Montague’s Theory of Translation: Laying the Groundwork

1. Some Context and Questions to Have in the Background

(1) Fundamental Fact: Translation ≠ Semantics

- Translating expressions from a language \( L \) into a language \( L' \) doesn’t necessarily tell you anything about the semantics of the expressions in language \( L \).
  - For example, simply knowing that (1a) can be translated as (1b) doesn’t tell you anything about what (1a) means.

  a. **Tlingit Sentence:** Ax éet yaan uwaháa
  b. **Haida Translation:** Dìì.uu q’wiidang gwaa.

- However, translating an expression from a language \( L \) into a language \( L' \) whose semantics are known *can* inform you of the semantics of the expressions in \( L \).
  - For example, given that you speak English, knowing that (1a) can be translated as (1c) *would* inform you of the meaning of (1a).

  c. **English Translation:** I’m hungry. (~ Hunger moves to me imperceptibly.)

- Of course, even such translations don’t necessarily mean one has a compositional semantics for language \( L \).

(2) Question (Not Necessarily Montague’s)

Given our background theory of language and meaning, under what conditions (if any) can we guarantee that translating from one language \( L \) into another language \( L' \) gives us a compositional semantics for \( L \).

(3) Some Historical Context: Semantics in Generative Grammar Before Montague

Prior to Montague (and for some time afterwards), generative linguists conceived of natural language semantics as having the following goal:

- Develop a theory of the system that maps syntactic structures to some kind of ‘mentalese’ (conceptual structures) encoding the information in the sentence.

**Illustration: Jackendoff’s ‘Conceptual Semantics’**

“A dog is a reptile” \( \rightarrow \) \([\text{State Is-included in} ( [\text{Thing Type: Dog}] ) ( [\text{Thing Type: Reptile}] )] \)
(Jackendoff 1983: 96)

---

1 These notes are based upon material in the following readings: Halvorsen & Ladusaw (1979), Dowty et al. (1981) Chapter 8, and Thomason (1974) Chapter 7 (Montague’s “Universal Grammar”).

Early and Perennial Criticism of Such Approaches

Given the fundamental fact in (1), simply translating a sentence of English to a sentence of ‘mentalese’ isn’t (necessarily) providing a semantics for the English sentence.

• The problem of providing a semantics for English now becomes the problem of providing a semantics for the ‘mentalese’ notation (one that had never been taken up)

Question (Not Necessarily Montague’s)

Under what conditions can this problem in (4) be circumvented? Again, under what conditions (if any) can we guarantee that translating from one language $L$ into another language (notation) $L'$ gives us a compositional semantics for $L$.

2. Key Ingredient: First Order Logic as a Family of Languages

Up to now, we’ve been using the term “First Order Logic” to refer to a single language…

• However, at this point, it will be important to view First Order Logic not as a single language, but rather as a family of infinitely many different languages…

First Order Logic $A$: $\exists x ( (Px) \& ((Qa)b) )$
First Order Logic $B$: $\exists x ( (\text{dog' } x) \& ((\text{loves' bill'} ) \text{ mary' }))$
First Order Logic $C$: $\exists x ( (\mathcal{O} \ast ) \& (( \mathcal{F} \ast ) \star ) )$

First Order Language (Logic)

A first order language (first order logic) is a language whose vocabulary of symbols satisfies the conditions in (6a) and whose WFFs satisfy the conditions in (6b).

a. The Vocabulary of a First Order Language (Logic):
(i) The Logical Constants: $\neg, \&, \forall, (v, \to, \exists$ are ‘abbreviations’)
(ii) Syntactic Symbols: ( , )
(iii) The Non-Logical Constants:
   1. A countable set of predicate letters (with associated arities)
   2. A countable set of individual constants
   3. A countably infinite set of variables $\{x, y, z, \ldots, x_1, x_2, x_3, \ldots \}$

b. The Well-Formed Formulas of a First Order Language (Logic):
(i) If $\varphi$ is an $n$-ary predicate letter and each of $\alpha_1, \ldots, \alpha_n$ is either an individual constant or a variable, then Concat(…(Concat(Concat($\varphi, \alpha_1), \alpha_2), \ldots, \alpha_n) \in WFF
(ii) If $\varphi, \psi \in WFF$, then $\neg \varphi \in WFF$ and $(\varphi \& \psi) \in WFF
(iii) If $\varphi \in WFF$ and $v$ is a variable, then $\forall v \varphi \in WFF$
(7) **Remarks**

a. Note that a first order language only needs a *countable* set of predicate letters and individual constants. Thus, in a first order language, those sets can be *finite*.

b. Note that we are requiring a first order language to use the vocabulary in (6ai, ii), the (infinite) variables in (iii), and the exact syntax rules in (6b).
   - Thus, a logic in ‘Polish notation’ wouldn’t be a FOL according to (6); nor would one making use of the alternate symbols ‘¬’ and ‘∧’
   - The reason for this is simply because we need to keep *some* things constant between FOLs; it works for us right now to keep these constant.

c. The definition in (6) defines an infinite set of different languages/logics.
   - We can think of the general term ‘First Order Logic’ as referring to this infinite set of different languages.

