The Proper Treatment of Quantification in Ordinary English, Part 3: The Translation System, Part 1 [1]

In these notes, we will begin to explore Montague's system for mapping expressions in our fragment of English to formulae of Intensional Logic.

1. The Category-to-Type Mapping

The first key ingredient of a PTQ-style translation system (and a UG-style translation base) is a function mapping the categories of the natural language to the categories of the logical language (*i.e.,*, the types).

- *It is at this first step that the semantic analysis developed in PTQ immediately gets rather complicated.*

- *Understanding this mapping and why it is this way is key to understanding everything that follows…*

(1) **The Core Issue: Opaque Environments in English**

- Recall from the handout "*The Fragment of English*" that our fragment of English contains the following basic expressions: **seek**, **allegedly**, **about**

- Recall also that these expressions create an 'opaque' (non-extensional) environment.
  - "John seeks a unicorn" doesn't entail "there is a unicorn."
  - "John talked about a unicorn" doesn't entail "there is a unicorn."
  - "John allegedly danced" doesn't entail "John danced."

- Consequently, the meanings (extensions) of these lexical items ([[X]]) must combine with the intension of their complements
  - [[**seek**]] takes the intension of **a unicorn** as argument
  - [[**about**]] takes the intension of **a unicorn** as argument
  - [[**allegedly**]] takes the intension of **dance** as argument.

- Thus, [[**seek**]] is not of type <e,<e,t>>, and so in our indirect semantics, the translation of **seek** cannot be of type <e,<e,t>>
  - Similarly, the translation of **about** cannot be of type <e,<e,t>>
  - Similarly, the translation of **allegedly** cannot be of type <<e,t>,<e,t>>

(2) **Burning Question:** What should the types of [[**seek**]], [[**about**]], [[**allegedly**]] be?

Before we develop an answer to this question in (2), let us notice two additional important issues

[1] These notes are based upon material in Dowty *et al.* (1981) Chapter 7.

(3) **Issue 1: Higher, Intensional Types for *Everything***

- Since **seek** is a TV, and its translation is *not* of type <e,<e,t>>, then – due to the need for category-to-type correspondence in the translation – **no TV can have a translation of type <e,<e,t>>**
  - Therefore, even the translations of **find**, **eat**, and **love** will be of the same high, intensional type as that of **seek**
- Similarly, since **allegedly** is an IAV, it follows that **no IAV (not even 'rapidly') can have a translation of type <<e,t>,<e,t>>**
  - Therefore, the translation of **rapidly** must be of the same, high, intensional type as **allegedly**
- Similarly, since **about** is an IAV/T (preposition), it follows that **no IAV/T (not even 'in') can have a translation of type <e,<e,t>>**
  - Therefore, the translation of **in** must also be of the same, high, intnsional type as **about**

(4) **Immediate Problem: Failure to Predict Key Inferences**

If the translations (and induced interpretations) of **eat**, **rapidly**, and **in** are of the same high, intensional types as **seek**, **allegedly**, and **about**, our semantics will fail to predict the validity of the following inferences:

a. "John ate a unicorn" *does* entail "There is a unicorn."
b. "John talked in a house" *does* entail "There is a house."
c. "John rapidly danced" *does* entail "John danced."

(5) **The Solution (Preview)**

As we'll see later, we can solve the problem in (4) by adding 'meaning postulates' for **eat**, **rapidly**, **in** that guarantee the inferences in (4).

- Thus, by adding these meaning postulates, we guarantee that these lexical items behave *as if* they received a purely extensional semantics (even though they don't).

(6) **Issue 2: All 'Function Application' is *Intensional* Function Application**

- Given (3), all TVs will have an (induced) meaning that takes the *intension* of a term as its argument:
    - Both **seek** and **eat** take the *intension* of **a unicorn** as argument.

- Given (3), all IAVs will have an (induced) meaning that takes the *intension* of an IV as its argument.
    - Both **allegedly** and **rapidly** take the *intension* of **dance** as argument.

- Given (3), all IAV/Ts will have an (induced) meaning that takes the *intension* of a term as its argument.
    - Both **about** and **in** take the *intension* of **a unicorn** as argument.

