Problem Set on Languages and Interpretations:
Answers and Notes

1. Notes on the Answers

In Section 2, I have copied some illustrative answers from the problem sets submitted to me. In this section, I provide some notes on the answers below as well as on the problems themselves.

(1) Basic Comprehension Questions on Our (Provisional) Definition of a 'Language'

- For (1c), (1g), and (1i,iii), the ideal way to represent the relevant indexed families is as $\{X_\delta\}_{\delta \in \Delta}$, $\{ F_\gamma \}_{\gamma \in \Gamma}$, and $\{C_\delta\}_{\delta \in \Delta}$, respectively.

- For (1h), an ideal answer would be something along the following lines:

  "The meaningful expressions of *L*, ME_L, is defined to be a subset of A. However, ME_L = A when ME_L is closed under the syntactic operations $\{ F_\gamma \}_{\gamma \in \Gamma}$. This can arise when there is only one syntactic category $\delta_0 \in \Delta$ and $C_{\delta 0}$ is closed under $\{F_\gamma\}_{\gamma \in \Gamma}$. This particular situation holds for PL (where δ_0 = t)."

- For (1i,iv), an acceptable answer needed to mention at least S and $\{X_\delta\}_{\delta \in \Delta}$, the rules and the basic expressions. It would also be acceptable to include any and all of the following: A, $\{ F_\gamma \}_{\gamma \in \Gamma}$, Γ, and Δ, since they all also factor into the key definition in (16) on "An Algebraic Perspective on the Syntax of First Order Logic".

  - The only thing that would have been incorrect to mention here would be δ_0, since it doesn't factor into the definition of CAT at all.

(2) Basic Exercise in Language Design

- An ideal answer to (2c) would state something along the following lines:

  "(i)-(iii) are not meaningful expressions of this language, because (1) they are not to be found in the basic expressions of the language, and (2) they are not in the range of any of the syntactic operations. Consequently, (i)-(iii) are not members of the set A. Thus, since by definition $ME_L \subseteq$ A, it follows that none of (i)-(iii) are members of ME_L."

- The fact that (i)-(iii) are not members of ME_L is, of course, a significant problem for our current syntax of English, since this entails that our syntax does not treat these sequences as constituents (which they are).
  - However, it's not an *essential* problem. For example, it is possible to construct a Montagovian analysis of English where *and* is a basic expression and combines with S's to form 'Conjunction Phrases', which can then append to Ss to form other S's. (Try it out for yourself!)

(3) **More Advanced Exercise on Language Design**

- For exercise (3a), the main trick is to see that the basic expressions of FOL can be (partly) defined as follows:

  X_{var} = { x : x is a variable in the vocabulary of FOL }
  X_e = { x : x is a constant in the vocabulary of FOL } $\cup$ X_{var}

- Under these definitions, C_{var} will contain all and only the variables of FOL, while C_e will contain *both* the individual constants *and* the variables. Consequently, it will be possible to have the following two syntactic rules:

  $< F_{Ext}$, < var, t >, t>
  $< F_{All}$, < var, t >, t>

- With these rules, the quantifier symbols $\exists$ and $\forall$ will rightly only ever be followed by variables (never individual constants or other expressions). However, since the variables are also in C_e it will also follow that atomic formulae will be able to contain variables.

- **This is the key to representing FOL – and other languages with variable binding – as (disambiguated) languages.**

(4) **Another Exercise on Language Design** (no comments)

(5) **An Exercise on Models and (Montagovian) Interpretations**

- For (5a), several folks had in their 'base step' an argument of the following sort:
  (n) $g(\Phi)(g(\alpha_1))\ldots(g(\alpha_n))$ = (by the fact that 'g' is a homomorphism)
  (n+1) $g((\ldots(\Phi\alpha_1)\ldots\alpha_n))$

- Unfortunately, this is not sufficient. In order to properly show that $g(\Phi)(g(\alpha_1))\ldots(g(\alpha_n)) = g((\ldots(\Phi\alpha_1)\ldots\alpha_n))$, one must explicitly invoke the operations F_{Concat} and G_{Concat} in the following way:

  (n) $g(\Phi)(g(\alpha_1))\ldots(g(\alpha_n))$ = (by the definition of G_{Concat})

  (n+1) $G_{Concat}(\ldots G_{Concat}(g(\Phi), g(\alpha_1)), \ldots, (g(\alpha_n))$ = (by the fact that g is a homomorphism)

  (n+2) $g(F_{Concat}(\ldots F_{Concat}(\Phi, \alpha_1), \ldots, \alpha_n)$ = (by the definition of F_{Concat})

  (n+3) $g((\ldots(\Phi\alpha_1)\ldots\alpha_n))$

- An ideal answer to (5b) would state something along the following lines:

