An Algebraic Approach to Quantification and Lambda Abstraction: Preliminaries [1]

(1) **Our Analytic Toolbox (The Framework)**

a. A Theory of Syntax
   (i) 'Disambiguated' languages as quintuples $\langle A, F_\gamma, X_\delta, S, \delta_0\rangle_{\gamma \in \Gamma, \delta \in \Delta}$
   (ii) Languages as pairs $\langle \langle A, F_\gamma, X_\delta, S, \delta_0\rangle_{\gamma \in \Gamma, \delta \in \Delta}, R\rangle$

b. A Theory of Semantics
   (i) Interpretations as structures $\langle B, G_\gamma, f\rangle_{\gamma \in \Gamma}$
   (ii) Meaning assignments as homomorphisms from $\langle A, F_\gamma\rangle_{\gamma\in\Gamma}$ to $\langle B, G_\gamma\rangle_{\gamma\in\Gamma}$

c. A Theory of Translations Inducing Interpretations
   (i) Translation bases as structures $\langle g, H_{\gamma'}, j\rangle_{\gamma' \in \Gamma'}$
   (ii) Translations as homomorphisms from $\langle A', F_{\gamma'}\rangle_{\gamma' \in \Gamma'}$ to $\langle A, H_{\gamma'}\rangle_{\gamma' \in \Gamma'}$
   (iii) Translation from *L*' to *L* induces the interpretation $\langle B, G'_{\gamma'}, f'\rangle_{\gamma' \in \Gamma'}$

We're now going to apply these analytic tools (framework) to the phenomenon of quantification, both in English and in logical languages…

(2) **An Outline of the Plan**

a. Step One: Introduce a logical language with quantification.

- Since we're ultimately going to use this logical language to indirectly interpret English quantificational NPs, this language will also include 'λ'

b. Step Two: Introduce a model-theoretic semantics for the language

- This will give us a more comprehensible means for establishing certain key semantic properties of the language.
- It will also provide the basis for Step Three

c. Step Three: Introduce a (Montagovian) interpretation structure for the language

- **This is the intellectually hardest step.**

d. Step Four: Introduce a translation base from English to the logical language.

- **This takes work, but the key ideas will already be familiar to you.**

[1] These notes are based upon material in the following readings: Partee *et al.* (1993) Chapter 13, Dowty *et al.* (1981) Chapter 4, and Thomason (1974) Chapter 7 (Montague's "Universal Grammar").

1. The Logical Language TL: Non-Montagovian Presentation

I assume that everyone has a basic familiarity with lambda abstraction and its applications to natural language semantics.

- For a review, the student is referred to Heim & Kratzer (1998: 34-40) and Partee *et al.* (1993: 336-369).

(3) **Typed Logic (TL)**
A typed logic (language) is a language whose vocabulary of symbols satisfies the conditions in (4), and whose syntax satisfies the conditions in (7).

(4) **The Vocabulary of a TL**

a. The Logical Constants: [2]
(i) *Sentence Connectives:* ~, &, ∨, →
(ii) *Quantifiers:* ∀, ∃
(iii) ***Lambda Operator:*** **λ**

b. The Syntactic Symbols: (,)

c. The Non-Logical Constants:
(i) *Constants:*
For every type $\tau \in T$, a **countable** set of constants of type τ, CON_τ

(ii) *Variables:*
For every type $\tau \in T$, a **countably infinite** set of variables of type τ: $VAR_\tau = \{ v_{\tau, n} : n \in \mathbb{N} \}$

(5) **Remarks**

a. Note that in (4c), we are using the types to categorize expressions of our language, just as in our Montagovian definitions of FOL-NoQ and Politics-NoQ.

b. Given (4ci), we will have individual constants (CON_e) and for any $n \in \mathbb{N}$, a set of n-ary predicate constants ($CON_{<e,t>}$, $CON_{<e,<e,t>>}$, $CON_{<e<e<e,t>>>}$, …)

c. Given (4ci), we can also have constants of type *t*, <<e,t>,t>, <e,e>, etc.
   - Given our intended applications, we won't make use of such constants here.

d. Given (4cii), our variables now come with a subscript indicating their type.
   - $v_{e,3}$ $v_{<e,t>,4}$ $v_{<e,<e,t>>,2}$ $v_{<<e,t>,t>,6}$ $v_{t,190}$

---

[2] We'll now make use of a larger set of primitive logical operators. This will make the statement of our semantics longer, but it will later simplify our analysis of English.

(6) **Meta-Language Abbreviations for Variables**
Although our variables 'officially' all look like those in (5d), to save space we will make use of the following meta-language abbreviations for variables of type *e* and <*e,t*>:

a. $x_n = v_{e,n}$ b. $P_n = v_{<e,t>,n}$

In our (relatively) informal syntaxes for PL, FOL, FOL-NoQ, Politics, and Politics-NoQ, our syntax defined a set of 'WFFs'…

- But, now that our logical language contains 'λ', it will be more efficient to arrange our syntax so that it defines sets of 'meaningful expressions' of various types $\tau \in T$ (ME_τ)