- Also, note that the (induced) meaning of IV/t verbs like **believe that** take as argument the *intension* of their sentential complements.

- Similarly, the (induced) meaning of IV//IV verbs like **try to** take as argument the *intentsion* of their IV complements

The General Pattern Emerging Here:
The translation (induced meaning) of a predicative expression in English always takes as argument the *intension* of the translation (induced meaning) of its syntactic argument.

(7) **Immediate Consequence: Higher Types for the Translations of Terms**

- Recall that we view Ts (= t/IV) as being predicates of IVs

- Thus, the general pattern in (6) entails that, in sentence (7a) below, the translation of a T like **John** should take as argument the *intension* of the translation of **smokes**

a. **John smokes.**

- Thus, the translation of **John** should be something like $[\lambda v_{0,<s,<e,t>>}\ v_{0,<s,<e,t>>}\{j\}]$

Rough Illustration:

(i)	The translation of **John smokes**	=	(by (6))
(ii)	The translation of **John** taking as argument the intension of the translation of **smokes**.	=	(by assumption, and notation)
(iii)	$[\lambda v_{0,<s,<e,t>>}\ v_{0,<s,<e,t>>}\{j\}]([$^**smokes'**$])$	⇔	(by λ-conversion)
(iv)	[^**smokes'**]$\{j\}$	=	(by Curly Bracket Notation)
(v)	$[^{\vee}$[^**smokes'**]](j)	⇔	(by Down-Up Cancellation)
(vi)	**smokes'**(j)		

All these observations taken together lead us (and Montague) to the following general view of the category-to-type correspondence in PTQ…

(8) **Montague's Category-to-Type Correspondence in PTQ**
The function f has *Cat* as its domain and is such that:

a.	f(e)	=	e		
b.	f(t)	=	t		
c.	f(A/B) =		f(A//B)	=	< < s, f(B) >, f(A) >

(9) **Remarks**
Thus, a predicative expression in English (*i.e.*, one of category A/B or A//B) will always get a translation that takes as argument an *intensional* expression (type <s,f(B)>)

- Thus, if φ in English combines syntactically with an expression of category B (*i.e.,* φ is of category A/B)…
    - **Then the translation of φ will combine syntactically with the *intension* of the translation of an expression of category B (i.e., <s, f(B)>)**

- Thus, we immediately capture the higher intensional types of the translations of IVs, IAVs, IAV/Ts, IV/ts, and IV//IVs

(10) **A Possible Snag**
As beautiful as (8) is, it entails the following category-to-type correspondences.

a.	f(CN)	=	f(t//e)	=	< <s,e> , t >
b.	f(IV)	=	f(t/e)	=	< <s,e> , t >
c.	f(T)	=	f(t/(t/e))	=	< < s, < <s,e>, t > >, t >

- Thus, (8) predicts that the translation of **walk** will be a function of type <<s,e>,t>
    - Thus, the translation (meaning) of **walk** takes *individual concepts* as argument

- Consequently, terms end up translated to expressions of *extremely* high type

(11) **But, Is This Really a 'Snag'**

a. <u>A 'Pro' of the Correspondences in (10):</u>
By treating all IVs as (indirectly) denoting <<s,e>,t> functions, Montague can analyze such puzzling sentences as "The temperature is 90 and rising."

b. <u>'Cons' of the Correspondences in (10):</u>
- As detailed by Dowty *et al.* (1981), there may be other, superior analyses of "The temperature is 90 and rising."
- The resulting translations for (relatively simple) expressions of English get rather complicated.

(12) **A Simpler Category-to-Type Corespondence**
As discovered by Bennett (1976) and detailed by Dowty *et al.* (1981), the following category-to-type mapping necessitates very few changes to the overall PTQ system.

