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, intensional 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 *intenssion* 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 \[\lambda v_0, v_0, s, l, t > v_0, s, l, t > \{j\}\]

**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**.
(iii) \[\lambda v_0, v_0, s, l, t > v_0, s, l, t > \{j\}\](\[^{\text{smokes'}}\]) = (by assumption, and notation)
(iv) \[^{\text{smokes'}}\]{j} = (by λ-conversion)
(v) \[^{\text{^{\text{smokes'}}}}\]{j} = (by Curly Bracket Notation)
(vi) **smokes’**(j) = (by Down-Up Cancellation)
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 \( \text{Cat} \) 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 \( \varphi \) in English combines syntactically with an expression of category \( B \) (\( i.e. \), \( \varphi \)
is of category \( A/B \))…
  - Then the translation of \( \varphi \) 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 \( \text{walk} \) will be a function of type \( < <s,e>, t > \)
  - Thus, the translation (meaning) of \( \text{walk} \) takes \text{individual concepts} as argument

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

(11) But, Is This Really a ‘Snag’

a. A ‘Pro’ of the Correspondences in (10):
   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. ‘Cons’ of the Correspondences in (10):
   - 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.
A Simpler Category-to-Type Correspondence

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 \textit{Cat} as its domain and is such that:

a. \( f(e) = e \)

b. \( f(t) = t \)

c. \( f(CN) = f(IV) = \langle e, t \rangle \)

d. For all other categories A/B, \( f(A/B) = f(A//B) = \langle \langle s, f(B) \rangle, f(A) \rangle \)

Category-to-Type Correspondences, Under (12)

<table>
<thead>
<tr>
<th>Category</th>
<th>Corresponding Type</th>
<th>Set-Theoretic Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>t (sentences)</td>
<td>t</td>
<td>truth-value</td>
</tr>
<tr>
<td>CN</td>
<td>( \langle e, t \rangle )</td>
<td>function from entities to truth-values</td>
</tr>
<tr>
<td>IV</td>
<td>( \langle e, t \rangle )</td>
<td>function from entities to truth-values</td>
</tr>
</tbody>
</table>
| T (= t/IV)          | \( \langle \langle s, \langle e, t \rangle, \rangle, t \rangle \) | function from properties to truth-values
| IV/t                | \( \langle \langle s, t \rangle, \langle e, t \rangle \rangle \) | function from propositions to \( \langle e, t \rangle \)-functions |
| IV/IV               | \( \langle \langle s, \langle e, t \rangle, \rangle, \langle e, t \rangle \rangle \) | function from properties to \( \langle e, t \rangle \)-functions |
| IAV (= IV/IV)       | \( \langle \langle s, \langle e, t \rangle, \rangle, \langle e, t \rangle \rangle \) | function from properties to \( \langle e, t \rangle \)-functions |
| TV (= IV/T)         | \( \langle \langle s, \langle s, \langle e, t \rangle, t \rangle, \rangle, \langle e, t \rangle \rangle \) | function from the intensions of GQs to an \( \langle e, t \rangle \)-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…

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 \textit{constants} of the logical language.

- Consequently, this function in PTQ will \textit{not} 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, \{\textit{John, Mary, Bill, ninety, he}_0, \textit{he}_1, \textit{he}_2, \ldots\}\n
  - \textit{necessarily}

  - \textit{be}

\(^2\) Such functions, I will (somewhat misleadingly) refer to as ‘GQs’
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 $A \in \text{Cat}$, $\alpha \in B_A$, and $\alpha$ is in the domain of $g$, $g(\alpha) \in \text{Con}_{\text{e},T}$

A Picture of the ‘Lexical’ Translation Function

a. The IVs: $g(\text{run}), g(\text{walk}), g(\text{talk}), g(\text{rise}), g(\text{change}) \in \text{Con}_{\text{e},T}$

b. The CNs: $g(\text{man}), g(\text{woman}), g(\text{park}), g(\text{fish}), g(\text{pen}), g(\text{unicorn})$, $g(\text{price}), g(\text{temperature}) \in \text{Con}_{\text{e},T}$

c. The IV/t’s: $g(\text{believe that}), g(\text{assert that}) \in \text{Con}_{\text{e},T}$

d. The IV//IVs: $g(\text{try to}), g(\text{wish to}) \in \text{Con}_{\text{e},T}$

e. The IAVs: $g(\text{rapidly}), g(\text{slowly}), g(\text{voluntarily})$, $g(\text{allegedly}) \in \text{Con}_{\text{e},T}$

f. The TVs: $g(\text{find}), g(\text{lose}), g(\text{eat}), g(\text{love}), g(\text{date})$, $g(\text{seek}), g(\text{conceive}) \in \text{Con}_{\text{e},T}$

g. The IAV/Ts: $g(\text{in}), g(\text{about}) \in \text{Con}_{\text{e},T}$

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. The Convention: If $\alpha$ is in the domain of $g$, then $\alpha' = g(\alpha)$

b. Illustration:

(i) $\text{run'} = g(\text{run})$ [i.e., whatever $\text{Con}_{\text{e},T}$ $g$ maps run to]

(ii) $\text{man'} = g(\text{man})$ [i.e., whatever $\text{Con}_{\text{e},T}$ $g$ maps man to]

(iii) $\text{find'} = g(\text{find})$ [i.e., whatever $\text{Con}_{\text{e},T}$ $g$ maps find to]

(iv) $\text{believe that'} = g(\text{believe that})$ [i.e., whatever $\text{Con}_{\text{e},T}$ $g$ maps believe that to]
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. 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) $P,Q$ abbreviates $v_{0,<s,<e,t>}$, $v_{1,<s,<e,t>}$ (property variable)
(iv) $P$ abbreviates $v_{0,<s,<e,t>}$ (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

Rule T1 (For Basic Expressions)

a. If $\alpha$ is in the domain of $g$, then $\alpha$ translates into $g(\alpha)$

Illustration: run translates to run' ($= g(\text{run})$)
man translates to man' ($= g(\text{man})$)
find translates to find' ($= g(\text{find})$)
believes that translates to believes that' ($= g(\text{believes that})$)

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

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 $\lambda P[P\{j\}]$
(ii) Mary translates into $\lambda P[P\{m\}]$
(iii) Bill translates into $\lambda P[P\{b\}]$
(iv) ninety translates into $\lambda P[P\{n\}]$

Note:
• In PTQ itself, the actual translation of (e.g.) John is $\lambda P[P\{^j\}]$
• 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. he$_n$ translates into $\lambda P[P\{x_n\}]$

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 $\delta \in P_{vIV}$, $\beta \in P_{IV}$, and $\delta, \beta$ translate into $\delta'$, $\beta'$ respectively, then $F_4(\delta, \beta)$ translates into $\delta'([\beta'])$

Note: Just as promised in (7), the translation of a term $\delta$ will take as argument the intension of the translation of an IV $\beta$.

(20) Illustration of Rule T4

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

b. Simplifying the Translation
   (i) $[\lambda P[P\{j\}]](^{\text{run'}})$ ⇔ (λ-Conv.)
   (ii) $[^{\text{run'}}]\{j\}$ ⇔ (CBN)
   (iii) $[^{\text{run'}}][j]$ ⇔ (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):

  $$\forall x \big[ x = \alpha \big], \text{ where } \alpha \text{ is } j, m, b, \text{ or } n$$

  *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 ‘$\lambda$-convert’ them into the scope of $\forall, \square, W, H$
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 $\alpha, \delta$ translate into $\alpha', \delta'$ respectively, then

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

Illustrations:

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

Note:
Thus, the translation of John doesn't run is logically equivalent to $\neg \text{run}'(j)$

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) $F_{12}(\text{John, run})$ translates into $W[\lambda P[P\{j\}]][\text{run}']$ (T17)
(iii) John will run translates into $W[\lambda P[P\{j\}]][\text{run}']$ (def. of $F_{12}$)
(iv) $W[\lambda P[P\{j\}]][\text{run}'] \iff W \text{run}'(j)$ ($\lambda$-Conv., CBN, DUC)

Note:
Thus, the translation of John will run is logically equivalent to $W \text{run}'(j)$

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) $F_{15}(\text{John, run})$ translates into $\neg H [\lambda P[P\{j\}]][\text{run}']$ (T17)
(iii) John hasn't run translates into $\neg H [\lambda P[P\{j\}]][\text{run}']$ (def. of $F_{15}$)
(iv) $\neg H [\lambda P[P\{j\}]][\text{run}'] \iff \neg H \text{run}'(j)$ ($\lambda$-Conv., CBN, DUC)