(7) **The Syntax of a TL**

a. If $\varphi \in CON_\tau$ or $\varphi \in VAR_\tau$, then $\varphi \in ME_\tau$

b. If $\varphi \in ME_{<\sigma,\tau>}$ and $\psi \in ME_\sigma$, then $(\varphi\ \psi) \in ME_\tau$

c. If $\varphi, \psi \in ME_t$, then
(i) $\sim\varphi \in ME_t$
(ii) $(\varphi\ \&\ \psi) \in ME_t$
(iii) $(\varphi \vee \psi) \in ME_t$
(iv) $(\varphi \rightarrow \psi) \in ME_t$

d. If $v \in VAR_\tau$, and $\varphi \in ME_t$, then
(i) $\exists v\varphi \in ME_t$
(ii) $\forall v\varphi \in ME_t$

e. **If $v \in VAR_\sigma$, and $\varphi \in ME_\tau$, then $(\lambda v\ \varphi) \in ME_{<\sigma,\tau>}$**

(8) **Illustration of a TL Language: Politics+λ**

a. Vocabulary of Politics+λ
(i) *Logical Constants:* (as in (4a))
(ii) *Syntactic Symbols:* (,)
(iii) *Non-Logical Constants*
1. Variables (as in (4cii))
2. Constants: **CON_e = { mitt', barack', michelle' }**
$CON_{<et>}$ = { smokes', man', president' }
$CON_{<e<e,t>>}$ = { loves' }
For all other $\tau \in T$, $CON_\tau = \varnothing$

b. Syntax of Politics+λ: (as in (7))

(9) **Some Illustrative Meaningful Expressions of Politics+λ**

a. ~ ((**loves' mitt'**) **barack'**)

(i) **loves'** $\in CON_{<e<e,t>>}$, **mitt'**, **barack'** $\in CON_e$ (8a)
(ii) **loves'** $\in ME_{<e<e,t>>}$, **mitt'**, **barack'** $\in ME_e$ (7a)
(iii) (**loves' mitt'**) $\in ME_{<e,t>}$ (7b)
(iv) ((**loves' mitt'**) **barack'**) $\in ME_t$ (7b)
(v) ~ ((**loves' mitt'**) **barack'**) $\in ME_t$ (7c)

b. $\forall x_3$ ((**smokes'** x_3) → ~ ((**loves' mitt'**) x_3))

(i) **loves'** $\in CON_{<e<e,t>>}$, **smokes'** $\in CON_{<e,t>}$, **mitt'**, **barack'** $\in CON_e$, $x_3 \in VAR_e$ (8a)
(ii) **loves'** $\in ME_{<e<e,t>>}$, **smokes'** $\in ME_{<e,t>}$, **mitt'**, **barack'**, $x_3 \in ME_e$ (7a)
(iii) (**loves' mitt'**) $\in ME_{<e,t>}$ (7b)
(iv) ((**loves' mitt'**) x_3) $\in ME_t$ (7b)
(v) ~((**loves' mitt'**) x_3) $\in ME_t$ (7c)
(vi) (**smokes'** x_3) $\in ME_t$ (7b)
(vii) ((**smokes'** x_3) → ~((**loves' mitt'**) x_3)) $\in ME_t$ (7c)
(viii) $\forall x_3$ ((**smokes'** x_3) → ~ ((**loves' mitt'**) x_3)) $\in ME_t$ (7d)

c. $\exists P_4$ (P_4 **michelle'**)

(i) **michelle'** $\in CON_e$, $P_4 \in VAR_{<e,t>}$ (8a)
(ii) **michelle'** $\in ME_e$, $P_4 \in ME_{<e,t>}$ (7a)
(iii) (P_4 **michelle'**) $\in ME_t$ (7b)
(iv) $\exists P_4$ (P_4 **michelle'**) $\in ME_t$ (7d)

d. ((λx_3 (**man'** x_3)) **mitt'**)

(i) **man'** $\in CON_{<e,t>}$, **mitt'** $\in CON_e$, $x_3 \in VAR_e$ (8a)
(ii) **man'** $\in ME_{<e,t>}$, **mitt'**, $x_3 \in ME_e$ (7a)
(iii) (**man'** x_3) $\in ME_t$ (7b)
(iv) (λx_3 (**man'** x_3)) $\in ME_{<e,t>}$ (7e)
(v) ((λx_3 (**man'** x_3)) **mitt'**) $\in ME_t$ (7b)

e. (λP_4 $\forall x_3$ ((**man'** x_3) → (P_4 x_3)))

(i) **man'** $\in CON_{<e,t>}$, $P_4 \in VAR_{<e,t>}$, $x_3 \in VAR_e$ (8a)
(ii) **man'**, $P_4 \in ME_{<e,t>}$, $x_3 \in ME_e$ (7a)
(iii) (**man'** x_3), (P_4 x_3) $\in ME_t$ (7b)
(iv) ((**man'** x_3) → (P_4 x_3)) $\in ME_t$ (7c)
(v) $\forall x_3$ ((**man'** x_3) → (P_4 x_3)) $\in ME_t$ (7d)
(vi) (λP_4 $\forall x_3$ ((**man'** x_3) → (P_4 x_3))) $\in ME_{<<e,t>,t>}$ (7e)

2. The Semantics of a TL: Non-Montagovian Presentation

In this section, we will develop a model-theoretic semantics for TL languages.