The function f has *Cat* as its domain and is such that:

a. f(e) = e
b. f(t) = t
c. f(CN) = f(IV) = <e,t>
d. For all other categories A/B, f(A/B) = f(A//B) = < <s, f(B)>, f(A) >

(13) **Category-to-Type Correspondences, Under (12)**

Category	Corresponding Type	Set-Theoretic Object
t (sentences)	t	truth-value
CN	<e,t>	function from entities to truth-values
IV	<e,t>	function from entities to truth-values
T (= t/IV)	< <s,<e,t>>,t>	function from properties to truth-values[2]
IV/t	< <s,t>, <e,t> >	function from propositions to <e,t>-functions
IV//IV	< <s, <e,t>>, <e,t>>	function from properties to <e,t>-functions
IAV (= IV/IV)	< <s, <e,t>>, <e,t>>	function from properties to <e,t>-functions
TV (= IV/T)	**< <s, < <s,<e,t>>,t>>, <e,t>>**	**function from the intensions of GQs to an <e,t>-function.**

I will follow Bennett (1976) and Dowty et al. (1981) in adopting the category-to-type correspondences in (12)-(13).

- Again, these will not necessitate any serious changes from what's in the original article…
- The benefit is that the system ends up being relatively simpler…

2. The 'Lexical' Translation Function

The second main ingredient of a PTQ-style translation system (and the third main ingredient of a UG-style translation base) is a function translating the basic expressions of the natural language.

- Again, in PTQ, the range of this function can only be *constants* of the logical language.

- Consequently, this function in PTQ will <u>*not*</u> have the following basic expressions in its range (since we want them translated as either (i) variables, or (ii) complex expressions)

  - B_T The basic terms of English, {**John, Mary, Bill, ninety, $\mathbf{he_0}$, $\mathbf{he_1}$, $\mathbf{he_2}$, …**}
  - **necessarily**
  - **be**

---

[2] Such functions, I will (somewhat misleadingly) refer to as 'GQs'

(14) **The 'Lexical' Translation Function**
Let g be a function such that:

a. The domain of g is the set of basic expressions of our fragment of English other than **be**, **necessarily**, and the members of B_T

b. Whenever $A \in Cat$, $\alpha \in B_A$, and α is in the domain of g, $g(\alpha) \in Con_{f(A)}$

(15) **A Picture of the 'Lexical' Translation Function**

a. The IVs: g(**run**), g(**walk**), g(**talk**), g(**rise**), g(**change**) $\in Con_{<e,t>}$

b. The CNs: g(**man**), g(**woman**), g(**park**), g(**fish**), g(**pen**), g(**unicorn**), g(**price**), g(**temperature**) $\in Con_{<e,t>}$

c. The IV/t's: g(**believe that**), g(**assert that**) $\in Con_{<<s,t>,<e,t>>}$

d. The IV//IVs: g(**try to**), g(**wish to**) $\in Con_{<<s, <e,t>>,<e,t>>}$

e. The IAVs: g(**rapidly**), g(**slowly**), g(**voluntarily**), g(**allegedly**) $\in Con_{<<s, <e,t>>,<e,t>>}$

f. The TVs: g(**find**), g(**lose**), g(**eat**), g(**love**), g(**date**), g(**seek**), g(**conceive**) $\in Con_{< <s, < <s,<e,t>>,t>>, <e,t>>}$

g. The IAV/Ts: g(**in**), g(**about**) $\in Con_{< <s, < <s,<e,t>>,t>>, <<s,<e,t>>,<e,t>>>}$

(16) **Some Useful Meta-Linguistic Abbreviations**

- Recall that Montague never specifies *what* the constants of IL look like.
- Nevertheless, it will be very useful to have some elegant, compact way of referring to the translations of various basic expressions of English.
- Montague therefore introduces the following convention for forming *meta-linguistic* labels / abbreviations for various translations

a. <u>The Convention:</u> If α is in the domain of g, then $\alpha' = g(\alpha)$

b. <u>Illustration:</u>
(i) **run'** = g(**run**) [i.e., whatever $Con_{<e,t>}$ g maps **run** to]
(ii) **man'** = g(**man**) [i.e., whatever $Con_{<e,t>}$ g maps **man** to]
(iii) **find'** = g(**find**) [i.e., whatever $Con_{< <s, < <s,<e,t>>,t>>, <e,t>>}$ g maps **find** to]
(iv) **believe that'** =
g(**believe that**) [i.e., whatever $Con_{<<s,t>,<e,t>>}$ g maps **believe that** to]

(17) **Some Further Useful Meta-Linguistic Abbreviations**

a. In our translation rule for proper names, we will want to make reference to certain specified members of Con_e Montague introduces *j, m, b, n* as meta-linguistic labels for such constants.