(8) **Illustration: The First Order Language ‘Politics’**

The language ‘Politics’ is the first order language whose vocabulary is as in (8a) and whose WFFs are defined in (8b).

a. **The Vocabulary of ‘Politics’**
   (i) *The Logical Constants:*  ¬, &, ∀,
   (ii) *Syntactic Symbols:*  (, )
   (iii) *The Non-Logical Constants:*
      1. Predicate Letters:
         - Unary Predicate Letters:  { *smokes*’ }
         - Binary Predicate Letters:  { *loves*’ }
      2. Individual Constants:  { *michelle*’, *barack*’, *mitt*’ }
      3. Variables:  {x, y, z, …, x₁, x₂, x₃, … }

b. **The WFFs of ‘Politics’**  (exactly as in (6b))

(9) **Some Illustrative Formulae of Politics**

a. ¬(*smokes*’ *barack’)

b. ( *loves*’ *barack’) *michelle’

c. ¬(( *smokes* *barack’) & ( *smokes*’ *mitt’))

d. ∀x ¬(( *smokes*’ x) & ¬(( *loves*’ *mitt’) x))
3. **Introducing the Central Characters**

(10) **A Roadmap of Where We’re Headed**

We’re going to build up Montague’s theory of translation by showing how a fragment of English can be ‘translated’ (formally) into a fragment of Politics, and then showing how that (formal) translation also gives us a semantics for the fragment of English.

a. **Step One:** Build the relevant fragment of Politics

b. **Step Two:** Try to build the relevant fragment of English

c. **Step Three:** NOTICE A FUNDAMENTAL PROBLEM IN STEP TWO

d. **Step Four:** Fix that problem, leading to a further refinement of our definition of what a language is...

As we’ve done before, we’re going to make our lives easier by putting aside quantification for the moment...

(11) **A Useful Fragment of Politics: Politics-NoQ**

The language ‘Politics-NoQ’ is the language whose vocabulary is as in (11a) and whose WFFs are defined in (11b).

a. **The Vocabulary of ‘Politics-NoQ’**

   (i) *The Logical Constants:* \(\sim, \&\)

   (ii) *Syntactic Symbols:* ( , )

   (iii) *The Non-Logical Constants:*

      1. Predicate Letters:
         - Unary Predicate Letters: \{ smokes’ \}
         - Binary Predicate Letters: \{ loves’ \}

      2. Individual Constants: \{ michelle’, barack’, mitt’ \}

b. **The WFFs of ‘Politics-NoQ’**

   (i) If \(\varphi\) is an n-ary predicate letter and each of \(\alpha_1, \ldots, \alpha_n\) is an individual constant, then \(\text{Concat}(\ldots(\text{Concat}(\varphi, \alpha_1), \alpha_2), \ldots, \alpha_n) \in \text{WFF}\)

   (ii) If \(\varphi, \psi \in \text{WFF}\), then \(\sim \varphi \in \text{WFF}\) and \((\varphi \& \psi) \in \text{WFF}\)

(12) **Some Illustrative Formulae of Politics-NoQ**

a. \(\sim(\text{smokes’ barack’})\)

b. \(( (\text{loves’ barack’) michelle’ })\)

c. \(\sim( (\text{smokes barack’}) \& (\text{smokes’ mitt’}) )\)
Remarks

- Politics-NoQ is *not* a ‘first order language’, as defined in (6).
- In terms of its structure, Politics-NoQ is quite similar to our language FOL-NoQ from the last two handouts.
- Consequently, we can easily see how to characterize Politics-NoQ in terms of our general (Montagovian) definition of a language.