- This model-theoretic semantics will provide the basis from which we will develop a (Montagovian) interpretation for a TL language.
- Moreover, the model-theoretic semantics will allow for a more perspicuous demonstration of a key validity in TL languages ('lambda conversion')
- Furthermore, seeing the relationship between this model-theoretic semantics and the later interpretation will lay important groundwork for our presentation of PTQ

(10) **The Denotations Based on a Set E** [3]
Let T be the set of types and E be some non-empty set (of entities). If $\tau \in T$, then the set $D_{\tau, E}$ of *denotations of type τ based on E* is defined as follows:
(i) $D_{e,E}$ = E
(ii) $D_{t,E}$ = $\{ 0, 1 \}$
(iii) If $\sigma, \tau \in T$, then $D_{<\sigma,\tau>,E}$ = the set of functions from $D_{\sigma,E}$ to $D_{\tau,E}$

(11) **Definition of a Model for a TL Language**
A model $\mathcal{M}$ for a TL language *L* is a pair <E, I> consisting of: [4]

a. A *non-empty* set E, called the 'domain of $\mathcal{M}$'

b. A function I, whose domain is equal to (i) and whose range satisfies the condition in (ii).
(i) *Domain of I:* $\cup_{\tau \in T} CON_\tau$
(ii) *Condition on Range of I:* If $\alpha \in CON_\tau$, then $I(\alpha) \in D_{\tau,E}$

(12) **Remarks**
If <E,I> is a model for a TL language, then:

a. If α is an individual constant ($\alpha \in CON_e$), then $I(\alpha)$ is a member of E ($D_{e,E}$)

b. If α is an n-ary predicate letter ($\alpha \in CON_{<e,\dots,t>}$), then $I(\alpha)$ is the curried characteristic function of an n-ary relation in E ($D_{<e,\dots,t>,E}$)

---

[3] Note that since our model-theoretic semantics won't ever interpret 'syntactic garbage', we needn't add the special element **garbage** to $D_{e,E}$.
[4] To avoid confusion with the notation in (10), the domains of models will now generally be represented as 'E'.

(13) **Illustration: A Model for Politics+λ**

Let the model $\mathcal{M}$ be the pair <{Barack, Michelle, Mitt}, I>, where I consists of the following mappings:

a. I(**michelle'**) = Michelle
b. I(**barack'**) = Barack
c. I(**mitt'**) = Mitt

d. I(**smokes'**) = *h* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }
e. I(**man'**) = *i* = { <Michelle, 0>, <Barack,1>, <Mitt,1> }
f. I(**president'**) = *k* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }

e. *f*(**loves'**)= *j* =

$$\left[\begin{array}{lcl} \text{Michelle} & \rightarrow & \left[\begin{array}{l} \text{Michelle} \rightarrow 1 \\ \text{Barack} \rightarrow 1 \\ \text{Mitt} \rightarrow 0 \end{array}\right] \\ \text{Barack} & \rightarrow & \left[\begin{array}{l} \text{Michelle} \rightarrow 1 \\ \text{Barack} \rightarrow 1 \\ \text{Mitt} \rightarrow 0 \end{array}\right] \\ \text{Mitt} & \rightarrow & \left[\begin{array}{l} \text{Michelle} \rightarrow 0 \\ \text{Barack} \rightarrow 0 \\ \text{Mitt} \rightarrow 1 \end{array}\right] \end{array}\right]$$

Given that we now have variables for types other than e, we also need a concomitant change in our definition of a variable assignment…

(14) **Variable Assignment**

Let $\mathcal{M}$ be a model <E, I> of a TL language. Then g is a *variable assignment (based on* $\mathcal{M}$*)* if its domain is equal to (i) and its range satisfies the property in (ii).

(i) *Domain of g:* $\cup_{\tau \in T} VAR_{\tau}$
(iii) *Condition on Range of g:* If $\alpha \in VAR_{\tau}$, then $g(\alpha) \in D_{\tau,E}$

(15) **Remarks**

If $\mathcal{M}$ = <E,I> is a model for a TL language, and g is a variable assignment based on $\mathcal{M}$

a. If α is an individual variable ($\alpha \in VAR_e$), then $g(\alpha)$ is a member of E ($D_{e,E}$)

b. If α is an n-ary predicate letter variable ($\alpha \in VAR_{<e,\ldots,t>}$), then $g(\alpha)$ is the curried characteristic function of an n-ary relation in E ($D_{<e,\ldots,t>,E}$)

With the definitions in (11) and (14), we can now define the notion of 'interpretation with respect to a model $\mathcal{M}$ and a variable assignment *g*, $[[.]]^{M,g}$.'

(16) **Interpretation With Respect to a Model and a Variable Assignment**
Let $\mathcal{M}$ be a model <E, I> for a TL language *L* and g be a variable assignment based on $\mathcal{M}$. The interpretation (a.k.a. denotation) of a meaningful expression of *L* relative to $\mathcal{M}$ and g $[[.]]^{M,g}$ is defined as follows:

a. If $v \in \cup_{\tau \in T} VAR_\tau$, then $[[\ v\ ]]^{M,g} = g(v)$

b. If $\alpha \in \cup_{\tau \in T} CON_\tau$, then $[[\ \alpha\ ]]^{M,g} = I(\alpha)$

c. If $\varphi = (\psi\ \chi)$, then $[[\varphi\ ]]^{M,g} = [[\psi]]^{M,g}([[\chi]]^{M,g})$

d. If $\varphi = \sim\psi$, then $[[\varphi]]^{M,g} = 1$ *iff* $[[\psi]]^{M,g} = 0$

