First Steps Towards PTQ: A New Presentation of Our System for Quantifiers

(1) **From UG to PTQ**

- By far and away, the most cited paper by Montague is "The Proper Treatment of Quantification in Ordinary English" (PTQ).

- When compared to "Universal Grammar", PTQ is a relatively *informal* presentation of Montague's key ideas.
    - It was written as a paper for the 1970 Stanford Workshop on Grammar and Semantics.

- Consequently, PTQ doesn't hold to letter of the UG system and its notations, and introduces certain simplifications…
    - Many aspects of Montague's analysis in PTQ can be easily rephrased in the strictly algebraic UG framework
    - **Others, as we will see, cannot (without sacrificing some elegance)**

In these notes, we will take the analysis of English quantification developed in the last few handouts, and 'convert' it into a format closer to what is found in PTQ…

- Many of the changes are simply superficial ones of terminology and notation…

---

1. The Fragment 'Mini-English+Q': PTQ-Style Presentation

(2) **Redefinition: Category**
In PTQ, the indices that we've been referring to as 'category labels' (Δ in UG) are instead dubbed simply 'categories'. The set of 'categories' is represented as *Cat* ($\sim\Delta$ in UG)
- This more closely approximates the terminology in generative linguistics.

(3) **The Categories of ME+Q**: *Cat* = {TV, IV, S, T, CN, PR}

(4) **New Notation: Basic Expressions**
In PTQ, B_A is the set of basic expressions of category A. ($\sim X_\delta$ in UG)

(5) **The Basic Expressions of ME+Q**
B_{TV} = { **loves** }
B_{IV} = { **smokes** }
B_T = { **Barack, Mitt, Michelle** }
B_{CN} = { **man, president**, **woman** }
B_{PR} = { **he *n*** : $n \in \mathbb{N}$ } $\cup$ { **she *n*** : $n \in \mathbb{N}$ }
B_S = $\varnothing$

(6) **Remarks**
- Our English expressions are now boldfaced rather than italicized.
- Following PTQ (rather than UG), pronouns are now primitive expressions, rather than ones syntactically derived from indices.
- To set up something interesting for later, we've also added **woman** to our set of CNs.
- **Note that our English expressions are now *strings* again, rather than trees. As we'll see, this won't pose any problems for the PTQ 'version' of MG….**

(7) **New Terminology: Phrases**
In PTQ, the sets that we've been referring to as (e.g.) 'the category δ' (C_δ in UG) are instead dubbed 'the phrases of category δ'.
- P_A = The phrases of category A

(8) **The Syntactic Rules**
As in UG, the set $\cup_{A \in Cat} P_A$ ($\sim \cup_{\delta \in \Delta} C_\delta$ in UG) will be defined via the *syntactic rules*.
- In PTQ, however, a 'syntactic rule' is a rather different object from that in UG.
- As shown below, in PTQ, the 'rules' incorporate *both* (i) the definition of the syntactic operations in the language, and (ii) the information contained in a UG 'rule'.

(9) **The Rule S1**

a. The Rule: $B_A \subseteq P_A$, for every category A.

b. Remarks:
- This is clearly not a 'syntactic rule', in the sense found in UG.
- What S1 does is move to the system of 'syntactic rules' a condition that was (in UG) part of the general definition of the set of categories ($\cup_{\delta \in \Delta} C_\delta$)

(10) **The Rule S2**

a. The Rule:
If $\zeta \in P_{CN}$, then $F_0(\zeta)$, $F_2(\zeta) \in P_T$, where $F_0(\zeta)$ = **every** ζ and $F_2(\zeta)$ = **some** ζ

b. Remarks:
- Again, this is not a 'syntactic rule', in the sense found in UG.
- What S2 does is combine together the information we had earlier factored out into (i) the definitions of K_{Every} and K_{Some}, and (ii) the rules <K_{Every}, <CN>, T> and <K_{Some}, <CN>, T>

Note:

- Following Montague in PTQ (and UG), the indices on our rules will no longer be evocative mnemonics. Rather, they will simply be numerals.
- I will also be numbering both **rules** and **operations** in a way that matches the corresponding rules and operations in PTQ (thus, F_0 for **every** and F_2 for **some**).

(11) **The Rule S4**

a. The Rule: If $\alpha \in P_{PR}$ and $\delta \in P_{IV}$, then $F_4(\alpha,\delta) \in P_S$, where $F_4(\alpha,\delta) = \alpha\ \delta$

b. Remarks:
Again, as with S2 and the rules below, this 'syntactic rule' combines together the information that in the UG system was factored out into:
- The definition of $K_{\text{Merge-S}}$, and
- The rule < $K_{\text{Merge-S}}$, < PR, IV >, S >

(12) **The Rule S5**

a. The Rule:
If $\delta \in P_{TV}$ and $\beta \in P_{PR}$, then $F_5(\delta,\beta) \in P_{IV}$, where $F_5(\delta,\beta) = \delta$ **him *n*** if β has the form **he *n*** , $F_5(\delta,\beta) = \delta$ **her *n*** if β has the form **she *n***, and $F_5(\delta,\beta) = \delta\ \beta$ otherwise.

b. Remarks: Again, this rule combines together the following information:
- The definition of $K_{\text{Merge-IV}}$
- The rule < $K_{\text{Merge-IV}}$, < TV, PR >, IV >