- Again, the letters '*j*', '*m*', '*b*', and '*n*' aren't (necessarily) constants.
- Rather, *j, m, b, n* are simply labels *we* are using to refer to such constants.

b. In our translation rules, we will also want to use certain specific variables. Therefore, to save space, Montague introduces the following abbreviations:

(i) x, y, x_n *abbreviates* $v_{1,e}$, $v_{3,e}$ $v_{2n,e}$ (entity variable)

(ii) p *abbreviates* $v_{0,<s,t>}$ (proposition variable)

(iii) P,Q *abbreviate* $v_{0,<s,<e,t>>}$, $v_{1,<s,<e,t>>}$ (property variable)

(iv) $\mathcal{P}$ *abbreviates* $v_{0,<s, <<s,<e,t>>,t>>}$ (variable over intensions of GQs)

With all these ingredients in place, we are now ready to lay out the translation rules of PTQ!

3. Some Translation Rules of PTQ

We will begin with the first translation rule, which governs the basic expressions of the fragment

(18) **Rule T1 (For Basic Expressions)**

a. If α is in the domain of *g*, then α translates into *g*(α)

Illustration:
run translates to **run'** (= *g*(**run**))
man translates to **man'** (= *g*(**man**))
find translates to **find'** (= *g*(**find**))
believes that translates to **believes that'** (= *g*(**believes that**))

b. **be** translates into $\lambda\mathcal{P}\lambda x\ \mathcal{P}\{\,[{}^\wedge\lambda y\,[{}^\vee x = {}^\vee y]]\,\}$

Note: Don't worry about this for now; we may discuss it later.

c. **necessarily** translates into $\lambda p\ [\ \Box\ {}^\vee p\ ]$

Note: Again, don't worry about this one for now.

d. (i) **John** translates into λP[P{*j*}]
(ii) **Mary** translates into λP[P{*m*}]
(iii) **Bill** translates into λP[P{*b*}]
(iv) **ninety** translates into λP[P{*n*}]

Note:
- In PTQ itself, the actual translation of (e.g.) **John** is λP[P{^*j*}]
- This is because in PTQ, Montague held IVs like **walk** to be of type <<s,e>,t>
- Since we're assuming the simpler category-to-type mapping in (12), we can use the simpler translations in (18d) above.

e. **$\mathbf{he_n}$** translates into λP[P{x_n}]

Thus, translation rule T1 will cover the translations of all the basic expressions in our fragment.

- From this point on, our discussion of the translation rules will not follow the order of the rules in PTQ.
- Rather, as with Dowty *et al.* (1981), we will discuss the rules in a more 'pedagogically oriented' order.
- We'll thus next turn directly for the translation rule handling subject-predicate structures.

(19) **Rule T4 (Subject-Predicate Translation Rule)**

If $\delta \in P_{t/IV}$, $\beta \in P_{IV}$, and δ, β translate into δ', β' respectively, then F_4(δ, β) translates into δ'([^β'])

Note: Just as promised in (7), the translation of a term δ will take as argument the *intension* of the translation of an IV β.

(20) **Illustration of Rule T4**

a. *Obtaining the Translation*
(i) **John** translates into λP[P{*j*}], **run** translates into **run'** (T1)
(ii) F_4(**John**, **run**) translates into [λP[P{*j*}]](^**run'**) (T4)
(iii) **John runs** translates into [λP[P{*j*}]](^**run'**) (def. of F_4)

b. Simplifying the Translation
(i) [λP[P{*j*}]](^**run'**) ⇔ (λ-Conv.)
(ii) [^**run'**]{*j*} ⇔ (CBN)
(iii) [$^\vee$[^**run'**]](*j*) ⇔ (DUC)
(iv) **run'**(*j*)