e. If $\varphi = (\psi\ \&\ \chi)$, then $[[\varphi]]^{M,g} = 1$ *iff* $[[\psi]]^{M,g} = 1$ and $[[\chi]]^{M,g} = 1$

f. If $\varphi = (\psi \vee \chi)$, then $[[\varphi]]^{M,g} = 1$ *iff* $[[\psi]]^{M,g} = 1$ or $[[\chi]]^{M,g} = 1$

g. If $\varphi = (\psi \rightarrow \chi)$, then $[[\varphi]]^{M,g} = 1$ *iff* $[[\psi]]^{M,g} = 0$ or $[[\chi]]^{M,g} = 1$

h. If $\varphi = \exists v\psi$ **and** $\boldsymbol{v} \in \mathbf{VAR}_\tau$, then $[[\varphi]]^{M,g} = 1$ *iff*
there is an $\mathbf{a} \in \mathbf{D}_{\tau,\mathbf{E}}$ such that $[[\psi]]^{M,g(v/a)} = 1$

i. If $\varphi = \forall v\psi$ **and** $\boldsymbol{v} \in \mathbf{VAR}_\tau$, then $[[\varphi]]^{M,g} = 1$ *iff* for all $\mathbf{a} \in \mathbf{D}_{\tau,\mathbf{E}}$, $[[\psi]]^{M,g(v/a)} = 1$

j. If $\varphi = (\lambda v\psi)$, $\boldsymbol{v} \in \mathbf{VAR}_\sigma$ **and** $\psi \in \mathbf{ME}_\tau$, then $[[\varphi]]^{M,g} =$
The function *p* whose domain is $\mathbf{D}_{\sigma,\mathbf{E}}$, whose range is $\mathbf{D}_{\tau,\mathbf{E}}$ and for all $\mathbf{a} \in \mathbf{D}_{\sigma,\mathbf{E}}$,
$p(a) = [[\psi]]^{M,g(v/a)}$

(17) **Remarks**

a. Given the sorting of variables into types, along with the definition in (14), our definitions in (16h,i) allow our language to quantify, not only over entities, but **also over objects of all other types.**

b. Similarly, given the definition in (14), our definition in (16j) entails that the type of the variable in a lambda expression will determine the domain of the function denoted by the expression.

I will now use the definitions in (16) to show how our model in (13) can be used to assign semantic values to meaningful expressions of Politics+λ.

- For reasons of space, the calculations below are greatly appreviated…
- Students should work out for themselves the full calculations based on (16)…

(18) **Interpreting Expressions of Politics+λ**
Let $\mathcal{M}$ be the model defined in (13). Let g be some arbitrary variable assignment based on $\mathcal{M}$.

a. ∀x_3 (**man’** x_3)

(i) $[[\forall x_3$ (**man’** $x_3)]]^{M,g} = 1$ *iff*
(ii) For all a ∈ $D_{e,E}$, *i*(a) = 1 *iff*
(iii) For all a ∈ {Michelle, Barack, Mitt},
{ <Michelle, 0>, <Barack,1>, <Mitt,1> }(a) = 1
(iv) Thus, $[[\forall x_3$ (**man’** $x_3)]]^{M,g} = 0$

b. ∃P_4 (P_4 **michelle’**)

(i) [[∃P_4 (P_4 **michelle’**) $]]^{M,g}$ = 1 *iff*
(ii) There is an **a ∈ $D_{<et>,E}$** such that a(Michelle) = 1
(iii) Thus, [[∃P_4 (P_4 **michelle’**) $]]^{M,g}$ = 1

Note: Even though neither *h*, *i*, nor *k* map Michelle to 1, there are still other functions *f* ∈ $D_{<et>,E}$ such that *f*(Michelle) = 1

c. (λx_3 ((**man’** x_3) & (**smokes’** x_3)))

(i) [[(λx_3 ((**man’** x_3) & (**smokes’** x_3)))$]]^{M,g}$ =

(ii) The function *p* with domain $D_{e,E}$, range $D_{t,E}$, and for all a ∈ $D_{e,E}$,
p(a) = 1 iff i(a) = 1 and h(a) = 1 =

(iv) The function *p* with domain {Mitt, Barack, Michelle}, range {0,1}, and for all a ∈ {Mitt, Barack, Michelle},
p(a) = 1 iff { <Michelle, 0>, <Barack,1>, <Mitt,1> }(a) = 1 and
{ <Michelle, 0>, <Barack,1>, <Mitt,0> }(a) = 1 =

(v) **{ <Michelle, 0>, <Barack,1>, <Mitt,0> }** =

(vi) **The characteristic function of the set of ‘men who smoke’**

d. $(\lambda P_4\ (P_4\ \textbf{mitt'}))$

(i) $[[(\lambda P_4\ (P_4\ \textbf{mitt'}))]]^{M,g}$ =

(ii) The function *p* with domain $D_{<et>,E}$, range $D_{t,E}$ and for all a $\in D_{<et>,E}$,
p(a) = a(Mitt) =

(iii) **The function *p* with domain $D_{<et>,E}$, range $D_{t,E}$ and for all a $\in D_{<et>,E}$,
p(a) = 1 iff a(Mitt) = 1**

(iv) **The characteristic function of the set of <et>-functions *f* such that f(Mitt) = 1**