(13) **The Rule S11**

a. The Rule:
If $\varphi, \psi \in P_S$, then $F_8(\varphi, \psi), F_9(\varphi, \psi) \in P_S$, where $F_8(\varphi,\psi) = \varphi$ **and** ψ, and $F_9(\varphi,\psi) = \varphi$ **or** ψ.

b. Remarks:

- Again, we see how this one 'syntactic rule' contains both (i) the definition of the syntactic operations F_8 ($\sim K_{\text{And}}$) and F_9 ($\sim K_{\text{Or}}$), and (ii) the UG-syntactic rules references those operations.
- I will follow Montague in PTQ by only having these rules and operations; I will henceforth drop the operations K_{If}, K_{Not}, and the rules involving them.
  - **After all, they give us <u>horrible</u> analyses of conditionals and negation in English ;)**

(14) **The Rule S14**

a. <u>The Rule:</u>
If $\alpha \in P_T$ and $\varphi \in P_S$, then $F_{10,n}(\alpha,\varphi) \in P_S$, where $F_{10,n}(\alpha,\varphi)$ comes from φ by replacing the first occurrence of **he *n*** , **him *n*** , **she *n*** , or **her *n***, by α.

b. <u>Remarks:</u>
- Again, this one 'syntactic rule' combines together both:
  (i) The definition of the operations $F_{10,n}(\alpha,\varphi)$ ($\sim K_{QI}(<n,\varnothing>,\alpha,\varphi)$)
  (ii) The UG-style rule $< F_{10,n} , <T, S>, S>$

- **Note that this rule doesn't appeal to a *single* syntactic operation, but a whole infinite family of them, $\{ F_{10,n} : n \in \mathbb{N} \}$**

With the rules above, we are able to offer the following definition of the set of phrases for ME+Q

(15) **The Phrases of Mini-English+Q**
$\{ P_A \}_{A \in Cat}$ is the smallest family of sets indexed by *Cat* such that S1-S14 are true.

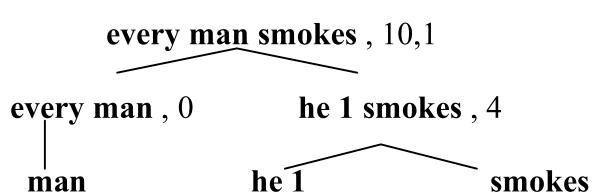
(16) **The Meaningful Expressions of English**
φ is a meaningful expression of ME+Q if there is an $A \in Cat$ such that $\varphi \in P_A$.

(17) **Some Illustrative Members of P_S**

a. **Every man smokes**

(i)	**man** $\in B_{CN}$, **smokes** $\in B_{IV}$, **he 1** $\in B_{PR}$	(by (5))
(ii)	**man** $\in P_{CN}$, **smokes** $\in P_{IV}$, **he 1** $\in P_{PR}$	(by S1)
(iii)	**he 1 smokes** $\in P_S$	(by S4)
(iv)	**every man** $\in P_T$	(by S2)
(v)	**every man smokes** $\in P_S$	(by S14)

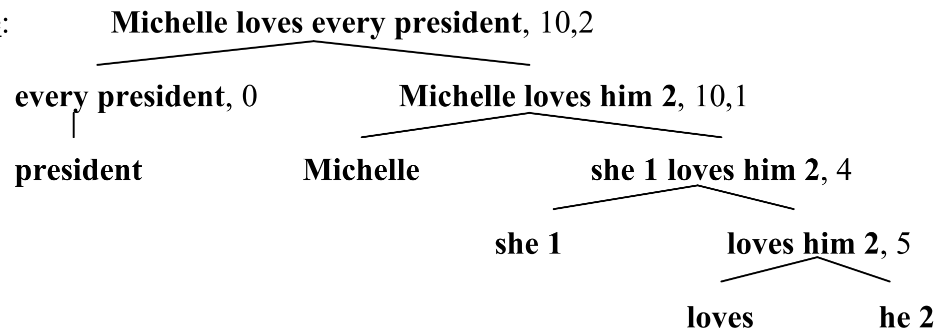
<u>Analysis Tree:</u>



b. **Michelle loves every president**

(i)	**Michelle** $\in B_T$, **president** $\in B_{CN}$, **loves** $\in B_{TV}$, **she 1, he 2** $\in B_{PR}$	(5)
(ii)	**Michelle** $\in P_T$, **president** $\in P_{CN}$, **loves** $\in P_{TV}$, **she 1, he 2** $\in P_{PR}$	(S1)
(iii)	**loves him 2** $\in P_{IV}$	(S5)
(iv)	**she 1 loves him 2** $\in P_S$	(S4)
(v)	**Michelle loves him 2** $\in P_S$	(S14)
(vi)	**Michelle loves every president** $\in P_S$	(S14)

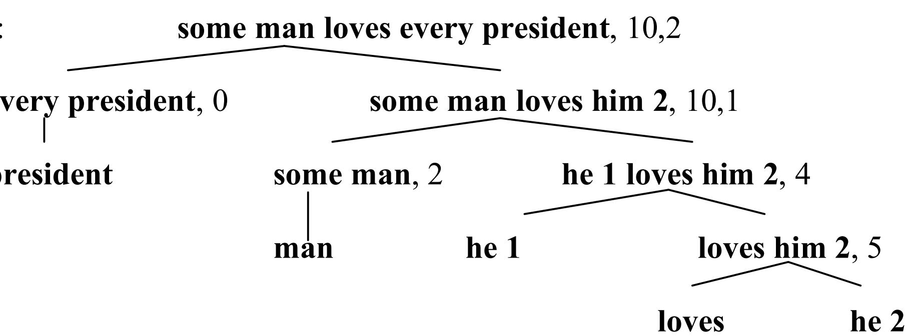
Analysis Tree:



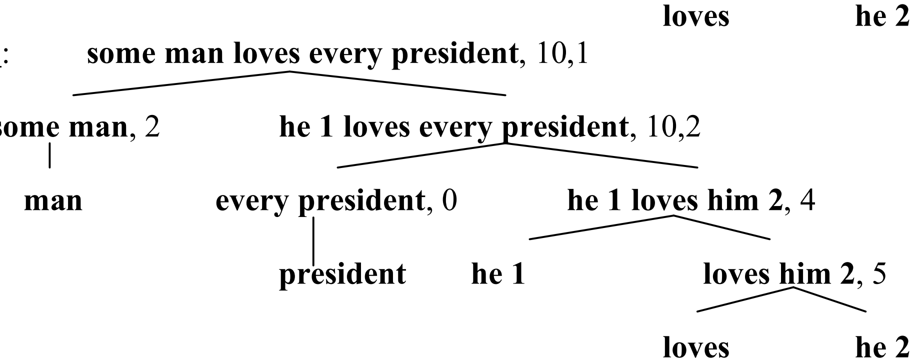
c. **Some man loves every president.**

(i)	**man, president** $\in B_{CN}$, **loves** $\in B_{TV}$, **he 1, he 2** $\in B_{PR}$	(5)
(ii)	**man, president** $\in P_{CN}$, **loves** $\in P_{TV}$, **he 1, he 2** $\in P_{PR}$	(S1)
(iii)	**loves him 2** $\in P_{IV}$	(S5)
(iv)	**he 1 loves him 2** $\in P_S$	(S4)
(v)	**some man**, **every president** $\in P_T$	(S2)
(vi)	**some man loves him 2** $\in P_S$ / **he 1 loves every president** $\in P_S$	(S14)
(vii)	**some man loves every president** $\in P_S$	(S14)

Analysis Tree 1:



Analysis Tree 2:



(18) **Key Observation**
As shown in (17c), there are meaningful expressions of ME+Q that are genuinely syntactically ambiguous.

- ME+Q as defined above would not be a 'disambiguated language' in the sense of UG.

(19) **Quote by Montague**
"Thus our fragment admits genuinely (that is, semantically) ambiguous sentences. If it were desired to construct a corresponding unambiguous language, it would be convenient to take the analysis trees themselves as the expressions of that language; it would then be obvious how to characterize…the structural operations of that language and the correspondence relation between its expressions and those of ordinary English. For present purposes, however, no such construction is necessary." (Montague 1974; p. 23).

No such construction is necessary?...
As we will see shortly, translation in PTQ is simply a *relation*, and need not be a function

2. The Logical Language TL: PTQ-Style Presentation

(20) **Logical Languages in PTQ: First Key Difference**

- In PTQ, Montague does not represent the logical translation language as a 'disambiguated language' in the sense of UG.
  - o Rather, he gives a (relatively) simple, recursive definition of its syntax.

- In PTQ, Montague does not represent the semantics of the logical translation language as a '(Fregean) interpretation' in the sense of UG.
  - o Rather, he gives a (relatively) simple, model-theoretic semantics.

- **However, given the equivalences between these systems for syntax and semantics, it is possible (though laborious) to convert between them (see UG)**

  - o **Thus, there is no substantive difference in the theory of the syntax/semantics of the logical translation language (the presentation in PTQ is just 'simpler')**

(21) **Logical Languages in PTQ: The Second Key Difference**

- Up to now, we've been translating our natural language into a 'tailor-made' logical language with a finite set of constants.

- **Technically speaking, in PTQ (and UG), Montague doesn't do that, but rather lays out a logical language with an infinite set of constants**
  - o That is, in PTQ and UG, **man'** isn't the *translation* of **man**, but rather a meta-language abbreviation for "whatever constant *c* is the translation of **man**"

In what follows, we define the syntax and semantics of a <u>*single*</u> *language, dubbed 'Typed Logic'.*

(22) **The Vocabulary of Typed Logic (TL)**

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

b. <u>The Syntactic Symbols:</u> (,)

c. <u>The Non-Logical Constants:</u>
(i) *Constants:*
For every type $\tau \in T$, a **countably infinite** set of constants of type τ:
$CON_\tau = \{ c_{\tau, n} : n \in \mathbb{N} \}$

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

(23) **Meta-Language Abbreviations for Variables and Constants**
Although our variables 'officially' all look like those in (22c), to save space we will make use of the following meta-language abbreviations:

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

c. $a_n = c_{e,n}$ d. $Q_n = c_{<e,t>,n}$ e. $R_n = c_{<e, <e,t>>,n}$

(23) **The Syntax of 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 $(\lambda v\ \varphi) \in ME_{<\sigma,\tau>}$