The Language Politics-NoQ (Montagovian Presentation)

The language Politics-NoQ is the structure \(< A, F_\gamma, X_\tau, S, t >_\gamma \subseteq \{\text{Concat, Not, And}\}, \tau \in T\)

where:

a. \(< A, F_\gamma >_\gamma \subseteq \{\text{Concat, Not, And}\}\) is the algebra such that
   (i) \(F_{\text{Concat}}, F_{\text{Not}}, F_{\text{And}}\) are as defined previously.
   (ii) \(A\) is the smallest set such that:
       1. \(\{\text{smokes’}, \text{loves’}, \text{michelle’}, \text{barack’}, \text{mitt’}\}\) \(\subseteq A\)
       2. It is closed under \(F_{\text{Concat}}, F_{\text{Not}}, F_{\text{And}}\)

b. The basic categories \(X_\tau\) are such that:
   (i) \(X_e = \{\text{michelle’}, \text{barack’}, \text{mitt’}\}\)
   (ii) \(X_{<e,t>} = \{\text{smokes’}\}\)
   (iii) \(X_{<e,<e,t>)} = \{\text{loves’}\}\)
   (iii) For all other types \(\tau \in T\), \(X_\tau = \emptyset\).

c. The set \(S\) is the following (infinite) set of syntactic rules:
   (i) \(< F_{\text{Not}}, < t >, t >\)
   (ii) \(< F_{\text{And}}, < t, t >, t >\)
   (iii) \(< F_{\text{Concat}}, < < \sigma, \tau >, \sigma >, \tau >, \) for all \(\sigma, \tau \in T\)

Some Illustrative Members of Category \(C_\tau\) of Politics-NoQ

a. \(\neg(\text{smokes’ barack’})\)

b. \((\text{loves’ barack’}) \text{ michelle’}\)

C. \(\neg(\text{smokes barack’}) \& (\text{smokes’ mitt’})\)

Given this structural similarity between Politics-NoQ and FOL-NoQ, it’s also rather easy to set up a (Fregan) interpretation for Politics-NoQ!
A (Fregean) Interpretation of Politics-NoQ

Let the set $S = \{ \text{Michelle, Barack, Mitt} \}$. Let $B = \langle B, G_\gamma, f_\gamma \rangle$ be the Fregean interpretation based on $S$, such that $f$ consists of the following mappings:

a. $f(\text{michelle}') = \text{Michelle}$
b. $f(\text{barack}') = \text{Barack}$
c. $f(\text{mitt}') = \text{Mitt}$
d. $f(\text{smokes}') = h = \{ \langle \text{Michelle, 0}, \langle \text{Barack,1}, \langle \text{Mitt,0}\rangle \} \}$
e. $f(\text{loves}') = j = \begin{cases} \text{Michelle} & \rightarrow \{ \langle \text{Michelle, 1}, \langle \text{Barack,1}, \langle \text{Mitt,0}\rangle \} \\
\text{Barack} & \rightarrow \{ \langle \text{Michelle, 1}, \langle \text{Barack,1}, \langle \text{Mitt,0}\rangle \} \\
\text{Mitt} & \rightarrow \{ \langle \text{Michelle, 0}, \langle \text{Barack,0}, \langle \text{Mitt,1}\rangle \} \end{cases}$

Note: We’ve basically interpreted ‘smokes’ as the property ‘smokes’, and we’ve interpreted ‘loves’ as the (curried) relation ‘x is loved by y’. ²

Using the Fregean Interpretation in (16) to Interpret Sentences of Politics-NoQ

(i) $g( \neg(\text{smokes'} \text{ barack'}))$ = (by definition of Politics-NoQ)
(ii) $g( F_{\neg}( F_{\text{Concat}}(\text{smokes'}, \text{barack'})))$ = (by homomorphism property of $g$)
(iii) $G_{\neg}( g(F_{\text{Concat}}(\text{smokes'}, \text{barack'})))$ = (by homomorphism property of $g$)
(iv) $G_{\neg}( G_{\text{Concat}}( g(\text{smokes'}), g(\text{barack'})))$ = (by definition of $g$)
(v) $G_{\neg}( G_{\text{Concat}}( f(\text{smokes'}), f(\text{barack'})))$ = (by definition of $f$ in (16))
(vi) $G_{\neg}( G_{\text{Concat}}( h, \text{Barack} ))$ = (by definition of $G_{\text{Concat}}$)
(vii) $G_{\neg}( h(\text{Barack}) )$ = (by definition of $h$ in (16d))
(viii) $G_{\neg}( 1 )$ = (by definition of $G_{\neg}$)
(ix) 0

² That is, in the notation of Heim & Kratzer (1998), we’re interpreting loves’ as $[\lambda y: [\lambda x: x \text{ loves } y]]$. The semanticists in the house can probably guess why we’re doing this ; )
Finally, to get our third player on the field, let’s recall that fragment of English that we defined a while back…