(v) **The characteristic function of the set of 'properties that Mitt has'.**

Note:
Given the result in (18d), we also have it that:

- $[[((\lambda P_4\ (P_4\ \textbf{mitt'}))\ \textbf{smokes'})]]^{M,g}$ = $p([[\textbf{smokes'}]]^{M,g})$ = $p(h)$ = h(Mitt) = 0
- $[[((\lambda P_4\ (P_4\ \textbf{mitt'}))\ \textbf{man'})]]^{M,g}$ = $p([[\textbf{man'}]]^{M,g})$ = $p(i)$ = i(Mitt) = 1

e. $(\lambda P_4\ \forall x_3\ (\ (\textbf{man'}\ x_3) \rightarrow (P_4\ x_3)\ )\ )$

(i) $[[(\lambda P_4\ \forall x_3\ (\ (\textbf{man'}\ x_3) \rightarrow (P_4\ x_3)))]]^{M,g}$ =

(ii) The function *p* with domain $D_{<et>,E}$, range $D_{t,E}$ and for all a $\in D_{<et>,E}$,
p(a) = 1 iff for all a' $\in D_{e,E}$, either *i*(a') = 0 or a(a') = 1 =

(iii) The function *p* with domain $D_{<et>,E}$, range $D_{t,E}$ and for all a $\in D_{<et>,E}$,
p(a) = 1 iff for all a' $\in D_{e,E}$, if *i*(a') = 1 then a(a') = 1

(iv) **The characteristic function of the set of <et>-functions *f* such that if x is 'a man', then f(x) = 1**

(v) **The characteristic function of the set of 'properties every man has'**

Note:
Given the result in (18e), we also have it that:

- $[[((\lambda P_4\ \forall x_3\ (\ (\textbf{man'}\ x_3) \rightarrow (P_4\ x_3)))\ \textbf{smokes'})]]^{M,g}$ = $p([[\textbf{smokes'}]]^{M,g})$ = $p(h)$ = 0
- $[[((\lambda P_4\ \forall x_3\ (\ (\textbf{man'}\ x_3) \rightarrow (P_4\ x_3)))\ \textbf{man'})]]^{M,g}$ = $p([[\textbf{man'}]]^{M,g})$ = $p(i)$ = 1

3. A Key Logical Equivalence

*The model-theoretic semantics introduced above yields a key logical equivalence, one that will be of **much** use to us shortly…*

(19) **Notational Preliminary**
If φ, ψ are meaningful expressions of a TL language *L*, $\varphi \in ME_\tau$ and $v \in VAR_\tau$, then '[φ/*v*]ψ' is the meaningful expression just like ψ except that every free instance of *v* is replaced with φ.

[**barack'**/x_3] ((**man'** x_3) & (**smokes'** x_3)) = ((**man' barack'**) & (**smokes' barack'**))

[**smokes'**/P_4]∀x_3((**man'** x_3) → (P_4 x_3)) = ∀x_3((**man'** x_3) → (**smokes** x_3))

(20) **Key Validity: Lambda Conversion**
Let (λ*v*ψ) and φ be meaningful expressions with no variables in common, and let $\varphi \in ME_\tau$ and $v \in VAR_\tau$. It follows that the meaningful expressions in (a) and (b) are logically equivalent:
a. ((λ*v*ψ) φ) b. [φ/*v*]ψ

(21) **Illustration of Lambda Conversion**
Each of the following pairs of meaningful expressions are logically equivalent.

a. ((λx_3 ((**man'** x_3) & (**smokes'** x_3))) **barack'**) ⇔
((**man' barack'**) & (**smokes' barack'**))

b. ((λP_4 ∀x_3 ((**man'** x_3) → (P_4 x_3))) **smokes'**) ⇔
∀x_3((**man'** x_3) → (**smokes** x_3))

(22) **Informal Proof of Lambda Conversion**
Let $\mathcal{M}$ be a model <E, I> of a TL language and g be a variable assignment based on $\mathcal{M}$.

- If (λ*v*ψ) is such that $v \in VAR_\tau$ and $\psi \in ME_\sigma$, then by (16j), $[[(\lambda v\psi)]]^{M,g}$ is the function *p* with domain $D_{\tau,E}$ and range $D_{\sigma,E}$ such that for all a in $D_{\tau,E}$, $p(a) = [[\psi]]^{M,g(v/a)}$

- If $\varphi \in ME_\tau$, then by (16) it follows that there is an $a \in D_{\tau,E}$ such that $[[\varphi]]^{M,g} = a$.

- By (16c), it follows that $[[((\lambda v\psi)\ \varphi)]]^{M,g} = [[(\lambda v\psi)]]^{M,g}([[\varphi]]^{M,g}) = p(a) = [[\psi]]^{M,g(v/a)}$

- Finally, **it is intuitively the case that** $[[\psi]]^{M,g(v/a)} = [[\ [\varphi/v]\psi\ ]]^{M,g}$
Thus, $\mathbf{[[\ ((\lambda v\psi)\ \varphi)\ ]]^{M,g} = [[\ [\varphi/v]\psi\ ]]^{M,g}}$

(23) a. **Question:**
Why must we assume in (20) that $(\lambda v\psi)$ and φ have no variables in common?

b. **Answer:**
If φ contains a free variable v' which is bound in ψ, then it can sometimes happen that v' is free in $((\lambda v\psi)\ \varphi)$ but bound in $[\varphi/v]\psi$.

