Montague's Theory of Translation: Laying the Groundwork [1]

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: Ax̲ éet yaan uwaháa
b. Haida Translation: Dii.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" → $[_{\text{State}}$ 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").

(4) **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)

(5) **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: ∃x((Px) & ((Qa)b))
First Order Logic B: ∃x((**dog'** x) & ((**loves' bill'**) **mary'**))
First Order Logic C: ∃x((☽ ⋆) & ((¶ ✳) ✱)

(6) **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. <u>The Vocabulary of a First Order Language (Logic):</u>
 (i) *The Logical Constants:* ~, &, ∀, (v, →, ∃ 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. <u>The Well-Formed Formulas of a First Order Language (Logic):</u>
 (i) If φ is an n-ary predicate letter and each of $\alpha_1, \ldots, \alpha_n$ is either an individual constant or a variable, then $\text{Concat}(\ldots(\text{Concat}(\text{Concat}(\varphi,\alpha_1), \alpha_2), \ldots, \alpha_n) \in \text{WFF}$
 (ii) If $\varphi, \psi \in \text{WFF}$, then $\sim\varphi \in \text{WFF}$ and $(\varphi \,\&\, \psi) \in \text{WFF}$
 (iii) If $\varphi \in \text{WFF}$ and υ is a variable, then $\forall\upsilon\varphi \in \text{WFF}$

(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, …, x_1, x_2, x_3, … }

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))

Notice how the vocabulary in our logical language is boldfaced, and followed by primes?...
Get used to that!...

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:* ~, &
   (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 φ is an n-ary predicate letter and each of $\alpha_1, \ldots, \alpha_n$ is an individual constant, then $\text{Concat}(\ldots(\text{Concat}(\text{Concat}(\varphi,\alpha_1), \alpha_2), \ldots, \alpha_n) \in \text{WFF}$
   (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. ~(**smokes' barack'**)
b. ((**loves' barack'**) **michelle'**)
c. ~((**smokes barack'**) & (**smokes' mitt'**))

(13) **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.

(14) **The Language Politics-NoQ (Montagovian Presentation)**

The language Politics-NoQ is the structure $< A , F_\gamma , X_\tau , S , t >_{\gamma \in \{Concat, Not, And\}, \tau \in T}$ where:

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

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

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

(15) **Some Illustrative Members of Category C_t of Politics-NoQ**

a. ~(**smokes' barack'**)
b. ((**loves' barack'**) **michelle'**)
c. ~((**smokes barack'**) & (**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!

(16) **A (Fregean) Interpretation of Politics-NoQ**

Let the set S = { Michelle, Barack, Mitt }. Let **B** = $<B, G_\gamma , f>_{\gamma \in \{\text{Concat, Not, And}\}}$ be the Fregean interpretation based on S, such that *f* consists of the following mappings:

a. *f*(**michelle’**) = Michelle
b. *f*(**barack’**) = Barack
c. *f*(**mitt’**) = Mitt
d. *f*(**smokes’**) = *h* = { <Michelle, 0>, <Barack,1>, <Mitt,0> }

e. *f*(**loves’**)= *j* =

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

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]

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

(i) g(~(**smokes’ barack’**)) = (by definition of Politics-NoQ)

(ii) g(F_{Not}(F_{Concat}(**smokes’**, **barack’**))) = (by homomorphism property of g)

(iii) G_{Not}(g(F_{Concat}(**smokes’** , **barack’**))) = (by homomorphism property of g)

(iv) G_{Not}(G_{Concat}(g(**smokes’**), g(**barack’**))) = (by definition of g)

(v) G_{Not}(G_{Concat}(*f*(**smokes’**), *f*(**barack’**))) = (by definition of f in (16))

(vi) G_{Not}(G_{Concat}(*h*, Barack)) = (by definition of G_{Concat})

(vii) G_{Not}(*h*(Barack)) = (by definition of h in (16d))

(viii) G_{Not}(1) = (by definition of G_{Not})