(24) **Some Illustrative Meaningful Expressions of TL**

a. $\sim ((R_3\ a_2)\ a_1)$ [~ ((**loves' mitt'**) **barack'**) in Politics+λ]

b. $\forall x_3\ (\ (Q_2\ x_3) \rightarrow \sim ((R_3\ a_2)\ x_3))$ [$\forall x_3$((**smokes'** x_3) → ~ ((**loves' mitt'**) x_3))]

c. $\exists P_4\ (P_4\ a_3)$ [$\exists P_4$ (P_4 **michelle'**) in in Politics+λ]

d. $(\ (\lambda x_3\ (Q_3\ x_3))\ a_2\ )$ [((λx_3 (**man'** x_3)) **mitt'**) in Politics+λ]

e. $(\lambda P_4\ \forall x_3\ (\ (Q_3\ x_3) \rightarrow (P_4\ x_3)\ )\ )$ [($\lambda P_4\ \forall x_3$ ((**man'** x_3) → ($P_4\ x_3$))), Politics+λ]

Our semantics for TL is going to exactly follow our model-theoretic semantics for Politics+λ

(25) **The Semantics of TL, Part 1: The Denotations Based on a Set E**
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 $D_{\tau,E}$

(26) **The Semantics of TL, Part 2: A Model for TL**
A model $\mathcal{M}$ for TL is a pair <E, I> consisting of:
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 $I(\alpha) \in D_{\tau,E}$

(27) **The Semantics of TL, Part 3: Variable Assignments**
Let $\mathcal{M}$ be a model <E, I> of TL. 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 $g(\alpha) \in D_{\tau,E}$

(28) **The Semantics of TL, Part 4: Interpretation w.r.t. Model and Variable Assignment**

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

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

b. If $\alpha \in \cup_{\tau \in T} \text{CON}_\tau$, 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 $\mathbf{a} \in \mathbf{D_{\tau,E}}$ 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 $\mathbf{a} \in \mathbf{D_{\tau,E}}$, $[[\psi]]^{M,g(v/a)} = 1$

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

(29) **Illustration of the Semantics: A Model for TL**

Let the model $\mathcal{M}$ be the pair <{Barack, Michelle, Mitt}, I>, where I contains the following mappings (amongst infinitely many others):

a. $I(a_1)$ = Barack
b. $I(a_2)$ = Mitt
c. $I(a_3)$ = Michelle

d. $I(Q_2)$ = *h* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }
e. $I(Q_3)$ = *i* = { <Michelle, 0>, <Barack,1>, <Mitt,1> }
f. $I(Q_1)$ = *k* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }

e. $f(R_3)$ = *j* =

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

(30) **Illustration of the Semantics: Interpretation of Illustrative Expressions**
Let $\mathcal{M}$ be the model defined in (29). Let g be some arbitrary variable assignment based on $\mathcal{M}$.

a. $[[(\lambda P_4\ (P_4\ a_2))]]^{M,g}$ =
The function *p* with domain $D_{<et>,E}$, range $D_{t,E}$ and for all $a \in D_{<et>,E}$,
p(a) = a(Mitt)

b. $[[(\lambda P_4\ \forall x_3\ ((Q_3\ x_3) \rightarrow (P_4\ x_3)))]]^{M,g}$ =
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 =

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 =

The characteristic function of the set of 'properties every man has'

---

3. The Translation from ME+Q to TL: PTQ-Style Presentation

In PTQ, the system for translating from ME+Q to TL differs slightly from that in UG.

- As we will see, however, the differences are not that deep or fundamental…

(31) **First Ingredient: Category-to-Type Mapping**
Just like with the UG notion of a translation base, one of the key ingredients to the PTQ translation system is a mapping from *Cat* to the set of types (in our logical language).

f(TV)	=	<e, <e,t>>	f(IV)	=	<e,t>
f(S)	=	t	f(T)	=	<<e,t>,t>
f(CN)	=	<e,t>	f(PR)	=	e

Note: The mapping above is *not at all* the one that actually appears in PTQ.
Recall that we're right now just 'converting' our system from the last handout into the PTQ style.

(32) **Second Ingredient: Lexical Translation Function**
Again, just as in UG's notion of a translation base, the second key ingredient to the PTQ translation system is a function mapping the basic expressions of ME+Q to ones in TL.

Let g be a function whose domain is the set of basic expressions in (5) **except for B_T and B_{PR},** and for all A $\in$ *Cat*, $\alpha \in B_A$, and $\alpha \in$ Domain(g), g(α) $\in CON_{f(A)}$

- That is, g maps expressions of category A to *constants* of the corresponding type
- This restriction that g only maps to *constants* is unique to PTQ

(33) **Meta-Language Abbreviations**

Recall that the constants of our language TL are all of the form $c_{\tau,\, n}$. In what follows, we'll make use of the following meta-language abbreviations.

a. **loves'** = g(**loves**) [whatever <e,<e,t>> constant g maps **loves** to]
b. **smokes'** = g(**smokes**) [whatever <e,t> constant g maps **smokes** to]
c. **man'** = g(**man**) [whatever <e,t> constant g maps **man** to]
d. **president'** = g(**president**) [whatever <e,t> constant g maps **president** to]
e. **woman'** = g(**woman**) [whatever <e,t> constant g maps **woman** to]

(34) **Third Ingredient: Translation Rules**

In the PTQ-system, the work done in UG by the translation base and the definition of the polynomial operations is instead done by a system of 'translation rules'.

- These 'translation rules' stand in a one-to-one correspondence with the syntactic rules in (9)-(14)
  - o Much as how in UG the polynomial operations are in correspondence with the syntactic operations, and the derived syntactic rules with the syntactic rules.

- As we'll see, these 'translation rules' do the work of both (i) defining the 'polynomial operation' each syntactic operation corresponds to; (ii) putting the operations in correspondence.

(35) **The Rule T1**

a. The Rule: Rule T1 consists of a collection of sub-rules.