(18) **The Definition of ‘Mini-English’**

‘Mini-English’ is the structure \( \langle E, K_\gamma, X_\delta, S \rangle \in \{\text{Concat, Not, And}\}, \delta \in \Delta \) such that:

a. **The Syntactic Categories:** \( \Delta = \{\text{NP, IV, TV, S}\} \)

b. **The Syntactic Operations:**
   (i) \( K_{\text{Concat}} = \text{Merge} \) (from previous notes)
   (ii) \( K_{\text{Not}} = \text{Not}_E \) (from previous notes)
   (iii) \( K_{\text{And}} = \text{And}_E \) (from previous notes)

c. **The Basic Expressions:**
   (i) \( X_{\text{NP}} = \{\text{Barack, Michelle, Mitt}\} \)
   (ii) \( X_{\text{IV}} = \{\text{smokes}\} \)
   (iii) \( X_{\text{TV}} = \{\text{loves}\} \)
   (iv) \( X_S = \emptyset \)

c. **The Syntactic Algebra:**

   E is the smallest set such that:
   (i) For all \( \delta \in \Delta, X_\delta \subseteq E \).
   (ii) E is closed under the operations \( K_{\text{Concat}}, K_{\text{Not}} \) and \( K_{\text{And}} \)

d. **The Syntactic Rules:** The set \( S_E \) consists of the following tuples:
   (i) \( < K_{\text{Concat}}, < \text{TV}, \text{NP} >, \text{IV} > \)
   (ii) \( < K_{\text{Concat}}, < \text{NP}, \text{IV} >, S > \)
   (iii) \( < K_{\text{And}}, < S, S >, S > \)
   (iv) \( < K_{\text{Not}}, < S >, S > \)

(19) **Illustrative Sentence (Expression of Category C_S) of Mini-English**

*It is not the case that Barack smokes and Mitt smokes.*

\(< \text{It is not the case that Barack smokes and Mitt smokes, } K_{\text{And}} >\)

\(< \text{It is not the case that Barack smokes, } K_{\text{Not}} > \quad < \text{Mitt smokes, } K_{\text{Concat}} >\)

\(< \text{Barack smokes, } K_{\text{Concat}} > \quad \text{Mitt smokes}\)

\(\text{Barack smokes}\)
Unfortunately, there’s a fundamental problem with the language as defined in (18). To see this, recall our ultimate goal, informally sketched out below.

(20) **Our Goal for a Theory of Translation**

We want to develop a way of homomorphically mapping expressions of mini-English to expressions of Politics-NoQ (so that we can ultimately get a semantics for English)

Indirect Interpretation in a Picture (Oversimplified):

$$<E, K_\gamma, X_\delta, S, S>_{\gamma} \in \{\text{Concat, Not, And}\}, \delta \in \Delta$$

$$h \ (\text{translation from mini-Eng to Pol-NoQ})$$

$$<A, F_\gamma, X_\tau, S, t>_{\gamma} \in \{\text{Concat, Not, And}\}, \tau \in \Gamma$$

$$g (\text{semantic interpretation of Politics-NoQ})$$

$$<B, G_\gamma, f>_{\gamma} \in \{\text{Concat, Not, And}\}$$

(21) **Critical Problem**

We currently conceive of the set E (expressions of mini-English) as consisting of *strings of English words*.

- However, some such strings in our mini-English language can be created in *more than one way* from the syntactic operations (and rules) of our language.

**Example:**

$$<\text{It is not the case that Barack smokes and Mitt smokes}, K_{\text{Not}}>$$

$$<\text{Barack smokes and Mitt smokes}, K_{\text{And}}>$$

- Intuitively, we want these two different ways of constructing the mini-English sentence to lead to two different Politics-NoQ translations:
  - $$\sim (\text{smokes’ barack’} \ & \ (\text{smokes’ mitt’}))$$
  - $$\sim (\text{smokes’ barack’} \ & \ (\text{smokes’ mitt’}))$$

- But if the translation $h$ is a mapping from *strings* of English to expressions of Politics-NoQ, each such string will be mapped to only one translation!
Some More General Remarks

- Ultimately, we want there to be two different Politics-NoQ translations of sentence (19) because we want this string to be paired with two different semantic values.
  - There is a reading where (19) is true
    \[
    \approx \neg((\text{smokes' barack'}) \land (\text{smokes' mitt'}))
    \]
  - There is a reading where (19) is false
    \[
    \approx \neg((\text{smokes' barack'}) \land (\text{smokes' mitt'}))
    \]

- Also, we have the background belief that (19) has these two readings because of the different ways that the sentence can be constructed in English
  - (i.e., it’s not because of any ambiguity in what the words mean…)

- But, if we are semantically interpreting strings of English words – and ‘interpretation’ is conceived as a homomorphism (function) from expressions to meanings – then each string will be mapped to exactly one meaning.

- Thus, unlike with Politics-NoQ, it is not feasible to build a semantics that interprets (directly or indirectly) English strings.

- In LING 610, this problem doesn’t even arise, because right from the start we’re interpreting phrase structure trees
  - After all, a given tree is only ever constructed in one way by Merge and Move...  