(ix) 0

[2] That is, in the notation of Heim & Kratzer (1998), we’re interpreting **loves’** as [λy: [λx: x 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 $< E, K_{\gamma}, X_{\delta}, S_E, S >_{\gamma \in \{\text{Concat, Not, And}\},\ \delta \in \Delta}$ such that:

a. <u>The Syntactic Categories:</u> $\Delta = \{NP, IV, TV, S\}$

b. <u>The Syntactic Operations:</u>
- (i) K_{Concat} = Merge (from previous notes)
- (ii) K_{Not} = Not_E (from previous notes)
- (iii) K_{And} = And_E (from previous notes)

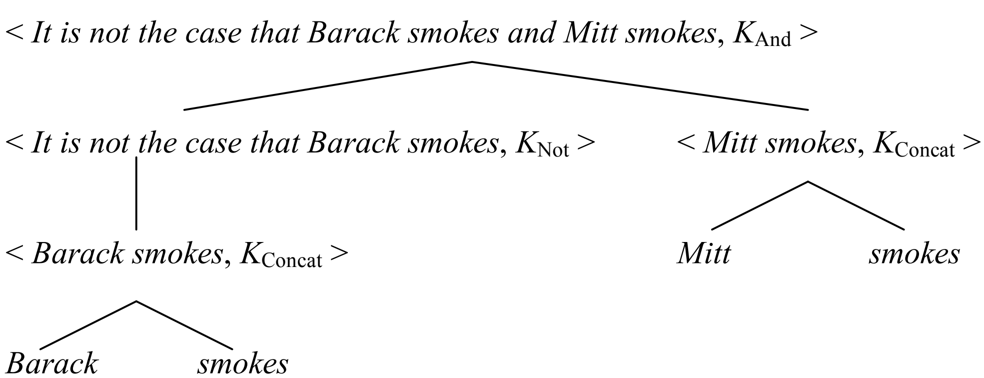
c. <u>The Basic Expressions:</u>
- (i) X_{NP} = { *Barack, Michelle, Mitt* }
- (ii) X_{IV} = { *smokes* }
- (iii) X_{TV} = { *loves* }
- (iv) X_S = $\varnothing$

c. <u>The Syntactic Algebra:</u>
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_{Concat}, K_{Not} and K_{And}

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

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

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



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):

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

↓ *h (translation from mini-Eng to Pol-NoQ)*

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

↓ *g (semantic interpretation of Politics-NoQ)*

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

g°h (the semantic interpretation of mini-Eng)

(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: < *It is not the case that Barack smokes and Mitt smokes*, K_{Not} >
|
< *Barack smokes and Mitt smokes*, K_{And} >
< *Barack smokes*, K_{Concat} > < *Mitt smokes*, K_{Concat} >
Barack *smokes* *Mitt* *smokes*

- Intuitively, we want these two different ways of constructing the mini-English sentence to lead to two different Politics-NoQ translations:
(~(**smokes' barack'**) & (**smokes' mitt'**))
~((**smokes' barack'**) & (**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!**

(22) **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*
    ≈ ~((**smokes' barack'**) & (**smokes' mitt'**))

  - *There is a reading where (19) is false*
    ≈ (~(**smokes' barack'**) & (**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 us 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…** ***hmm…***

(23) **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_{Concat} and F_{Concat} to correspond. This will get the right interpretation for VPs (IVs) after all:

  h(*loves michelle*) = *h* (K_{Concat} (*loves, Michelle*)) =
  F_{Concat}(h(*loves*), h(*Michelle*) = F_{Concat}(**loves'**, **michelle'**) = (**loves' michelle'**)

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

  h(*Barack smokes*) = *h* (K_{Concat} (*Barack, smokes*)) =
  F_{Concat}(h(*Barack*), h(*smokes*) = F_{Concat}(**barack'**, **smokes'**) = (**barack' smokes'**)