(i) If α is in the domain of g, then α translates to g(α).

(ii) **Barack** translates to (λP_4 (P_4 **barack'**)), where **barack**' $\in CON_e$
Mitt translates to (λP_4 (P_4 **mitt'**)), where **mitt**' $\in CON_e$
Michelle translates to (λP_4 (P_4 **michelle'**)), where **michelle**' $\in CON_e$

(iii) **he n** and **she n** translate to: x_n (= $v_{e,n}$)

b. Remarks:
- This 'translation rule' incorporates the following information:
  (i) The lexical translation function *g* is a subset of the full translation relation
  (ii) Proper names translate as <<e,t>,t> formula; pronouns as type-e variables.

- **In the UG system, the information in (ii)-(iii) is part of the lexical translation function; in PTQ, though, that function can only map to constants (and so we need to pack it in as a separate rule)**

(36) **The Rule T2**

a. The Rule: If $\zeta \in P_{CN}$ and ζ translates to ζ', then
(i) $F_0(\zeta)$ translates to $(\lambda P_0\ \forall x_0\ ((\zeta'\ x_0) \rightarrow (P_0\ x_0)))$
(ii) $F_2(\zeta)$ translates to $(\lambda P_0\ \exists x_0\ ((\zeta'\ x_0)\ \&\ (P_0\ x_0)))$,

b. Remarks: As mentioned in (34), rule T2 does both the following:
(i) Defines the 'polynomial operations' corresponding to F_0 (= H_{Every}) and F_2 (= H_{Some})
(ii) Puts these polynomial operations 'in correspondence' with F_0 and F_2 in the translation relation.

(37) **The Rule T4**

a. The Rule:
If $\delta \in P_{PR}$ and $\beta \in P_{IV}$, and δ, β translate to δ', β' respectively, then $F_4(\delta,\beta)$ translates to $(\beta'\ \delta')$

b. Remarks: Again, as mentioned in (34), rule T4 does both the following:
(i) Defines the 'polynomial operation' corresponding to F_4 (= $H_{\text{Merge-S}}$)
(ii) Puts that polynomial operation in correspondence with F_4 in the translation relation.

(38) **The Rule T5**
If $\delta \in P_{TV}$ and $\beta \in P_{PR}$, and δ, β translate to δ', β' respectively, then $F_5(\delta,\beta)$ translates to $(\delta'\ \beta')$.

(39) **The Rule T11**
If $\varphi,\psi \in P_S$, and φ, ψ translate to φ', ψ' respectively, then $F_8(\varphi,\psi)$ translates to $(\varphi'\ \&\ \psi')$ and $F_9(\varphi,\psi)$ translates to $(\varphi' \vee \psi')$.

(40) **The Rule T14**

a. The Rule:
If $\alpha \in P_T$ and $\varphi \in P_S$, and α, φ translate to α', φ' respectively, then $F_{10,n}(\alpha,\varphi)$ translates to $(\ \alpha'\ (\lambda x_n\ \varphi')\ )$

b. Remarks:

- As in the corresponding syntactic rule S14, this translation rule covers not simply *one* syntactic operation, but a whole infinite family of them.

- This translation rule implicitly does the work of defining an infinite family of polynomial operations $H_{10,n}$, each corresponding to $F_{10,n}$

(41) **The Translation Relation**
The translation relation translates to between expressions of ME+Q and those of TL is the smallest binary relation satisfying T1-T14.

Given the correspondence between the syntactic rules S1-S14 and the translation rules T1-T14, we can build up the translation for a sentence (rule-by-rule) as we construct it.

(42) **Illustration of the Translation Rules, Part 1**

(i) **man**, **loves**, **president** translate to **man'**, **loves'**, **president'** respectively ((33), T1)

(ii) **he 1** and **he 2** translate to x_1 and x_2 respectively (T1)

(iii) F_5(**loves**, **he 2**) translates to (**loves'** x_2) (T5)

(iv) **loves him 2** translates to (**loves'** x_2) (def. of F_5)

(v) F_4(**he 1**, **loves him 2**) translates to ((**loves'** x_2) x_1) (T4)

(vi) **he 1 loves him 2** translates to ((**loves'** x_2) x_1) (def. of F_4)

(vii) F_2(**man**) translates to $(\lambda P_0\ \exists x_0\ ((\textbf{man'}\ x_0)\ \&\ (P_0\ x_0)))$ (T2)

(viii) **some man** translates to $(\lambda P_0\ \exists x_0\ ((\textbf{man'}\ x_0)\ \&\ (P_0\ x_0)))$ (def. of F_2)

(ix) $F_{10,1}$(**some man**, **he 1 loves him 2**) translates to
$((\lambda P_0\ \exists x_0\ ((\textbf{man'}\ x_0)\ \&\ (P_0\ x_0)))\ (\lambda x_1\ ((\textbf{loves'}\ x_2)\ x_1)))$ (T14)

(x) **some man loves him 2** translates to
$((\lambda P_0\ \exists x_0\ ((\textbf{man'}\ x_0)\ \&\ (P_0\ x_0)))\ (\lambda x_1\ ((\textbf{loves'}\ x_2)\ x_1)))$ (def. of $F_{10,1}$)

(xi) F_0(**president**) translates to $(\lambda P_0\ \forall x_0\ ((\textbf{president'}\ x_0) \rightarrow (P_0\ x_0)))$ (T2)