(21) **Remarks**

- Thus, the translation of **John runs** is logically equivalent to **run'**(*j*)

- Thus, under our induced semantics for English, we predict that relative to a model $\mathcal{M}$, a world i, a time j, and a variable assignment g, **John runs** will be 1 *iff*

  - The extension of **run'** w.r.t. $\mathcal{M}$, i, j, g maps the extension of *j* w.r.t. $\mathcal{M}$, i, j, g to 1
  - (Informally speaking) 'John is running' in world i at time j (in model $\mathcal{M}$)

- **Thus, we find that our induced semantics correctly predicts the truth-conditions of present tense sentences.**
  - Their truth at a particular world/time depends upon the extensions of the predicates at that world/time

(22) **An Important Addition: Meaning Postulates for Translations of Names**

- Note that, in order to *really* get the semantics of **John runs** right, we're going to want the extension of the constant *j* to have the same extension in all possible worlds/times.

- That is, in the translations of **John**, **Mary**, **Bill**, and **ninety** in (18d), we're going to want *j, m, b, n* to behave as 'rigid designators'.

- We can ensure this if we add the 'meaning postulate' below:

Meaning Postulate for Names
In a 'logically possible' interpretation for IL, the following formulae are true (at all worlds and times):

$\mathrm{V}x \,\Box[x = \alpha]$, where α is *j*, *m*, *b*, or *n*
Informally: 'There is a (single) entity x s.t. in all possible worlds/times x = *j*'
'There is a (single) entity x s.t. in all possible worlds/times x = *m*'
'There is a (single) entity x s.t. in all possible worlds/times x = *b*'
'There is a (single) entity x s.t. in all possible worlds/times x = *n*'

- Thus, the truth of the formula above (at worlds/times) ensures that the constant *j* denotes the same entity in all possible worlds (and the same holds for *m, b, n*)

(23) **Remark**
Since we'll only ever be considering 'logically possible' interpretations (models) for IL, we can now always regard the constants *j*, *m*, *b*, *n* as having constant intensions in IL.

- **Thus, we can now freely 'λ-convert' them into the scope of ^, $\Box$, W, H**

Rule T4 handles the translation/interpretation of (positive) present-tense sentences;
Rule T17 handles the translation/interpretation of negative and past/future tense sentences…

(24) **Rule T17 (Translation Rule for Negative, Perfect, Future Sentences)**
If $\alpha \in P_T$, $\delta \in P_{IV}$, and α, δ translate into α', δ' respectively, then

(i)	$F_{11}(\alpha, \delta)$	translates into	$\neg \alpha'([\hat{}\delta'])$
(ii)	$F_{12}(\alpha, \delta)$	translates into	$W\ \alpha'([\hat{}\delta'])$
(iii)	$F_{13}(\alpha, \delta)$	translates into	$\neg W\ \alpha'([\hat{}\delta'])$
(iv)	$F_{14}(\alpha, \delta)$	translates into	$H\ \alpha'([\hat{}\delta'])$
(v)	$F_{14}(\alpha, \delta)$	translates into	$\neg H\ \alpha'([\hat{}\delta'])$

Illustrations:

a.
(i) **John** translates into $\lambda P[P\{j\}]$, **run** translates into **run'** (T1)
(ii) F_{11}(**John**, **run**) translates into $\neg[\lambda P[P\{j\}]](\hat{}\mathbf{run'})$ (T17)
(iii) **John doesn't run** translates into $\neg[\lambda P[P\{j\}]](\hat{}\mathbf{run'})$ (def. of F_{11})
(iv) $\neg[\lambda P[P\{j\}]](\hat{}\mathbf{run'}) \Leftrightarrow \neg\mathbf{run'}(j)$ (λ-Conv., CBN, DUC)