  "In (5a), we showed that if an interpretation **B** = <B, G_γ , f>$_{\gamma \in \{\text{Concat, Not, And}\}}$ and a model $\mathcal{M}$ = <D, I> had the following relationship, then for every formula $\varphi \in C_t$, $[[\varphi]]^M = g(\varphi)$, where g is the meaning assignment based on **B**:

  (i) D = E (the set that **B** is based on)
  (ii) I = f

  In (5b), we are essentially asked to again assume that an interpretation **B** and a model $\mathcal{M}$ satisfy the properties in (i) and (ii). Consequently, it will automatically follow from our proof in (5a) that for every formula $\varphi \in C_t$, $[[\varphi]]^M = g(\varphi)$, where g is the meaning assignment based on **B**."

---

2. Illustrative Answers from Submitted Problem Sets

(1) Basic Comprehension Questions on Our (Provisional) Definition of a 'Language'

Let L be a language < A , F_γ , X_δ , S, δ_0>$_{\gamma \in \Gamma, \delta \in \Delta}$ as defined in (21) on the handout "An Algebraic Perspective on the Syntax of First Order Logic".

a. S

b. Δ

c. $\{X_\delta\}_{\delta \in \Delta}$

d. $\langle A, F_\gamma \rangle_{\gamma \in \Gamma}$

f. δ_0

g. $\{F_\gamma\}_{\gamma \in \Gamma}$

g. $ME_L \subseteq A$
$ME_L = A$ iff ME_L is closed under $\langle F_\gamma \rangle_{\gamma \in \Gamma}$, i.e. A contains no "garbage".

i.

(i) A

(ii) Δ

(iii) $\{C_\delta\}_{\delta \in \Delta}$

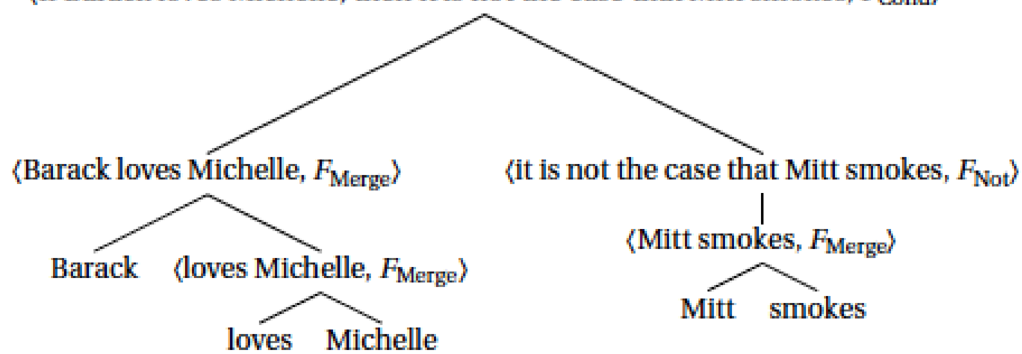
(iv) $\{X_\delta\}_{\delta \in \Delta}$ and S

(2) **Basic Exercise in Language Design**

a. Let L be defined as the language $\langle A, F_\gamma, X_\delta, S_E, S\rangle_{\gamma\in\Gamma,\delta\in\Delta}$ such that

- Γ = {Merge, And, Not, Cond}
- Δ = {NP, IV, TV, S} (as in (25) on the FOL handout)
- X_δ are as in (26) on the FOL handout
- F_γ are as in (27) on the FOL handout, with the addition of the following operation:
  - F_{Cond} is a binary operation mapping two strings 'x', 'y' to the string 'if x, then y.'
- S_E are as in (28) on the FOL handout, with the addition of the following rule:
  - $\langle F_{\text{Cond}}, \langle S, S\rangle, S\rangle$

b. ⟨If Barack loves Michelle, then it is not the case that Mitt smokes, F_{Cond}⟩



c. None of the three are meaningful expressions of L because none are lexical items and none can be generated by the syntactic rules of L.