(xii) **every president** translates to $(\lambda P_0\ \forall x_0\ ((\textbf{president'}\ x_0) \rightarrow (P_0\ x_0)))$ (def. of F_0)

(xiii) $F_{10,2}$(**every president**, **some man loves him 2**) translates to
$((\lambda P_0\ \forall x_0\ ((\textbf{president'}\ x_0) \rightarrow (P_0\ x_0)))$
$(\lambda x_2\ ((\lambda P_0\ \exists x_0\ ((\textbf{man'}\ x_0)\ \&\ (P_0\ x_0)))\ (\lambda x_1\ ((\textbf{loves'}\ x_2)\ x_1)))))$ (T14)

(xiv) **some man loves every president** translates to
$((\lambda P_0\ \forall x_0\ ((\textbf{president'}\ x_0) \rightarrow (P_0\ x_0)))$
$(\lambda x_2\ ((\lambda P_0\ \exists x_0\ ((\textbf{man'}\ x_0)\ \&\ (P_0\ x_0)))\ (\lambda x_1\ ((\textbf{loves'}\ x_2)\ x_1)))))$ $\Leftrightarrow$ (α- and λ-conv.)

(xv) $\forall x_0\ ((\textbf{president'}\ x_0) \rightarrow \exists x_2\ ((\textbf{man'}\ x_2)\ \&\ ((\textbf{loves'}\ x_0)\ x_2)))$

(43) **Illustration of the Translation Rules, Part 2**
Steps (i)-(vi) are exactly the same as those in (42):

(vii) F_0(**president**) translates to $(\lambda P_0\ \forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow (P_0\ x_0)))$ (T2)

(viii) **every president** translates to $(\lambda P_0\ \forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow (P_0\ x_0)))$ (def. of F_0)

(ix) $F_{10,2}$(**every president**, **he 1 loves him 2**) translates to
$((\lambda P_0\ \forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow (P_0\ x_0)))\ (\lambda x_2\ ((\mathbf{loves'}\ x_2)\ x_1)))$ (T14)

(x) **he 1 loves every president** translates to
$((\lambda P_0\ \forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow (P_0\ x_0)))\ (\lambda x_2\ ((\mathbf{loves'}\ x_2)\ x_1)))$ (def. of $F_{10,2}$)

(xi) F_2(**man**) translates to $(\lambda P_0\ \exists x_0\ ((\mathbf{man'}\ x_0)\ \&\ (P_0\ x_0)))$ (T2)

(xii) **some man** translates to $(\lambda P_0\ \exists x_0\ ((\mathbf{man'}\ x_0)\ \&\ (P_0\ x_0)))$ (def. of F_2)

(xiii) $F_{10,1}$(**some man**, **he 1 loves every president**) translates to:
$((\lambda P_0\ \exists x_0\ ((\mathbf{man'}\ x_0)\ \&\ (P_0\ x_0)))$
$(\lambda x_1\ ((\lambda P_0\ \forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow (P_0\ x_0)))\ (\lambda x_2\ ((\mathbf{loves'}\ x_2)\ x_1)))))$ (T14)

(xiv) **some man loves every president** translates to:
$((\lambda P_0\ \exists x_0\ ((\mathbf{man'}\ x_0)\ \&\ (P_0\ x_0)))$
$(\lambda x_1\ ((\lambda P_0\ \forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow (P_0\ x_0)))\ (\lambda x_2\ ((\mathbf{loves'}\ x_2)\ x_1))))) \Leftrightarrow$ (α- and λ-conv.)

(xv) $\exists x_0\ ((\mathbf{man'}\ x_0)\ \&\ \forall x_2\ ((\mathbf{president'}\ x_2) \rightarrow ((\mathbf{loves'}\ x_2)\ x_0)))$

(44) **Key Observation**
As shown by (42) and (43), the relation 'translates to' is ***not*** a function.

- When our syntactic derivation of **some man loves every president** follows the procedure in Analysis Tree 1 in (17c), the translation is logically equivalent to:
  $\forall x_0\ ((\mathbf{president'}\ x_0) \rightarrow \exists x_2\ ((\mathbf{man'}\ x_2)\ \&\ ((\mathbf{loves'}\ x_0)\ x_2)))$

- When our syntactic derivation of **some man loves every president** follows the procedure in Analysis Tree 2 in (17c), the translation is logically equivalent to:
  $\exists x_0\ ((\mathbf{man'}\ x_0)\ \&\ \forall x_2\ ((\mathbf{president'}\ x_2) \rightarrow ((\mathbf{loves'}\ x_2)\ x_0)))$

That is, syntactically ambiguous strings in ME+Q can be paired with *more than one translation*, corresponding to the different ways the strings can be derived in the syntax.

- Of course, if the language were *not* syntactically ambiguous, then the translation relation would be a function….

4. Introducing 'Meaning Postulates'

(45) **A Potential Issue for Our System**
Our semantics for TL set up in Section 2 was quite general and permissive. Consequently, there are models for TL that would not be appropriate as (induced) interpretations of ME+Q.

Illustration:
- It is perfectly consistent with the definitions in Section 2 for a model of TL to map the constants **man'** and **woman'** to the same, or overlapping <et>-functions.
- However, it is (maybe) part of the grammar of English that these terms are antonyms.