Note:
Thus, the translation of **John doesn't run** is logically equivalent to $\neg\mathbf{run'}(j)$
- Thus, informally speaking, **John doesn't run** will be true at a world/time if it's false that John is running at that world/time

b.
(i) **John** translates into $\lambda P[P\{j\}]$, **run** translates into **run'** (T1)
(ii) F_{12}(**John**, **run**) translates into $W[\lambda P[P\{j\}]](\hat{}\mathbf{run'})$ (T17)
(iii) **John will run** translates into $W[\lambda P[P\{j\}]](\hat{}\mathbf{run'})$ (def. of F_{12})
(iv) $W[\lambda P[P\{j\}]](\hat{}\mathbf{run'}) \Leftrightarrow W\ \mathbf{run'}(j)$ (λ-Conv., CBN, DUC)

Note:
Thus, the translation of **John will run** is logically equivalent to $W\ \mathbf{run'}(j)$
- Thus, informally speaking, **John will run** is true at a world/time if John runs at some future time at that world.

c.
(i) **John** translates into $\lambda P[P\{j\}]$, **run** translates into **run'** (T1)
(ii) F_{15}(**John**, **run**) translates into $\neg H[\lambda P[P\{j\}]](\hat{}\mathbf{run'})$ (T17)
(iii) **John hasn't run** translates into $\neg H\ [\lambda P[P\{j\}]](\hat{}\mathbf{run'})$ (def. of F_{15})
(iv) $\neg H\ [\lambda P[P\{j\}]](\hat{}\mathbf{run'}) \Leftrightarrow \neg H\ \mathbf{run'}(j)$ (λ-Conv., CBN, DUC)

Note:
Thus, the translation of **John hasn't run** is logically equivalent to $\neg H\ \mathbf{run'}(j)$
- Thus, informally speaking **John hasn't run** is true at a world/time if there is no previous time at that world where John runs.

Thus far, we've only been illustrating the compositional rules with proper names…
Another easy set of rules to examine, though, are the translation rules for quantificational terms

(25) **Rule T2 (Translation Rule for Quantificational Terms)**

If $\zeta \in P_{CN}$ and ζ translates into ζ', then:

(i) $F_0(\zeta)$ translates into $\lambda P\ \Lambda x[\zeta'(x) \rightarrow P\{x\}]$
(ii) $F_1(\zeta)$ translates into $\lambda P\ Vy\ [\Lambda x\ [\zeta'(x) \leftrightarrow x = y]\ \&\ P\{y\}]$
(iii) $F_2(\zeta)$ translates into $\lambda P\ Vx[\zeta'(x) \wedge P\{x\}]$

Illustrations:

a. (i) **man** translates into **man'** (T1)
(ii) F_0(**man**) translates into λP Λx[**man'**(x) → P{x}] (T2)
(iii) **every man** translates into λP Λx[**man'**(x) → P{x}] (def. of F_0)

b. (i) **man** translates into **man'** (T1)
(ii) F_1(**man**) translates into λP Vy [Λx [**man'**(x) ↔ x = y] & P{y}] (T2)
(iii) **the man** translates into λP Vy [Λx [**man'**(x) ↔ x = y] & P{y}] (def. of F_1)

c. (i) **man** translates into **man'** (T1)
(ii) F_2(**man**) translates into λP Vx[**man'**(x) ∧ P{x}] (T2)
(iii) **a man** translates into λP Vx[**man'**(x) ∧ P{x}] (def. of F_2)

Note: As previewed a few classes ago, Montague adopts a 'Russelian' analysis of definite descriptions like "the man".

(26) **Interactions Between Rules T2, T4, and T17**
With rules T2, T4, and T17, we can now easily translate/interpret sentences where quantificational terms occupy subject position.

a. (i) **a man** translates into λP Vx[**man'**(x) ∧ P{x}] (T2)
(ii) **run** translates into **run'** (T1)
(iii) F_4(**a man**, **run**) translates into [λP Vx[**man'**(x) ∧ P{x}]](^**run'**) (T4)
(iv) **a man runs** translates into [λP Vx[**man'**(x) ∧ P{x}]](^**run'**) (def. of F_4)