Another Critical Problem

In the picture in (20), translation is a homomorphism from \(<E, K_\gamma > \gamma \in \{\text{Concat, Not, And}\}\) for mini-English to the algebra \(<A, F_\gamma > \gamma \in \{\text{Concat, Not, And}\}\) for Politics-NoQ

- Under such a homomorphism, we’d naturally want \(K_{\text{Concat}}\) and \(F_{\text{Concat}}\) to correspond. This will get the right interpretation for VPs (IVs) after all:
  \[
  h(\text{loves michelle}) \quad = \quad h(K_{\text{Concat}}(\text{loves, Michelle})) = \quad F_{\text{Concat}}(h(\text{loves}), h(\text{Michelle})) \quad = \quad F_{\text{Concat}}(\text{loves', michelle'}) \quad = \quad (\text{loves' michelle'})
  \]

- However, this will get the wrong result for sentences! Sentences will end up mapped to syntactic garbage in A.

  \[
  h(\text{Barack smokes}) \quad = \quad h(K_{\text{Concat}}(\text{Barack, smokes})) \quad = \quad F_{\text{Concat}}(h(\text{Barack}), h(\text{smokes})) \quad = \quad F_{\text{Concat}}(\text{barack', smokes'}) \quad = \quad (\text{barack' smokes'})
  \]

Syntactic Garbage!!!
More General Remarks

Obviously, what we want is for \( h(\text{Barack smokes}) = (\text{smokes’ barack’}) \)

But, there is no operation in the algebra for Politics-NoQ which will take as argument the translation of \( \text{Barack (barack’)} \) and the translation of \( \text{smokes (smokes’)} \) and return the formula \( (\text{smokes’ barack’}) \)

So maybe our homomorphic translation function \( h \) shouldn’t actually be a homomorphism to the algebra \( < A, F, \gamma > \in \{\text{Concat, Not, And}\} \)

Maybe it should be a homomorphism to some other algebra that we can construct from \( < A, F, \gamma > \in \{\text{Concat, Not, And}\} \)

In these notes, we’ll deal only with the problem in (21)-(22)...
In the next set of notes, we’ll tackle the problem in (23)-(24)...

4. Montague’s Notion of a ‘Disambiguated Language’

What We Want

We want it to be that the interpreted expressions of our language can only ever be created from the syntactic operations (rules) in \textit{exactly one way}.

This way, we won’t ever have to worry about interpreting ‘syntactically ambiguous’ expressions \( (\text{because they just won’t exist in our language}) \).

Preview of Where This is Going:
We’ll relate such ‘syntactically unambiguous’ expressions to sentence strings of English via a special operation (akin to ‘linearization’ or ‘Spell Out’).

Question
Below we have our earlier (Montagovian) definition of a language. \textit{What do we have to add to this to ensure that no expressions are syntactically ambiguous?}

A language \( L \) is a structure \( < A, F, X_\delta, S, \delta_0 > \in \Gamma, \delta \in \Delta \) such that:

a. \( < A, F, > \in \Gamma \) is an algebra.

b. \( A \) is the smallest set such that:
   i. For all \( \delta \in \Delta, X_\delta \subseteq A \); (ii) \( A \) is closed under the operations \( F, \gamma \) for all \( \gamma \in \Gamma \)

c. \( S \) is a set of sequences of the form \( < F, \gamma, < \delta_1, ..., \delta_n, \delta >, \delta > \), where \( \gamma \in \Gamma, F, \gamma \) is an \( n \)-ary operation, and \( \delta_1, ..., \delta_n, \delta \in \Delta \)

b. \( \delta_0 \in \Delta \)
(27) **Montague’s Answer**  
At a minimum, we need to ensure that:

a. Nothing in the basic expressions $X_\delta$ (lexical items) can also be constructed by the syntactic operations.  
   
   That is: $X_\delta$ and the range of $F_\gamma$ are disjoint for all $\delta \in \Delta$ and $\gamma \in \Gamma$

b. No element of $A$ will be the output of two different operations $F_\gamma$ and $F_\gamma'$

c. No single operation $F_\gamma$ will take two different inputs $a, a' \in A$ and give the same output.

   That is: For all sequences $a_1, \ldots, a_n \in A^n$ and $a'_1, \ldots, a'_m \in A^m$, if $F_\gamma(a_1, \ldots, a_n) = F_\gamma'(a'_1, \ldots, a'_m)$, then $F_\gamma = F_\gamma'$ and $<a_1, \ldots, a_n> = <a'_1, \ldots, a'_m>$.

**Note that if the conditions in (27) hold, then every expression in $A$ will either be (i) a basic expression (lexical item), or (ii) constructible in exactly one way from the syntactic operations.**

(28) **Montagovian Definition of a ‘Diambiguated Language’**

The following definition now replaces our earlier concept of a language, as well as its concomitant definition.