(46) **Relevant Quote from Montague**
"The interpretations of intensional logic may, by way of the translation relation, be made to play a second role as interpretations of English. Not all interpretations of intensional logic, however, would be reasonable candidates for interpretations of English. In particular, it would be reasonable in this context to restrict attention to those interpretations of intensional logic in which the following formulas are true…" (Montague 1974; p. 27)

(47) **The Solution to Issue (45)**
A 'logically permissible model of TL' is one in which formulae (47a,b) are true:
a. $\forall x_0$ ((**man'** x_0) → ~(**woman'** x_0))
b. $\forall x_0$ ((**woman'** x_0) → ~(**man'** x_0))

In our analysis of English, we only consider 'logically permissible models of TL'.

(48) **On 'Meaning Postulates'**

- The term 'meaning postulate' is often used (though not by Montague himself) to refer to such conditions on models for a language.

- Though the example in (47) is dubious,[1] it's possible to imagine more plausible cases:
  - $\forall x_0$ ((**bachelor'** x_0) → (**man'** x_0))
  - $\forall x_0$ ((**wine'** x_0) → $\exists x_1$ ((**grape'** x_1) & ((**made-from'** x_1) x_0)))

- In general, such 'meaning postulates' allow us to encode facts about lexical semantics into our overall analysis of the natural language.
  - Whether this is a good *theory* of lexical semantics, though, is up for debate…

[1] After all, one could argue that is simply a *contingent biological fact* that no men are women, rather than it being a part of the *lexical semantics* of 'man' and 'woman'.

5. Direct and Indirect Interpretation in the PTQ-Style System

As should be obvious by now, given our model in (29), our translation system from Section 3 allows us to 'indirectly interpret' strings of ME+Q.

(49) **The Interpretation of ME+Q Relative to a Model and Variable Assignment**

Let φ be a meaningful expression of ME+Q, and let φ translate to φ'. Let $\mathcal{M}$ be a (logically permissible) model for TL and g be a variable assignment based on $\mathcal{M}$. We say that $[[\varphi']]^{M, g}$ is the **interpretation of φ under translation φ'** relative to $\mathcal{M}$ and g.

Illustration:

- The interpretation of **some man loves every president** under translation (42) and relative to the model in (29) is 1
  - o After all, Barack is the only president, and he loves himself.
- The interpretation of **some man loves every president** under translation (43) and relative to the model in (29) is also 1
  - o Again, this is rendered true by the fact that Barack loves himself.

(50) **The Equivalence of Indirect and Direct Interpretation in UG**
Once a translation base **T** is is specified for **L** and **L'**, if **L'** has an interpretation **B'**, it is trivial (mechanical) to specify an interpretation **B** for **L.**

(51) a. Question: Is indirect interpretation similarly 'eliminable' in a PTQ-style system?

b. Answer:
Yes, but it is a little less trivial/mechanical to build the model $\mathcal{M}$ for natural language **L** on the basis of model $\mathcal{M}'$ for logical language **L'** and the translation.

(52) **A General Method for Recasting Indirect Interpretation as Direct Interpretation**
For each 'translation rule' T_n , specify a new 'interpretation rule' I_n , where the output of I_n is the model-theoretic interpretation of T_n

(53) **Direct Interpretation of ME+Q, Step 1: Defining the Models**
A model $\mathcal{M}$ for ME+Q is a pair <E, I> consisting of (i) a *non-empty* set E, called the 'domain of $\mathcal{M}$', and (ii) a function I, whose domain is (a) and whose range satisfies the condition in (b).

(a) *Domain of I:* $B_{TV} \cup B_{IV} \cup B_{CN}$

(b) *Condition on Range of I:*
1. If $\varphi \in B_{TV}$, then $I(\varphi) \in D_{<e,<e,t>>, E}$
2. If $\varphi \in B_{IV}$, then $I(\varphi) \in D_{<e,t>, E}$
3. If $\varphi \in B_{CN}$, then $I(\varphi) \in D_{<e,t>, E}$

(54) **Illustration: A Model for ME+Q**
Let the model $\mathcal{M}$ be the pair <{Barack, Michelle, Mitt}, I>, where I contains the following mappings (amongst infinitely many others):

a. I(**smokes**) = *h* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }
b. I(**man**) = *i* = { <Michelle, 0>, <Barack,1>, <Mitt,1> }
c. I(**president**) = *k* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }
d. I(**woman**) = *l* = { <Michelle, 1>, <Barack,0>, <Mitt,0> }

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

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

(55) **Direct Interpretation of ME+Q, Step 2: Defining Variable Assignment**
Let $\mathcal{M}$ be a model <E, I> for ME+Q. 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:* $\mathbb{N}$

(iv) *Condition on Range of g:* For all $n \in \mathbb{N}$, $g(n) \in E$

We now must lay out a definition for 'interpretation relative to a model and a variable assignment'…

Notice how the definition below mirrors our translation rules T1-T14

(56) **Direct Interpretation of ME+Q, Step 3: Interpretation w.r.t. Model and Assignment**
Let $\mathcal{M}$ be a model <E, I> for ME+Q and g be a variable assignment based on $\mathcal{M}$. The relation 'X is an interpretation of φ relative to $\mathcal{M}$ and g' $[[.]]^{M,g}$ is defined as follows:

a. Rule I1:
(i) If φ is in the domain of I, then $[[\varphi]]^{M,g} = I(\varphi)$

(ii) $[[\textbf{Barack}]]^{M,g}$ =
the function *p* with domain $D_{<e,t>,E}$ such that for all $f \in D_{<e,t>,E}$
$p(f) = 1$ iff $f(\beta) = 1$ (where β is some specified member of E)