(v) [λP Vx[**man'**(x) ∧ P{x}]](^**run'**) ⇔ (λ-Conv.)
(vi) Vx[**man'**(x) ∧ [^**run'**]{x}] ⇔ (CBN)
(vii) Vx[**man'**(x) ∧ [$^{\vee}$[^**run'**]](x)] ⇔ (DUC)
(viii) Vx[**man'**(x) ∧ **run'**(x)]

b. (i) **a man** translates into $\lambda P \bigvee x[\mathbf{man'}(x) \wedge P\{x\}]$ (T2)
(ii) **run** translates into **run'** (T1)
(iii) F_{15}(**a man**, **run**) translates into $\neg H[\lambda P \bigvee x[\mathbf{man'}(x) \wedge P\{x\}]](\hat{}\,\mathbf{run'})$ (T17)
(iv) **a man hasn't run** translates into $\neg H[\lambda P \bigvee x[\mathbf{man'}(x) \wedge P\{x\}]](\hat{}\,\mathbf{run'})$ (def. of F_{15})

(v) $\neg H\,[\lambda P \bigvee x[\mathbf{man'}(x) \wedge P\{x\}]](\hat{}\,\mathbf{run'})$ $\Leftrightarrow$ (λ-Conv., CBN, DUC)
(vi) $\neg H \bigvee x[\mathbf{man'}(x) \wedge \mathbf{run'}(x)]$

(27) **Remarks**

a. Thus, we correctly predict that **a man runs** is true at a world/time *iff* there is a man x (at that world/time) who runs (at that world/time)

b. Thus, we also correctly predict that **a man hasn't run** has a 'reading' (translation/interpretation) that is true at a world/time *iff* there is no earlier time at that world where a man runs…

- Note that this amounts to the claim that there is a reading of **a man hasn't run** that is equivalent to **no man has run**

c. *Of course, there is also a 'wide-scope indefinite' reading of* **a man hasn't run**, where it is true if there is some (particular) man x such that x hasn't run.

- Once we bring QI into the mix, we'll see that our fragment predicts this reading as well!

Finally, we'll also examine the translation rules for conjunction and disjunction, since those are also relatively easy.

(28) **Rule T11 (Conjunction and Disjunction of Sentences)**

If $\varphi, \psi \in P_t$ and φ, ψ translates into φ', ψ' respectively, then
(i) $F_8(\varphi, \psi)$ translates into $[\varphi' \wedge \psi']$
(ii) $F_9(\varphi, \psi)$ translates into $[\varphi' \vee \psi']$

Illustration: The student is asked to confirm that rule T11 entails the following:

a. The translation of **John runs and Mary talks** is logically equivalent to:
$[\mathbf{run'}(j) \wedge \mathbf{talk'}(m)]$
b. The translation of **John runs or Mary talks** is logically equivalent to:
$[\mathbf{run'}(j) \vee \mathbf{talk'}(m)]$

(29) **Rule T12 (Conjunction and Disjunction of IVs)**

If γ, $\delta \in P_{IV}$ and γ, δ translate into γ', δ' respectively, then
(i) $F_8(\gamma, \delta)$ translates into $\lambda x[\ \gamma'(x) \wedge \delta'(x)\ ]$
(ii) $F_9(\gamma, \delta)$ translates into $\lambda x[\ \gamma'(x) \vee \delta'(x)\ ]$

a. First Key Illustration

(i) **run** translates into **run'**, **talk** translates into **talk'** (T1)
(ii) F_8(**run**, **talk**) translates into λx[**run'**(x) ∧ **talk'**(x)] (T12)
(iii) **run and talk** translates into λx[**run'**(x) ∧ **talk'**(x)] (def. of F_8)
(iv) **John** translates into $\lambda P[P\{j\}]$ (T1)
(v) F_4(**John**, **run and talk**) translates into
[$\lambda P[P\{j\}]$]([^λx[**run'**(x) ∧ **talk'**(x)]]) (T4)
(vi) **John runs and talk** translates into
[$\lambda P[P\{j\}]$]([^λx[**run'**(x) ∧ **talk'**(x)]]) (def. of F_4)[3]