  ↑ ***Syntactic Garbage!!!***

(24) **More General Remarks**

- Obviously, what we want is for h(*Barack smokes*) = (**smokes' barack'**)
- But, there is no operation in the algebra for Politics-NoQ which will take as argument the translation of *Barack* (**barack'**) and the translation of *smokes* (**smokes'**) and return the formula (**smokes' barack'**)
- ***So maybe our homomorphic translation function h shouldn't actually be a homomorphism to the algebra*** $< \mathbf{A} , F_\gamma >_{\gamma \in \{\textbf{Concat, Not, And}\}}$
  - Maybe it should be a homomorphism to some *other* algebra that we can construct from $< A , F_\gamma >_{\gamma \in \{\text{Concat, Not, And}\}}$

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'

(25) **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 <u>*exactly one way*</u>.
- This way, we won't ever have to worry about interpreting 'syntactically ambiguous' expressions (*because they just won't exist in our language*).
- <u>Preview of Where This is Going:</u>
  We'll relate such 'syntactically unambiguous' expressions to sentence strings of English via a special operation (akin to 'linearization' or 'Spell Out').

(26) **Question**
Below we have our earlier (Montagovian) definition of a language. ***What do we have to add to this to ensure that <u>no</u> expressions are syntactically ambiguous?***

A language *L* 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_γ for all $\gamma \in \Gamma$
c. S is a set of sequences of the form $< F_\gamma , < \delta_1, \ldots, \delta_n >, \delta >$, where $\gamma \in \Gamma$, F_γ is an n-ary operation, and $\delta_1, \ldots, \delta_n, \delta \in \Delta$
b. $\delta_0 \in \Delta$

(27) **Montague's Answer**
At a minimum, we need to ensure that:

a. Nothing in the basic expressions X_δ (lexical items) can also be constructed by the syntactic operations.

<u>That is:</u> X_δ and the range of F_γ 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_γ and $F_{\gamma'}$

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

<u>That is:</u> 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_γ for all $\gamma \in \Gamma$

c. **X_δ and the range of F_γ 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_γ is an n-ary operation, and $\delta_1, \ldots, \delta_n, \delta \in \Delta$

f. $\delta_0 \in \Delta$

(29) **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 'disambiguated language' in (28), we need a more general concept of a 'language'.

(30) **Montagovian Definition of a Language (Final Version)**

A language is a pair < *L*, R >, where L = $< A , F_\gamma , X_\delta , S, \delta_0>_{\gamma \in \Gamma , \delta \in \Delta}$ is a disambiguated language, and R is a binary relation whose domain is A.

- This relation R is an 'ambiguating relation'.
  - It maps expressions in the disambiguated 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 disambiguated language being mapped to the *same* expression in the range of R (hence, the term 'ambiguating')

(31) **Illustration: Politics-NoQ-SansParens**

a. Informal Definition

(i) Vocabulary: *Same as Politics-NoQ*

(ii) The WFFs of 'Politics-NoQ-SansParens'

1. If φ is an n-ary predicate letter and each of $\alpha_1, \ldots, \alpha_n$ is either an individual constant or a variable, then $\varphi\alpha_1\ldots, \alpha_n \in$ WFF
2. If $\varphi, \psi \in$ WFF, then $\sim\varphi \in$ WFF and $\varphi \,\&\, \psi \in$ WFF

Illustrative Formulae:
~ **smokes' barack'**
loves' barack' michelle'
~ **smokes barack'** & **smokes' mitt'**

b. Formal Definition

The pair $< < A , F_\gamma , X_\tau , S , t >_{\gamma \in \{\text{Concat, Not, And}\}, \tau \in T}, R >$, where the structure $<A , F_\gamma , X_\tau , S , t >_{\gamma \in \{\text{Concat, Not, And}\}, \tau \in T}$ is Politics-NoQ, and R is the function that takes any element of A and deletes every parenthesis.

R(~((**smokes barack'**) & (**smokes' mitt'**))) =
~ **smokes barack'** & **smokes' mitt'**

(32) **Remark**
Every disambiguated language $< A , F_\gamma , X_\delta , S, \delta_0>_{\gamma \in \Gamma , \delta \in \Delta}$ 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 Γ.
c. The *category labels* of **L** is Δ
d. The *syntactic rules* of **L** is S.

