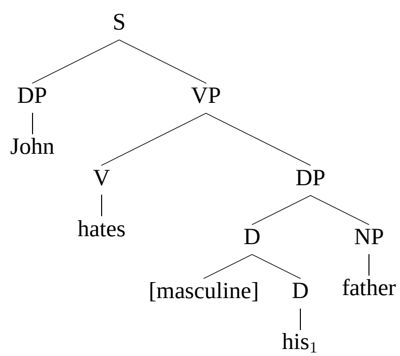
(67)



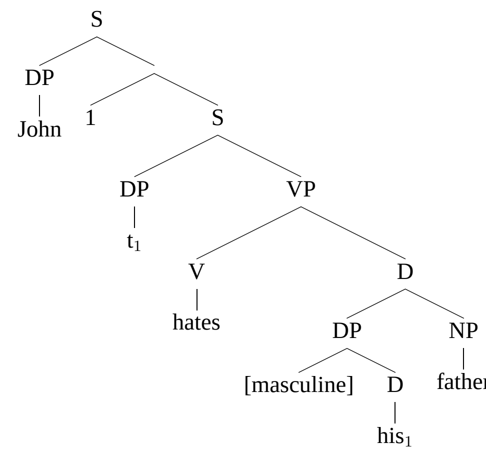
(68)



(69)



(70)



5.2 Assignments as part of the context

A consequence: "Treating referring pronouns as free variables implies a new way of looking at the role of variable assignments. Until now we have asssumed that an LF whose truth-value varied from one assignment to the next could *ipso facto* not represent a felicitous, complete utterance. We will no longer make this assumption. Instead, let us think of assignments as representing the contribution of the utterance situation."

(71) **Appropriateness Condition**
A context c is appropriate for an LF ϕ only if c determines a variable assignment g_c whose domain includes every index which has a free occurrence in ϕ.

Now truth and falsity will be relative to contexts:

(72) **Truth and Falsity Conditions for Utterances**
If ϕ is uttered in c and c is appropriate for ϕ, then the utterance of ϕ in c is *true* if $[\![\phi]\!]^{g_c} = 1$ and *false* if $[\![\phi]\!]^{g_c} = 0$.

Suppose the context c_1 "determines" the assignment g_1 (i.e., the context contains a bunch of information, among which is the assignment g_1), and g_1 is defined as follows:

$$g_1 = \begin{bmatrix} 1 \rightarrow \text{Kim} \\ 2 \rightarrow \text{Kim} \\ 3 \rightarrow \text{Sandy} \end{bmatrix}$$

Assume:

- Kim is male
- Sandy is female
- Kim is nice
- Sandy is not nice

$[\![$feminine$]\!]$ = λx : x is female . x

Questions:

- Is (67) appropriate for c_1?
- Is $[\![(67)]\!]^{g_1}$ defined?
- If so, what is $[\![(67)]\!]^{g_1}$? (I.e. is it true or false?)

6 Our fragment of English so far

6.1 Composition Rules

For branching nodes:

(73) **Functional Application (FA)**
If α is a branching node and $\{\beta, \gamma\}$ the set of its daughters, then, for any assignment g, if $[\![\beta]\!]^g$ is a function whose domain contains $[\![\gamma]\!]^g$, then $[\![\alpha]\!]^g$ = $[\![\beta]\!]^g([\![\gamma]\!]^g)$.

(74) **Predicate Modification (PM)**
If α is a branching node and $\{\beta, \gamma\}$ the set of its daughters, then, for any assignment g, if $[\![\beta]\!]^g$ and $[\![\gamma]\!]^g$ are both functions of type $\langle e, t\rangle$, then $[\![\alpha]\!]^g$ = λx $\in$ D . $[\![\beta]\!]^g$(x) = $[\![\gamma]\!]^g$(x) = 1.

(75) **Predicate Abstraction (PA)**
Let α be a branching node with daughters β and γ, where β dominates only a numerical index i. Then for any variable assignment g, $[\![\alpha]\!]^g$ =λx $\in$ D . $[\![\gamma]\!]^{g^{x/i}}$

For non-branching and terminal nodes:

(76) **Non-branching Nodes (NN)**
If α is a non-branching node and β its daughter, then, for any assignment g, $[\![\alpha]\!]^g$=$[\![\beta]\!]^g$.