(3) **More Advanced Exercise on Language Design**
Let First Order Logic (FOL) be the language defined in (16) on the handout "An Algebraic Perspective on Propositional Logic."

a. Let FOL be the language $\langle A, F_\gamma, X_\delta, S, \delta_0\rangle_{\gamma\in\Gamma,\delta\in\Delta}$ where the following conditions hold:

i. $\langle F_\gamma\rangle_{\gamma\in\Gamma}$ where $\Gamma = \{$Concat, Not, And, Or, If, Ext, All$\}$ contains the following operations:

- $F_{\text{Concat}}(\varphi,\psi)$ = the result of concatenating '(', φ, ψ, and ')'
- $F_{\text{Not}}(\varphi)$ = the result of concatenating '$\neg$' and φ
- $F_{\text{And}}(\varphi,\psi)$ = the result of concatenating '(', φ, '$\wedge$', ψ, and ')'
- $F_{\text{Or}}(\varphi,\psi)$ = the result of concatenating '(', φ, '$\vee$', ψ, and ')'
- $F_{\text{If}}(\varphi,\psi)$ = the result of concatenating '(', φ, '$\rightarrow$', ψ, and ')'
- $F_{\text{Ext}}(\upsilon,\varphi)$ = the result of concatenating '$\exists$', υ, and φ
- $F_{\text{All}}(\upsilon,\varphi)$ = the result of concatenating '$\forall$', υ, and φ

ii. A is the smallest set such that $\forall \delta \in \Delta, X_\delta \subseteq A$ and A is closed under the operations $\langle F_\gamma \rangle_{\gamma \in \Gamma}$.

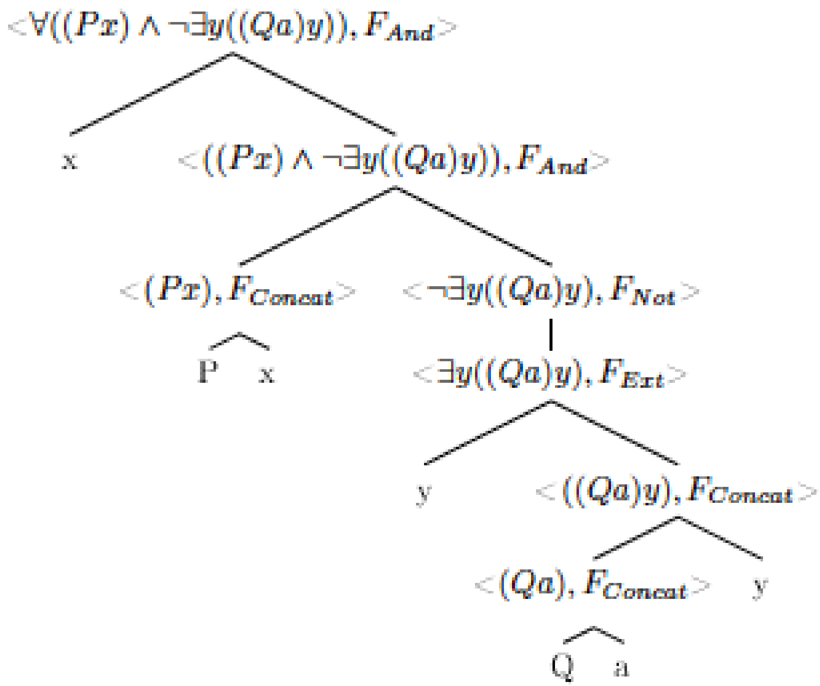
iii. $\Delta = T \cup \{\text{var}\}$ where T is the set of types and 'var' is the category label for variables.[1]

iv.
- X_{var} = the set of the variables
- X_e = the set of individual constants $\cup X_{\text{var}}$
- $X_{\langle e, \ldots, t\rangle}$ = the set of n-ary predicate letters where n = the number of times e appears in the category label $X_{\langle e, \ldots, t\rangle}$
- For all other types $\tau \in T$, $X_\tau = \emptyset$

v. S is the set of syntactic rules with the following rules:
- $\langle F_{\text{Concat}}, \langle \langle \sigma, \tau \rangle, \sigma \rangle, \tau \rangle$ where $\sigma, \tau \in T$ (class of rules)
- $\langle F_{\text{Not}}, \langle t, t \rangle, t \rangle$
- $\langle F_{\text{And}}, \langle t, t \rangle, t \rangle$
- $\langle F_{\text{Or}}, \langle t, t \rangle, t \rangle$
- $\langle F_{\text{If}}, \langle t, t \rangle, t \rangle$
- $\langle F_{\text{Ext}}, \langle \text{var}, t \rangle, t \rangle$
- $\langle F_{\text{All}}, \langle \text{var}, t \rangle, t \rangle$

vi. $\delta_0 = t$

b. Analysis tree.



c. Meaningful Expressions

1. ((Ra)x)

a. If R is a ternary predicate letter, then $R \in X_{<e, <e, <e,t>>>}$, and so $R \in C_{<e, <e, <e,t>>>}$
b. If a is an individual constant, then $a \in X_e$, and so $a \in C_e$
c. If x is a variable, then $x \in X_e$, and so $x \in C_e$
d. If R is a ternary predicate letter, and a is an individual constant and x is a variable, then $((Ra)x) \in C_{<e,t>}$