$[[\textbf{Mitt}]]^{M,g}$ =
the function *p* with domain $D_{<e,t>,E}$ such that for all $f \in D_{<e,t>,E}$
$p(f) = 1$ iff $f(\mu) = 1$ (where μ is some specified member of E)

$[[\textbf{Michelle}]]^{M,g}$ =
the function *p* with domain $D_{<e,t>,E}$ such that for all $f \in D_{<e,t>,E}$
$p(f) = 1$ iff $f(\nu) = 1$ (where ν is some specified member of E)

(iii) $[[\textbf{he } \boldsymbol{n}]]^{M,g} = [[\textbf{she } \boldsymbol{n}]]^{M,g} = g(n)$

b. Rule I2:
(i) If $\varphi \in P_{CN}$, then $[[F_0(\varphi)]]^{M,g}$ =
the function *p* with domain $D_{<e,t>,E}$ such that for all $f \in D_{<e,t>,E}$
$p(f) = 1$ iff for all $x \in E$, if $[[\varphi]]^{M,g}(x) = 1$, then $f(x) = 1$

(ii) If $\varphi \in P_{CN}$, then $[[F_2(\varphi)]]^{M,g}$ =
the function *p* with domain $D_{<e,t>,E}$ such that for all $f \in D_{<e,t>,E}$
$p(f) = 1$ iff there is an $x \in E$ such that $[[\varphi]]^{M,g}(x) = 1$ and $f(x) = 1$

c. Rule I4: If $\delta \in P_{PR}$ and $\beta \in P_{IV}$, then $[[F_4(\delta, \beta)]]^{M,g} = [[\beta]]^{M,g}([[\delta]]^{M,g})$

d. Rule I5: If $\delta \in P_{TV}$ and $\beta \in P_{PR}$, then $[[F_5(\delta, \beta)]]^{M,g} = [[\delta]]^{M,g}([[\beta]]^{M,g})$

e. Rule I11: If $\varphi, \psi \in P_S$, then $[[F_8(\varphi,\psi)]]^{M,g} = 1$ *iff* $[[\varphi]]^{M,g} = [[\psi]]^{M,g} = 1$, and $[[F_9(\varphi,\psi)]]^{M,g} = 1$ *iff* $[[\varphi]]^{M,g} = 1$ or $[[\psi]]^{M,g} = 1$

f. Rule I14:
If $\alpha \in P_T$ and $\varphi \in P_S$, then $[[F_{10,n}(\alpha,\varphi)]]^{M,g}$ =
$[[\alpha]]^{M,g}$(the function *p* with domain E such that for all $x \in E$, $p(x) = [[\varphi]]^{M,g(n/x)}$)

We can now use the definitions in (53)-(55) to directly interpret expressions of ME+Q!!

(57) **Illustration: Directly Interpreting a Sentence of ME+Q**
Let $\mathcal{M}$ be the model defined in (54) and g be any variable assignment based on $\mathcal{M}$.

(i) [[**every man smokes**]]^{M,g} = (by definition of ME+Q)

(ii) [[$F_{10,1}$ (F_0(**man**), F_4(**he 1**, **smokes**))]]^{M,g} = (by I14)

(iii) [[F_0(**man**)]]^{M,g}(the function *p* with domain E such that for all x ∈ E,
p(x) = [[F_4(**he 1**, **smokes**)]]$^{M,g(1/x)}$) = (by I4)

(iv) [[F_0(**man**)]]^{M,g}(the function *p* with domain E such that for all x ∈ E,
p(x) = [[**smokes**)]]$^{M,g(n1x)}$([[**he 1**]]$^{M,g(1/x)}$)) = (by I1)

(v) [[F_0(**man**)]]^{M,g}(the function *p* with domain E such that for all x ∈ E,
p(x) = I(**smokes**)(g(1/x)(1)) = (by (54))

(vi) [[F_0(**man**)]]^{M,g}(the function *p* with domain E such that for all x ∈ E,
p(x) = *h*(x) = (by meta-logical reasoning)

(vii) [[F_0(**man**)]]^{M,g}(*h*) = (by I2)

(viii) (the function *p* with domain $D_{<e,t>,E}$ such that for all *f* ∈ $D_{<e,t>,E}$
p(*f*) = 1 iff for all x ∈ E, if [[**man**]]^{M,g}(x)= 1, then *f*(x) = 1)(*h*) = (by I1)

(ix) (the function *p* with domain $D_{<e,t>,E}$ such that for all *f* ∈ $D_{<e,t>,E}$
p(*f*) = 1 iff for all x ∈ E, if *i*(x)= 1, then *f*(x) = 1)(*h*) = (by meta-logical reasoning)

(x) 1 *iff* for all x ∈ E, if *i*(x)= 1, then *h*(x) = 1 = (by (54))

(xi) 0

(58) **Remark**
- Note that for syntactically ambiguous strings, there can be two possible interpretations w.r.t a model and variable assignment.
- Thus, as the language in (56) suggests, [[.]]^{M,g} is not a function.

(59) **What We've Done So Far**
- Reviewed the UG architecture, and shown how it can provide a compositional semantics for a fragment of English with quantificational 'terms' (NPs)
- Reviewed the PTQ architectures, and shown how it also can provide a compositional semantics for a fragment of English with quantificational 'terms' (NPs)
- Examined how the systems in UG and PTQ relate to one another…

Now, let's actually study Montague's full semantic analysis of English in PTQ…