A disambiguated language is a structure $<A, F_\gamma, X_\delta, S, \delta_0>_{\gamma \in \Gamma, \delta \in \Delta}$ such that:

a. $<A, F_\gamma>_{\gamma \in \Gamma}$ is an algebra.

b. $A$ is the smallest set such that:
   (i) For all $\delta \in \Delta$, $X_\delta \subseteq A$;  (ii) $A$ is closed under the operations $F_\gamma$ for all $\gamma \in \Gamma$

c. $X_\delta$ and the range of $F_\gamma$ are disjoint for all $\delta \in \Delta$ and $\gamma \in \Gamma$

d. For all sequences $a_1, \ldots, a_n \in A^n$ and $a'_1, \ldots, a'_m \in A^m$, if $F_\gamma(a_1, \ldots, a_n) = F_\gamma'(a'_1, \ldots, a'_m)$, then $F_\gamma = F_\gamma'$ and $a_1, \ldots, a_n = a'_1, \ldots a'_m$.

e. $S$ is a set of sequences of the form $<F_\gamma, <\delta_1, \ldots, \delta_n>, \delta>$, where $\gamma \in \Gamma$, $F_\gamma$ is an n-ary operation, and $\delta_1, \ldots, \delta_n, \delta \in \Delta$

f. $\delta_0 \in \Delta$
Remarks
a. Our language Politics-NoQ is such a disambiguaged language.
b. Our language ‘mini-English’ is not a disambiguated language.

c. Potential Problem:
   If we assume that the expressions of mini-English (and English) are strings, then
   we just aren’t going to be able to represent those systems as disambiguated
   languages.

d. Solution:
   Along with the concept of a ‘disambiguaged language’ in (28), we need a more
general concept of a ‘language’.

Montagovian Definition of a Language (Final Version)
A language is a pair < L, R >, where L = < A, Fγ, Xδ, S, δ0γ, δ ∈ Δ is a
   disambiguaged language, and R is a binary relation whose domain is A.

   ▪ This relation R is an ‘ambiguating relation’.
      o It maps expressions in the disambiguaged language L to expressions not
         necessarily from L.

   ▪ Importantly, R can be many-to-one (surjection), and so we can have more than one
      expression from the disambiguaged language being mapped to the same
      expression in the range of R (hence, the term ‘ambiguating’)

Illustration: Politics-NoQ-SansParen
a. Informal Definition
   (i) Vocabulary: Same as Politics-NoQ
   (ii) The WFFs of ‘Politics-NoQ-SansParen’
      1. If φ is an n-ary predicate letter and each of α₁, …, αₙ is either an
         individual constant or a variable, then φα₁…αₙ ∈ WFF
      2. If φ, ψ ∈ WFF, then ~φ ∈ WFF and φ & ψ ∈ WFF

Illustrative Formulae: ~smokes’ barack’
                     loves’ barack’ michelle’
                     ~smokes barack’ & smokes’ mitt’

b. Formal Definition
   The pair < < A, Fγ, Xτ, S, t >γ, τ ∈ T, R >, where the structure
   < A, Fγ, Xτ, S, t >γ ∈ {Concat, Not, And}, τ ∈ T is Politics-NoQ, and R is the function
   that takes any element of A and deletes every parenthesis.

   \[ R( \neg( (\text{smokes barack’}) \& (\text{smokes’ mitt’}) ) ) = \]
   \[ \neg \text{smokes barack’} \& \text{smokes’ mitt’} \]
(32) **Remark**

Every disambiguated language \(< A, F_\gamma, X_\delta, S, \delta_0 >_\gamma \in \Gamma, \delta \in \Delta \rangle\) can also be represented as a language \(< < A, F_\gamma, X_\delta, S, \delta_0 >_\gamma \in \Gamma, \delta \in \Delta, R >\)

- Simply let \(R\) be the identity function!!

(33) **Some Concomitant Definitions**

Let \(L\) be a language \(< < A, F_\gamma, X_\delta, S, \delta_0 >_\gamma \in \Gamma, \delta \in \Delta, R >\).

a. The **proper expressions** of \(L\) is the range of \(R\).

b. The **operation indices** of \(L\) is \(\Gamma\).

c. The **category labels** of \(L\) is \(\Delta\).

d. The **syntactic rules** of \(L\) is \(S\).

e. The **basic expressions** of \(L\) of category \(\delta\) is \(\{ \varphi : \exists \psi \in X_\delta \text{ such that } \psi R \varphi \}\).

f. The **category** \(\delta\) of \(L\) is \(\{ \varphi : \exists \psi \in C_\delta \text{ such that } \psi R \varphi \}\), where \(C_\delta\) is in the family of categories generated by \(< A, F_\gamma, X_\delta, S, \delta_0 >_\gamma \in \Gamma, \delta \in \Delta\).