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

f. The *category* δ of **L** is $\{ \varphi : \exists\psi \in C_\delta$ such that $\psi R \varphi \}$, where C_δ 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 δ of **L**.

h. The *declarative sentences* of **L** is the category δ_0 of **L**

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

j. The language **L** is *syntactically ambiguous* if there is a meaningful expression φ 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.

Mini-English ⟶ *'Disambiguated' L* ⟶ *Politics-NoQ*
Ambiguating 'R' **homomorphic translation**

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* , K_{Concat}>

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?

(38) **Step One: The Category Labels**
The syntactic categories of Disambiguated mini-English will be just the same as before:

$\Delta = \{NP, IV, TV, S\}$

(39) **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_{NP} = { < *Barack*, ∅>, < *Michelle*, ∅>, < *Mitt*, ∅> }
b. X_{IV} = { < *smokes*, ∅> }
c. X_{TV} = { < *loves*, ∅> }
d. X_{S} = ∅

(40) **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, α and β are trees whose root nodes are ordered pairs. In addition α' and β' are the first members of the root nodes of α and β (respectively).

a. $K_{Concat}(\alpha, \beta)$ = < α' β' , Concat >
(daughters: α β)

b. $K_{Not}(\alpha)$ = < *it is not the case that* α' , Not >
(daughter: α)

c. $K_{And}(\alpha, \beta)$ = < α' and β' , And >
(daughters: α β)

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

d. $K_{If}(\alpha, \beta)$ = < If α' then β' , If >
(daughters: α β)

Note:
The right-hand member of a node is now the ***index* of the operation**, rather than the operation itself…

(41) **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_{Concat}, K_{Not}, K_{And}, and K_{If}

(42) **The Syntactic Rules**
We can retain much the same set of syntactic rules S_E that we had before:

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

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

<u>Some Illustrative Members of Category C_S</u>

a. < *It is not the case that Barack smokes and Mitt smokes*, Not >
- < *Barack smokes and Mitt smokes*, And >
  - < *Barack smokes*, Concat >
    - < *Barack* , ∅ >
    - < *smokes* , ∅ >
  - < *Mitt smokes*, Concat >
    - < *Mitt*, ∅>
    - < *smokes*, ∅ >

b. < *It is not the case that Barack smokes and Mitt smokes*, And >
- < *It is not the case that Barack smokes*, Not >
  - < *Barack smokes*, Concat >
    - < *Barack* , ∅ >
    - < *smokes* , ∅ >
- < *Mitt smokes*, Concat >
  - < *Mitt*, ∅>
  - < *smokes*, ∅ >

c. < *Barack loves Michelle*, Concat >
- < *Barack* , ∅ >
- < *loves Michelle*, Concat >
  - < *loves* ∅ >
  - < *Michelle*, ∅ >

(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

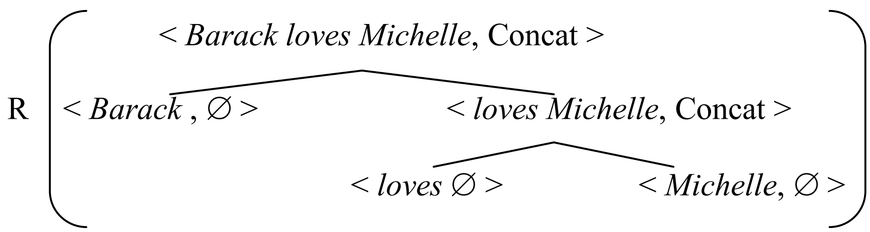
Now, we can use Disambiguated Mini-English to characterize Mini-English as a language, in the sense of (30)

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

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

a. The structure $< E, K_\gamma, X_\delta, S_E, S >_{\gamma \in \{\text{Concat, Not, And, If}\}, \delta \in \Delta}$ 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.



=

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

<u>**What Coming Up Next:**</u>

- 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**