Note:
Thus, the translation of John hasn't run is logically equivalent to $\neg H \text{run}'(j)$

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 $\zeta$ translates into $\zeta'$, then:

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

Illustrations:

a. (i) man translates into man’ (T1)
   (ii) $F_0(\text{man})$ translates into $\lambda P \Lambda x[\text{man}’(x) \rightarrow P\{x\}]$ (T2)
   (iii) every man translates into $\lambda P \Lambda x[\text{man}’(x) \rightarrow P\{x\}]$ (def. of $F_0$)

b. (i) man translates into man’ (T1)
   (ii) $F_1(\text{man})$ translates into $\lambda P \forall y [\Lambda x [\text{man}’(x) \leftrightarrow x = y] \land P\{y\}]$ (T2)
   (iii) the man translates into $\lambda P \forall y [\Lambda x [\text{man}’(x) \leftrightarrow x = y] \land P\{y\}]$ (def. of $F_1$)

c. (i) man translates into man’ (T1)
   (ii) $F_2(\text{man})$ translates into $\lambda P \forall x[\text{man}’(x) \land P\{x\}]$ (T2)
   (iii) a man translates into $\lambda P \forall x[\text{man}’(x) \land P\{x\}]$ (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 $\lambda P \forall x[\text{man}’(x) \land P\{x\}]$ (T2)
   (ii) run translates into run’ (T1)
   (iii) $F_4(\text{a man, run})$ translates into $[\lambda P \forall x[\text{man}’(x) \land P\{x\}]][\text{run}’]$ (T4)
   (iv) a man runs translates into $[\lambda P \forall x[\text{man}’(x) \land P\{x\}]][\text{run}’]$ (def. of $F_4$

(v) $[\lambda P \forall x[\text{man}’(x) \land P\{x\}]][\text{run}’] \leftrightarrow [\lambda P \forall x[\text{man}’(x) \land P\{x\}]][\text{run}’]$ (\lambda-Conv.)
(vi) $Vx[\text{man}’(x) \land [\text{run}’]\{x\}] \leftrightarrow [\text{run}’]\{x\}$ (CBN)
(vii) $Vx[\text{man}’(x) \land [\text{run}’]\{x\}] \leftrightarrow [\text{run}’]\{x\}$ (DUC)
(viii) $Vx[\text{man}’(x) \land \text{run}(x)]$ (DUC)
b. (i) **a man** translates into $\lambda P \forall x[\text{man}'(x) \land P\{x\}]$  
(ii) **run** translates into $\text{run}'$ 
(iii) $F_{15}(\text{a man, run})$ translates into $\neg H[\lambda P \forall x[\text{man}'(x) \land P\{x\}])(\text{run}')$ 
(iv) **a man hasn’t run** translates into $\neg H[\lambda P \forall x[\text{man}'(x) \land P\{x\}])(\text{run}')$ 
(v) $\neg H[\lambda P \forall x[\text{man}'(x) \land P\{x\}])(\text{run}') \iff (\lambda-\text{Conv., CBN, DUC})$ 
(vi) $\neg H \forall x[\text{man}'(x) \land \text{run'}(x)]$

(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…

   o 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.

   o 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 $\varphi, \psi \in P$, and $\varphi, \psi$ translates into $\varphi’, \psi’$ respectively, then

(i) $F_8(\varphi, \psi)$ translates into $[\varphi’ \land \psi’]$ 
(ii) $F_9(\varphi, \psi)$ translates into $[\varphi’ \lor \psi’]$ 

**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:

   $[\text{run'}(j) \land \text{talk'}(m)]$

b. The translation of **John runs or Mary talks** is logically equivalent to:

   $[\text{run'}(j) \lor \text{talk'}(m)]$
(29) **Rule T12 (Conjunction and Disjunction of IVs)**

If $\gamma, \delta \in P_{IV}$ and $\gamma, \delta$ translate into $\gamma', \delta'$ respectively, then

(i) $F_8(\gamma, \delta)$ translates into $\lambda x[\gamma'(x) \land \delta'(x)]$

(ii) $F_9(\gamma, \delta)$ translates into $\lambda x[\gamma'(x) \lor \delta'(x)]$

a. **First Key Illustration**

(i) *run* translates into *run’*, *talk* translates into *talk’*  
   (T1)