g. The **meaningful expressions** of \(L\) is the union of all the categories \(\delta\) of \(L\).

h. The **declarative sentences** of \(L\) is the category \(\delta_0\) of \(L\).

i. If \(\varphi\) is a meaningful expression of \(L\), then \(\varphi\) is **syntactically ambiguous** if there are distinct \(\psi, \psi' \in \bigcup_{\delta \in \Delta} C_\delta\) such that \(\psi R \varphi\) and \(\psi' R \varphi\).

j. The language \(L\) is **syntactically ambiguous** if there is a meaningful expression \(\varphi\) of \(L\) which is syntactically ambiguous.

(34) **New Goal**

Given all that we’ve laid out, it seems that we now want to do the following:

a. Represent mini-English as a (syntactically ambiguous) language \(< L, R >\), where
   
   (i) \(L\) is some syntactically unambiguous language, and
   
   (ii) \(R\) can ‘transform’ expressions of \(L\) into expressions of mini-English.

b. Translate mini-English into Politics-NoQ *indirectly*, via translation from \(L\) into Politics-NoQ.

\[
\text{Mini-English} \xrightarrow{\text{Ambiguating} \ ‘R’} \ ‘\text{Disambiguated}’ \ L \xrightarrow{\text{homomorphic translation}} \text{Politics-NoQ}
\]
5. **Representing Mini-English Via a Disambiguated Language**

(35) **Key Question**
Given our new goal in (34), what should the expressions of our ‘disambiguated mini-English’ look like?
- Well, each complex expression must transparently reflect how it was constructed by the syntactic operations…
- That is, for each complex expression, there should be exactly one analysis tree…

(36) **Montague’s Core Insight**
For a syntactically ambiguous natural language like English, we could assume that the syntactically disambiguated expressions are the analysis trees themselves!!

- That is, (mini-)English is a pair $<L, R>$, where the expressions of $L$ are analysis trees, and the relation $R$ simply maps an analysis tree to the string in its root node!

```
< Barack smokes , KConcat >

Barack  smokes  Ambiguating R  Barack smokes.
```

(37) **Remarks**
We’ll see in a moment how to actually implement the idea in (36). For the moment, let’s notice the similarities and differences between this and an ‘LF’-based semantics.

a. **Key Similarity:**
Our semantics does not directly interpret surface strings of English. Rather, it interprets abstract structures that represent how those strings can be derived.

b. **Key Difference:**
Unlike an ‘LF’-based semantics (like in 610), our system doesn’t first construct the analysis tree (LF structure) for a whole sentence and then ‘input’ that into semantic interpretation…

- That is, as will be clear in a few more classes, the syntax and semantics work in tandem with one another…

  - Every time a ‘move’ is made in the syntax to make a structure, a corresponding ‘move’ is made in the semantics to determine a meaning for that structure…

*But how do we construct a ‘disambiguated language’ where the expressions are analysis trees?*
Step One: The Category Labels
The syntactic categories of Disambiguated mini-English will be just the same as before:

\[ \Delta = \{ \text{NP, IV, TV, S} \} \]

The Basic Expressions
The basic expressions of Disambiguated mini-English will be ‘trivial trees’. The following trees consisting of root-nodes without any daughters.

a. \( X_{\text{NP}} = \{ < \text{Barack}, \emptyset >, < \text{Michelle}, \emptyset >, < \text{Mitt}, \emptyset > \} \)
b. \( X_{\text{IV}} = \{ < \text{smokes}, \emptyset > \} \)
c. \( X_{\text{TV}} = \{ < \text{loves}, \emptyset > \} \)
d. \( X_{\text{S}} = \emptyset \)

The Syntactic Operations
Our syntactic operations now take trees (including ‘trivial trees’) as input and output other trees, as defined below.

- In the definitions below, \( \alpha \) and \( \beta \) are trees whose root nodes are ordered pairs. In addition \( \alpha' \) and \( \beta' \) are the first members of the root nodes of \( \alpha \) and \( \beta \) (respectively).

a. \( K_{\text{Concat}}(\alpha, \beta) = < \alpha' \beta', \text{Concat} > \)
   \[ \begin{array}{c} \alpha \\
   \downarrow \\
   \beta \end{array} \]

b. \( K_{\text{Not}}(\alpha) = < \text{it is not the case that } \alpha', \text{Not} > \)
   \[ \begin{array}{c} \alpha' \\
   \downarrow \\
   \alpha \end{array} \]

c. \( K_{\text{And}}(\alpha, \beta) = < \alpha' \text{ and } \beta', \text{And} > \)
   \[ \begin{array}{c} \alpha \\
   \downarrow \\
   \beta \end{array} \]