Illustration: $((\lambda v_{t,3}\ \exists x_2((Px_2)\ \&\ v_{t,3}))\ (Qx_2))$ *is **not** logically equivalent to:*
$\exists x_2((Px_2)\ \&\ (Qx_2))$

(24) **The Utility of Lambda Conversion**

As the semanticists in the audience are no doubt aware, we will be making much use of lambda conversion when it comes time to translate structures of English to structures of Politics+λ.

- Our translation process will output rather complex formulae of Politics+λ.
- Lambda conversion will allow us to 'convert' those complex formulae into simpler, logically equivalent formulae.
    - These simpler formulae will more transparently represent the 'meanings' that our translation ends up assigning to the expressions of English.

Rough Illustration:
In the rough illustration below, imagine that *h* is our translation function from English to Politics+λ.

(i) h(*every man smokes*) =

(ii) $H_{\text{Merge-S}}$(h(*every man*), h(*smokes*)) =

(iii) $H_{\text{Merge-S}}((\lambda P_4\ \forall x_3\ ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3))), \mathbf{smokes'})$ =

(iv) $((\lambda P_4\ \forall x_3\ ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3)))\ \mathbf{smokes'})$ ⇔

(v) $\forall x_3((\mathbf{man'}\ x_3) \rightarrow (\mathbf{smokes}\ x_3))$

4. The Logical Language Politics+λ: Montagovian Presentation

In (8), we provide a relatively informal, non-Montagovian definition of the language Politics+λ.

- We will now represent Politics+λ as a 'disambiguated language', in the sense of Montague (1974).

- **In the next handout, we will develop a Montagovian interpretation for Politics+λ, based upon this definition…**

(25) **The Vocabulary of Politics+λ**

a. The Logical Constants:
   (i) *Sentence Connectives:* ~, &, v, →
   (ii) *Quantifiers:* ∀, ∃
   (iii) *Lambda Operator:* λ

b. The Syntactic Symbols: (,)

c. The Non-Logical Constants:
   (i) *Constants:*
   CON_e = { **mitt', barack', michelle'** }
   $CON_{<et>}$ = { **smokes', man', president'** }
   $CON_{<e<e,t>>}$ = { **loves'** }
   For all other $\tau \in T$, $CON_\tau = \varnothing$

   (ii) *Variables:*
   For every type $\tau \in T$, a **countably infinite** set of variables of type τ:
   $VAR_\tau = \{ v_{\tau, n} : n \in \mathbb{N} \}$

(26) **The Syntactic Operations of Politics+λ**

a. $F_{Concat}(\alpha, \beta)$ = $(\alpha\ \beta)$
b. $F_{Not}(\alpha)$ = $\sim\alpha$
c. $F_{And}(\alpha, \beta)$ = $(\alpha\ \&\ \beta)$
d. $F_{Or}(\alpha, \beta)$ = $(\alpha \vee \beta)$
e. $F_{If}(\alpha, \beta)$ = $(\alpha \rightarrow \beta)$
f. $F_{\exists}(\alpha, \beta)$ = $\exists\alpha\ \beta$
g. $F_{\forall}(\alpha, \beta)$ = $\forall\alpha\ \beta$
h. $F_{\lambda}(\alpha, \beta)$ = $(\lambda\alpha\ \beta)$

(27) **The Syntactic Algebra of Politics+λ**

$\langle A, F_\gamma \rangle_{\gamma \in \{Concat, Not, And, Or, If, \exists, \forall, \lambda\}}$ is the algebra such that:

(i) $\{ F_\gamma \}_{\gamma \in \{Concat, Not, And, Or, If, \exists, \forall, \lambda\}}$ are as defined in (26)

(ii) A is the smallest set such that the following holds:

1. For all $\tau \in T$, $CON_\tau \subseteq A$ and $VAR_\tau \subseteq A$
2. A is closed under $\{ F_\gamma \}_{\gamma \in \{Concat, Not, And, Or, If, \exists, \forall, \lambda\}}$

Note:
Thus, the set A contains all the constants and variables in (25), and is closed under the syntactic operations in (26).

(28) **The Syntactic Category Labels of Politics+λ**

Let T be the set of types. $\Delta = T \cup \{ \langle var,\tau \rangle : \tau \in T \}$

Note:
Thus, the syntactic category labels include (i) all the types, and (ii) for every type $\tau \in T$, the 'variable type' $\langle var, \tau \rangle$.

(29) **The Basic Expressions of Politics+λ**

For every type $\tau \in T$:

a. $X_{\langle var,\tau \rangle} = VAR_\tau$

b. $X_\tau = CON_\tau \cup X_{\langle var,\tau \rangle}$

Note: Thus, it follows that:

- $X_{\langle var,e \rangle} = \{ v_{e,n} : n \in \mathbb{N} \} = \{ x_n : n \in \mathbb{N} \}$
- $X_e = \{$ **mitt', barack', michelle'** $\} \cup \{ x_n : n \in \mathbb{N} \}$
- $X_{\langle var,\langle e,t \rangle\rangle} = \{ v_{\langle e,t \rangle, n} : n \in \mathbb{N} \} = \{ P_n : n \in \mathbb{N} \}$
- $X_{\langle e,t \rangle} = \{$ **smokes', man', president'** $\} \cup \{ P_n : n \in \mathbb{N} \}$
- $X_{\langle var,\langle e,\langle e,t \rangle\rangle\rangle} = \{ v_{\langle e,\langle e,t \rangle\rangle, n} : n \in \mathbb{N} \}$
- $X_{\langle e,\langle e,t \rangle\rangle} = \{$ **loves'** $\} \cup \{ v_{\langle e,\langle e,t \rangle\rangle, n} : n \in \mathbb{N} \}$
- For all other types $\tau \in T$, $X_{\langle var,\tau \rangle} = VAR_\tau = \{ v_{\tau,n} : n \in \mathbb{N} \} = X_\tau$

(30) **The Syntactic Rules of Politics+λ**

The (countably infinite) set S consists of (a) and (b) below:

a. The following triples:
(i) $\langle F_{Not}, \langle t \rangle, t \rangle$
(ii) $\langle F_{And}, \langle t,t \rangle, t \rangle$
(iii) $\langle F_{Or}, \langle t,t \rangle, t \rangle$
(vi) $\langle F_{If}, \langle t,t \rangle, t \rangle$

b. For every $\sigma, \tau \in T$, a triple of the following form:
(i) $\langle F_{Concat}, \langle \langle \sigma, \tau \rangle, \sigma \rangle, \tau \rangle$
(ii) $\langle F_{\exists}, \langle \langle var,\sigma \rangle, t \rangle, t \rangle$
(iii) $\langle F_{\forall}, \langle \langle var,\sigma \rangle, t \rangle, t \rangle$
(iv) $\langle F_{\lambda}, \langle \langle var,\sigma \rangle, \tau \rangle, \langle \sigma, \tau \rangle \rangle$

Note: Rules (30bii), (30biii), and (30biv) can be informally read as the following, each of which mirrors an informal syntactic rule in (7d,e).

- "Applying $F_{\exists}$ to a variable of category $\langle var,\sigma \rangle$ and an expression of category t forms an expression of category t."
- "Applying $F_{\forall}$ to a variable of category $\langle var,\sigma \rangle$ and an expression of category t forms an expression of category t."
- "Applying F_{λ} to a variable of category $\langle var,\sigma \rangle$ and an expression of category τ forms an expression of category $\langle \sigma, \tau \rangle$."

(31) **The Disambiguated Language Politics+λ**

Politics+λ is the structure $\langle A, F_{\gamma}, X_{\delta}, S, t \rangle_{\gamma \in \{Concat, Not, And, Or, If, \exists, \forall, \lambda\}, \delta \in \Delta}$ such that A, $\{ F_{\gamma} \}_{\gamma \in \{Concat, Not, And, Or, If, \exists, \forall, \lambda\}}$, $\{ X_{\delta} \}_{\delta \in \Delta}$, S, Δ are all as defined in (25)-(30)

Note: Politics+λ is a disambiguated language.

We will now illustrate the definitions above by examining some meaningful expressions of (31), with associated calculations.

(32) **Illustrative ME:** ~ ((**loves' mitt'**) x_3) $\in C_t$

(i)	**loves'** $\in X_{<e<e,t>>}$, **mitt'**, $x_3 \in X_e$	(by (29))
(ii)	**loves'** $\in C_{<e<e,t>>}$, **mitt'**, $x_3 \in C_e$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(iii)	$< F_{Concat}$, < <e,<e,t>>, e >, <e,t> > $\in$ S	(by (30))
(iv)	F_{Concat}(**loves'**, **mitt'**) $\in C_{<e,t>}$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(v)	(**loves' mitt'**) $\in C_{<e,t>}$	(by def. of F_{Concat})
(vi)	$< F_{Concat}$, < <e,t>, e >, t > $\in$ S	(by (30))
(vii)	F_{Concat}((**loves' mitt'**), x_3) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(viii)	((**loves' mitt'**) x_3) $\in C_t$	(by def. of F_{Concat})
(ix)	$<F_{Not}$, <t>, t> $\in$ S	(by (30))
(x)	F_{Not}(((**loves' mitt'**) x_3)) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(xi)	~ ((**loves' mitt'**) x_3) $\in C_t$	(by def. of F_{Not})

(33) **Illustrative ME:** $\forall x_3$ ((**smokes'** x_3) → ~ ((**loves' mitt'**) x_3)) $\in C_t$

(i)	**smokes'** $\in X_{<e,t>}$, $x_3 \in X_e$, $x_3 \in X_{<var,e>}$	(by (29))
(ii)	**smokes'** $\in C_{<e,t>}$, $x_3 \in C_e$, $x_3 \in C_{<var,e>}$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(iii)	$< F_{Concat}$, < <e,t>, e >, t > $\in$ S	(by (30))
(iv)	F_{Concat}(**smokes'**, x_3) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(v)	(**smokes'** x_3) $\in C_t$	(by def. of F_{Concat})
(vi)	~ ((**loves' mitt'**) x_3) $\in C_t$	(by (32))
(vii)	$<F_{If}$, <t,t>, t> $\in$ S	(by (30))
(viii)	F_{If}((**smokes'** x_3), ~((**loves' mitt'**) x_3)) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(ix)	((**smokes'** x_3) → ~ ((**loves' mitt'**) x_3)) $\in C_t$	(by def. of F_{If})
(x)	$< F_\forall$, < <var,e>, t >, t > $\in$ S	(by (30))
(xi)	$F_\forall$ (x_3, ((**smokes'** x_3) → ~((**loves' mitt'**) x_3)) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(xii)	$\forall x_3$ ((**smokes'** x_3) → ~ ((**loves' mitt'**) x_3)) $\in C_t$	(by def. of $F_\forall$)