(vii) [$\lambda P[P\{j\}]$]([^λx[**run'**(x) ∧ **talk'**(x)]]) ⇔ (λ-Conv.)
(viii) [^λx[**run'**(x) ∧ **talk'**(x)]]{*j*} ⇔ (CBN)
(ix) [$^\vee$[^λx[**run'**(x) ∧ **talk'**(x)]]](*j*) ⇔ (DUC)
(x) [λx[**run'**(x) ∧ **talk'**(x)](*j*) ⇔ (λ-Conv.)
(xi) [**run'**(*j*) ∧ **talk'**(*j*)]

Note: Thus, in our system, the translation of **John runs and talk(s)** is logically equivalent to [**run'**(*j*) ∧ **talk'**(*j*)], which is also the translation of **John runs and John talks**…

b. Second Key Illustration:

(i) **every man** translates into λP Λx[**man'**(x) → P{x}] (T1, T2)
(ii) **run or talk** translates into λx[**run'**(x) ∨ **talk'**(x)] (T1, T12)
(iii) **every man runs or talk** translates into
[λP Λx[**man'**(x) → P{x}]](^λx[**run'**(x) ∨ **talk'**(x)]) (T4)

(iv) [λP Λx[**man'**(x) → P{x}]](^λx[**run'**(x) ∨ **talk'**(x)]) ⇔ (λ-Conv., α-Conv.)
(v) Λx[**man'**(x) → [^λy[**run'**(y) ∨ **talk'**(y)]]{x}] ⇔ (CBN, DUC)
(vi) Λx[**man'**(x) → [λy[**run'**(y) ∨ **talk'**(y)]](x)] ⇔ (λ-Conv.)
(vii) Λx[**man'**(x) → [**run'**(x) ∨ **talk'**(x)]]
'For all x, if x is a man, then either x runs or x talks'

Note: In our system, the translation of **every man runs or talk(s)** is *not* logically equivalent to the translation of **every man runs or every man talks**

[3] Note that F_4 is defined so that it only inflects the first verb in a conjunction of IVs. This is clearly a problematic prediction of the PTQ system.

(30) **Rule T13 (Disjunction of Terms)**

If $\alpha, \beta \in P_T$ and α, β translate into α', β' respectively, then $F_9(\alpha, \beta)$ translates into $\lambda P\ [\alpha'(P) \vee \beta'(P)]$

Illustration:

(i) **John** translates into $\lambda P[P\{j\}]$, **Mary** translates into $\lambda P[P\{m\}]$ (T1)
(ii) **John or Mary** translates into $\lambda P[\ [\lambda P[P\{j\}]](P) \vee [\lambda P[P\{j\}]](P)]$ (T13)
(iii) **John or Mary runs** translates into $[\lambda P[\ [\lambda P[P\{j\}]](P) \vee [\lambda P[P\{j\}]](P)]]\ (\hat{}\mathbf{run'})$ (T4)

(iv) $[\lambda P[\ [\lambda P[P\{j\}]](P) \vee [\lambda P[P\{j\}]](P)]]\ (\hat{}\mathbf{run'})$ $\Leftrightarrow$ (λ-Conv.)
(v) $[\ [\lambda P[P\{j\}]](\hat{}\mathbf{run'}) \vee [\lambda P[P\{j\}]](\hat{}\mathbf{run'})]$ $\Leftrightarrow$ (λ-Conv.)
(vi) $[\ [\hat{}\mathbf{run'}]\{j\} \vee [\hat{}\mathbf{run'}]\{m\}\ ]$ $\Leftrightarrow$ (CBN)
(vii) $[\ [{}^{\vee}[\hat{}\mathbf{run'}]](j) \vee [{}^{\vee}[\hat{}\mathbf{run'}]](m)\ ]$ $\Leftrightarrow$ (DUC)
(viii) $[\ \mathbf{run'}(j) \vee \mathbf{run'}(m)\ ]$

Note: Thus, our system predicts that the translation of **John or Mary runs** will be logically equivalent to the translation of **John runs or Mary runs.**

Thus far, we've covered about as much of the translation system as we can *without also discussing* the translations of 'Quantifying In' and 'Direct Object' structures…

- **These will be the focus of the next handout…**
- **These are also, in a sense, the 'analytic centerpiece' of the PTQ system…**