(ii) $F_8(\text{run, talk})$ translates into $\lambda x[\text{run’}(x) \land \text{talk’}(x)]$  
   (T12)

(iii) *run and talk* translates into $\lambda x[\text{run’}(x) \land \text{talk’}(x)]$  
   (def. of $F_8$)

(iv) *John* translates into $\lambda P[P\{j\}]$  
   (T1)

(v) $F_4(\text{John, run and talk})$ translates into  
   $[\lambda P[P\{j\}]([\lambda x[\text{run’}(x) \land \text{talk’}(x)]])]$  
   (T4)

(vi) *John runs and talk* translates into  
   $[\lambda P[P\{j\}]([\lambda x[\text{run’}(x) \land \text{talk’}(x)]])]$  
   (def. of $F_4$)  

Note: Thus, in our system, the translation of *John runs and talk(s)* is logically equivalent to $[\text{run’}(j) \land \text{talk’}(j)]$, which is also the translation of *John runs and John talks*…

b. **Second Key Illustration:**

(i) *every man* translates into $\lambda P A x[\text{man’}(x) \rightarrow P\{x\}]$  
   (T1, T2)

(ii) *run or talk* translates into $\lambda x[\text{run’}(x) \lor \text{talk’}(x)]$  
   (T1, T12)

(iii) *every man runs or talk* translates into  
   $[\lambda P A x[\text{man’}(x) \rightarrow P\{x\}]([\lambda x[\text{run’}(x) \lor \text{talk’}(x)]])]$  
   (T4)

(iv) $[\lambda P A x[\text{man’}(x) \rightarrow P\{x\}]([\lambda x[\text{run’}(x) \lor \text{talk’}(x)]]) \Leftrightarrow (\lambda-\text{Conv.}, \alpha-\text{Conv.})$

(v) $A x[\text{man’}(x) \rightarrow [\lambda y[\text{run’}(y) \lor \text{talk’}(y)]\{x\}] \Leftrightarrow (\text{CBN, DUC})$

(vi) $A x[\text{man’}(x) \rightarrow [\lambda y[\text{run’}(y) \lor \text{talk’}(y)]\{x\}] \Leftrightarrow (\lambda-\text{Conv.})$

(vii) $A x[\text{man’}(x) \rightarrow [\text{run’}(x) \lor \text{talk’}(x)]]$

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 \( \alpha, \beta \in \mathcal{P} \) and \( \alpha, \beta \) translate into \( \alpha', \beta' \) respectively, then \( F_0(\alpha, \beta) \) translates into

\[
\lambda P \left[ \alpha'(P) \lor \beta'(P) \right]
\]

**Illustration:**

(i) **John** translates into \( \lambda P\left[ P\{j\} \right] \), **Mary** translates into \( \lambda P\left[ P\{m\} \right] \) (T1)

(ii) **John or Mary** translates into \( \lambda P\left[ \left[ \lambda P\left[ P\{j\}\right]\right](P) \lor \left[ \lambda P\left[ P\{j\}\right]\right](P) \right] \) (T13)

(iii) **John or Mary runs** translates into \( \left[ \lambda P\left[ \left[ \lambda P\left[ P\{j\}\right]\right](P) \lor \left[ \lambda P\left[ P\{j\}\right]\right](P) \right] \right] \) (^run') (T4)

(iv) \( \left[ \lambda P\left[ \left[ \lambda P\left[ P\{j\}\right]\right](P) \lor \left[ \lambda P\left[ P\{j\}\right]\right](P) \right] \right] \) (^run') \( \Leftrightarrow \) (\( \lambda \)-Conv.)

(v) \( \left[ \left[ \lambda P\left[ P\{j\}\right]\right](^\text{run'}) \lor \left[ \lambda P\left[ P\{j\}\right]\right](^\text{run'}) \right] \) \( \Leftrightarrow \) (\( \lambda \)-Conv.)

(vi) \( \left[ \left\langle ^\text{run'} \right\rangle \{j\} \lor \left\langle ^\text{run'} \right\rangle \{m\} \right] \) \( \Leftrightarrow \) (CBN)

(vii) \( \left[ \left\langle ^\text{run'} \right\rangle \{j\} \lor \left\langle ^\text{run'} \right\rangle \{m\} \right] \) \( \Leftrightarrow \) (DUC)

(viii) \( \left[ \text{run'}(j) \lor \text{run'}(m) \right] \)

**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…