Just for fun – since it will set us up for something important later, let’s also add the following syntactic operation.

d. \( K_{\text{If}}(\alpha, \beta) = < \text{If } \alpha' \text{ then } \beta', \text{If} > \)
   \[ \begin{array}{c} \alpha \\
   \downarrow \\
   \beta \end{array} \]

The Syntactic Algebra
E is the smallest set such that:

a. For all \( \delta \in \Delta, X_\delta \subseteq E \).
b. E is closed under the operations \( K_{\text{Concat}}, K_{\text{Not}}, K_{\text{And}}, \) and \( K_{\text{If}} \)
(42) **The Syntactic Rules**
We can retain much the same set of syntactic rules $S_E$ that we had before:

a. $< K_{\text{Concat}}, <TV, NP>, IV >$
b. $< K_{\text{Concat}}, <NP, IV>, S >$
c. $< K_{\text{And}}, <S, S>, S >$
d. $< K_{\text{If}}, <S, S>, S >$
e. $< K_{\text{Not}}, <S>, S >$

(43) **The Definition of Our Language: ‘Disambiguated Mini-English’**
The structure $< E, K_\gamma, X_\delta, S_E, S >_\gamma \in \{\text{Concat, Not, And, If}\}, \delta \in \Delta$ where $E, K_\gamma, X_\delta, S_E,$ and $\Delta$ are as defined in (38)-(42).

**Some Illustrative Members of Category $C_S$**

a. $< \text{It is not the case that Barack smokes and Mitt smokes}, \text{Not} >$
   $< \text{Barack smokes and Mitt smokes}, \text{And} >$
   $< \text{Barack smokes}, \text{Concat} >$
   $< \text{Mitt smokes}, \text{Concat} >$
   $< \text{Barack, }, \emptyset >$
   $< \text{smokes, }, \emptyset >$
   $< \text{Mitt, }, \emptyset >$
   $< \text{smokes, }, \emptyset >$

b. $< \text{It is not the case that Barack smokes and Mitt smokes}, \text{And} >$
   $< \text{It is not the case that Barack smokes}, \text{Not} >$
   $< \text{Mitt smokes}, \text{Concat} >$
   $< \text{Barack smokes}, \text{Concat} >$
   $< \text{Mitt, }, \emptyset >$
   $< \text{smokes, }, \emptyset >$

  $< \text{Barack, }, \emptyset >$
  $< \text{smokes, }, \emptyset >$

c. $< \text{Barack loves Michelle}, \text{Concat} >$
   $< \text{Barack, }, \emptyset >$
   $< \text{loves Michelle}, \text{Concat} >$
   $< \text{loves, }, \emptyset >$
   $< \text{Michelle, }, \emptyset >$

(44) **Remark**
Disambiguated Mini-English is indeed a disambiguated language.
- No syntactic operation will ever create a basic expression.
- Because of the way the trees are indexed, no two ops will ever have the same output.
Now, we can use Disambiguated Mini-English to characterize Mini-English as a language, in the sense of (30)

(45) **Montagovian Definition of Mini-English**

Mini-English is the structure $<<E, K_\gamma, X_\delta, S_\gamma, \delta > \in \{\text{Concat, Not, And, If}\}, \delta \in \Delta, R>$, where

a. The structure $<E, K_\gamma, X_\delta, S_\gamma, \delta > \in \{\text{Concat, Not, And, If}\}, \delta \in \Delta$ is Disambiguated Mini-English, as defined in (43).

b. R is a function which takes as input a tree T in E, and returns as output the first member of the root node of T.

\[
R \left( \langle Barack, \emptyset \rangle \right) = R \left( \langle loves \emptyset \rangle \right) = \langle loves \emptyset \rangle
\]

\[R \left( \langle loves Michelle, \emptyset \rangle \right) = R \left( \langle Michelle, \emptyset \rangle \right)
\]

Barack loves Michelle

(46) **Some Illustrative Members of the Category S for Mini-English**

a. Barack smokes.

b. Barack loves Michelle.

c. It is not the case that Barack smokes and Mitt smokes.

(47) **Remark** Mini-English is a syntactically ambiguous language (33j)

- After all, let T be the tree in (43a), and T’ be the tree in (43b).
- $R(43a) = R(43b) = \text{It is not the case that Barack smokes and Mitt smokes}.$

**What Coming Up Next:**

- We now have the following two disambiguated languages:
  - Politics-NoQ
  - Disambiguated Mini-English

- We have an interpretation for Politics-NoQ

- We’re now going to try to find a way of homomorphically mapping expressions of Disambiguated Mini-English to ones of Politics-NoQ