(34) **Illustrative ME:** $\exists P_4$ (P_4 **michelle'**) $\in C_t$

(i)	**michelle'** $\in X_e$, $P_4 \in X_{<e,t>}$, $P_4 \in X_{<var,<e,t>>}$	(by (29))
(ii)	**michelle'** $\in C_e$, $P_4 \in C_{<e,t>}$, $P_4 \in C_{<var,<e,t>>}$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(iii)	$< F_{Concat}$, < <e,t>, e >, t > $\in$ S	(by (30))
(iv)	F_{Concat}(P_4, **michelle'**) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(v)	(P_4 **michelle'**) $\in C_t$	(by def. of F_{Concat})
(vi)	$< F_\exists$, < <var,<e,t>>, t >, t > $\in$ S	(by (30))
(vii)	$F_\exists$ (P_4, (P_4 **michelle'**)) $\in C_t$	(by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(viii)	$\exists P_4$ (P_4 **michelle'**) $\in C_t$	(by def. of $F_\exists$)

(35) **Illustrative ME:** $((\lambda x_3\ (\mathbf{man'}\ x_3))\ \mathbf{mitt'}) \in C_t$

(i) $\mathbf{man'} \in X_{\langle e,t\rangle}$, $\mathbf{mitt'}, x_3 \in X_e$, $x_3 \in X_{\langle var,e\rangle}$ (by (29))
(ii) $\mathbf{man'} \in C_{\langle e,t\rangle}$, $\mathbf{mitt'}, x_3 \in C_e$, $x_3 \in C_{\langle var,e\rangle}$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(iii) $\langle F_{Concat}, \langle \langle e,t\rangle, e\rangle, t\rangle \in S$ (by (30))
(iv) $F_{Concat}(\mathbf{man'}, x_3) \in C_t$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(v) $(\mathbf{man'}\ x_3) \in C_t$ (by def. of F_{Concat})
(vi) $\langle F_\lambda, \langle \langle var,e\rangle, t\rangle, \langle e, t\rangle\rangle \in S$ (by (30))
(vii) $F_\lambda(x_3, (\mathbf{man'}\ x_3)) \in C_{\langle e,t\rangle}$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(viii) $(\lambda x_3\ (\mathbf{man'}\ x_3)) \in C_{\langle e,t\rangle}$ (by def. of F_λ)
(ix) $F_{Concat}((\lambda x_3\ (\mathbf{man'}\ x_3)), \mathbf{mitt'}) \in C_t$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(x) $((\lambda x_3\ (\mathbf{man'}\ x_3))\ \mathbf{mitt'}) \in C_t$ (by def. of F_{Concat})

(36) **Illustrative ME:** $(\lambda P_4\ \forall x_3\ ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3))) \in C_{\langle\langle e,t\rangle,t\rangle}$

(i) $\mathbf{man'} \in X_{\langle e,t\rangle}$, $x_3 \in X_e$, $X_{\langle var,e\rangle}$, $P_4 \in X_{\langle e,t\rangle}$, $X_{\langle var,\langle e,t\rangle\rangle}$ (by (29))
(ii) $\mathbf{man'} \in C_{\langle e,t\rangle}$, $x_3 \in C_e$, $C_{\langle var,e\rangle}$, $P_4 \in C_{\langle e,t\rangle}$, $C_{\langle var,\langle e,t\rangle\rangle}$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(iii) $\langle F_{Concat}, \langle \langle e,t\rangle, e\rangle, t\rangle \in S$ (by (30))
(iv) $F_{Concat}(\mathbf{man'}, x_3) \in C_t$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(v) $(\mathbf{man'}\ x_3) \in C_t$ (by def. of F_{Concat})
(vi) $F_{Concat}(P_4, x_3) \in C_t$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(vii) $(P_4\ x_3) \in C_t$ (by def. of F_{Concat})
(viii) $\langle F_{If}, \langle t,t\rangle, t\rangle \in S$ (by (30))
(viii) $F_{If}((\mathbf{man'}\ x_3), (P_4\ x_3)) \in C_t$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(ix) $((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3)) \in C_t$ (by def. of F_{If})
(x) $\langle F_\forall, \langle \langle var,e\rangle, t\rangle, t\rangle \in S$ (by (30))
(xi) $F_\forall\ (x_3, ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3))) \in C_t$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(xii) $\forall x_3\ ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3)) \in C_t$ (by def. of $F_\forall$)
(xiii) $\langle F_\lambda, \langle \langle var,\langle e,t\rangle\rangle, t\rangle, \langle\langle e,t\rangle,t\rangle\rangle \in S$ (by (30))
(xiv) $F_\lambda\ (P_4, \forall x_3\ ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3))) \in C_{\langle\langle e,t\rangle,t\rangle}$ (by def. of $\{C_\delta\}_{\delta \in \Delta}$)
(xv) $(\lambda P_4\ \forall x_3\ ((\mathbf{man'}\ x_3) \rightarrow (P_4\ x_3))) \in C_{\langle\langle e,t\rangle,t\rangle}$ (by def. of F_λ)