(77) **Lexical Terminals (LT)**
If α is a terminal node occupied by a lexical item, then $[\![\alpha]\!]$ is specified in the lexicon.

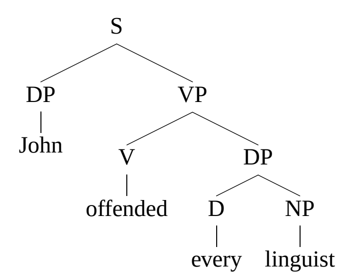
(78) **Traces and Pronouns Rule (TP)**
If α is a pronoun or trace and g is an assignment and i is in the domain of g, $[\![\alpha_i]\!]^g$ = $a(i)$

6.2 Additional principles

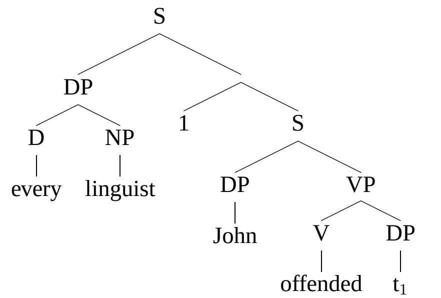
(79) **Bridge to assignment-independence**
For any tree α, α is in the domain of $[\![]\!]$ iff for all assignments g and g', $[\![\alpha]\!]^g$= $[\![\alpha]\!]^{g'}$.
If α is in the domain of $[\![]\!]$, then for all assignments g, $[\![\alpha]\!]$= $[\![\alpha]\!]^g$.

(80) **Quantifier Raising (QR)**
Surface structures containing quantificational NP like (80a) undergo a covert transformation to LFs like (80b)

a.



b.



(81) **Appropriateness Condition**
A context c is appropriate for an LF ϕ only if c determines a variable assignment g_c whose domain includes every index which has a free occurrence in ϕ.

(82) **Truth and Falsity Conditions for Utterances**
If ϕ is uttered in c and c is appropriate for ϕ, then the utterance of ϕ in c is *true* if $[\![\phi]\!]^{g_c} = 1$ and *false* if $[\![\phi]\!]^{g_c} = 0$.

(83) **LF Identity Condition on Ellipsis**
A constituent may be deleted at PF only if it is a copy of another constituent at LF.

(84) **Indexing Condition**
No LF representation (for a sentence or multisentential text) must contain both bound occurrences and free occurrences of the same index.

6.3 Lexical items

Proper names:

(85) $[\![$Rick Perry$]\!]$ = Rick Perry

(86) $[\![$Texas$]\!]$ = Texas

(87) $[\![$four$]\!]$ = 4

Nouns:

(88) $[\![$Republican$]\!] = \lambda x \in D_e$. x is a Republican

(89) $[\![$square root$]\!] = \lambda y \in D_e . \lambda x \in D_e$. x is the square root of y

Adjectives:

(90) $[\![\text{conservative}_{\langle e,t\rangle}]\!] = \lambda x \in D_e$. x is conservative

(91) $[\![\text{conservative}_{\langle\langle e,t\rangle,\langle e,t\rangle\rangle}]\!] = \lambda f \in D_{\langle e,t\rangle} . \lambda x \in D_e$. f(x) = 1 and x is conservative

(92) $[\![$negative$]\!] = \lambda x \in D_e$. x is negative

(93) $[\![$proud$]\!] = \lambda y \in D_e . \lambda x \in D_e$. x is proud of y

Verbs:

(94) $[\![$is$]\!] = \lambda f \in D_{\langle e,t\rangle}$. f

Prepositions:

(95) $[\![$in$]\!] = \lambda y \in D_e . \lambda x \in D_e$. x is in y

(96) $[\![$of$]\!] = \lambda x \in D_e$. x

Determiners:

(97) $[\![$a$]\!] = \lambda f \in D_{\langle e,t\rangle}$. f

(98) $[\![$the$]\!] = \lambda f \in D_{\langle e,t\rangle}$: there is exactly one x such that f(x) = 1 . the unique y such that f(y) = 1

Complementizers: $[\![$that$]\!] = \lambda p \in D_t$. p

Conclusion

Tiny bit more to do before we're done with English.

Luckily, we have the rest of the week.