- $R \in C_{<e, <e, <e,t>>>}$
- $a \in C_e$
- $x \in C_e$
- < Concat <<e, <e <e,t>>>, e>, <e <e,t>> > is a syntactic rule
- $Concat(R,a) \in C_{<e, <e,t>>}$
- $(Ra) \in C_{<e, <e,t>>}$
- < Concat < <e <e,t>>, e>, <e,t> > is a syntactic rule
- $Concat((Ra),x) \in C_{<e,t>}$
- $((Ra)x) \in C_{<e,t>}$

2. (((Ra)x)c)

a. If R is a ternary predicate letter, then $R \in X_{<e, <e, <e,t>>>}$, and so $R \in C_{<e, <e, <e,t>>>}$
b. If a,c are individual constants, then $a,c \in X_e$, and so $a,c \in C_e$
c. If x is a variable, then $x \in X_e$, and so $x \in C_e$
d. If R is a ternary predicate letter, and a is an individual constant and x is a variable, then $(((Ra)x)c) \in C_t$

- $R \in C_{<e, <e, <e,t>>>}$
- $a,c \in C_e$
- $x \in C_e$
- < Concat <<e, <e <e,t>>>, e>, <e <e,t>> > is a syntactic rule
- $Concat(R,a) \in C_{<e, <e,t>>}$
- $(Ra) \in C_{<e, <e,t>>}$
- < Concat < <e <e,t>>, e>, <e,t> > is a syntactic rule
- $Concat((Ra),x) \in C_{<e,t>}$
- $((Ra)x) \in C_{<e,t>}$
- < Concat < <e,t>, e>, t > is a syntactic rule
- $Concat(((Ra)x),c) \in C_t$
- $(((Ra)x),c) \in C_t$

3. ∃x ((Ra)x)

This is not a meaningful expression because our rule for F_{Ext} needs something of type t; it doesn't create a meaningful expression when it is supplied with something of type <e,t>, such as ((Ra)x).

(4) **Another Exercise on Language Design**
Let Propositional Logic (PL) be the language defined in (4) on the handout "An Algebraic Perspective on Propositional Logic." Please represent PL as a language, following the definition in (21) on the handout "An Algebraic Perspective on the Syntax of First Order Logic".

(i) **The Syntactic Operations:**

- $F_{Not}(\varphi)$ = the result of concatenating $\sim$ and φ
- $F_{And}(\varphi, \psi)$ = the result of concatenating '(', φ, '&', ψ, and ')'
- $F_{Or}(\varphi, \psi)$ = the result of concatenating '(', φ, '$\vee$', ψ, and ')'
- $F_{If}(\varphi, \psi)$ = the result of concatenating '(', φ, '$\rightarrow$', ψ, and ')'

(ii) $\langle A, F_\gamma \rangle_{\gamma \in \{Not, And, Or, If\}}$ is an algebra such that

- $\{F_\gamma\}_{\gamma \in \{Not, And, Or, If\}}$ are the syntactic operations defined in (i)
- A is the smallest set such that
  1. If β is a propositional letter, then $\beta \in A$
  2. A is closed under $\{F_\gamma\}_{\gamma \in \{Not, And, Or, If\}}$

(iii) Let the syntactic category labels Δ be T, where T is the set of types

(iv) **The Basic Expressions:** $\{X_\delta\}_{\delta \in \Delta}$

X_t = The set of propositional letters

For all other types $\tau \in T$, $X_\tau = \emptyset$

(v) The set of syntactic rules S consists of the following:

- $\langle F_{Not}, \langle t \rangle, t \rangle$
- $\langle F_{And}, \langle t, t \rangle, t \rangle$
- $\langle F_{Or}, \langle t, t \rangle, t \rangle$
- $\langle F_{If}, \langle t, t \rangle, t \rangle$

$\star$ PL is the language $\langle A, F_\gamma, X_\delta, S, t \rangle_{\gamma \in \{Not, And, Or, If\} \delta \in \Delta}$ such that:

1. A is the smallest set such that:

   (1) For all $\delta \in \Delta$, $X_\delta \subseteq A$

   (2) A is closed under the operations $\{F_\gamma\}_{\gamma \in \{Not, And, Or, If\}}$

2. $\{F_\gamma\}_{\gamma \in \{Not, And, Or, If\}}$, $\{X_\delta\}_{\delta \in \Delta}$, S, and Δ are all as defined in (i)-(v)

(5) **An Exercise on Models and (Montagovian) Interpretations**

Let FOL-NoQ be the language $\langle A, F_\gamma, X_\tau, S, t\rangle_{\gamma\in\Gamma,\tau\in T}$ where $\Gamma = \{\text{Concat, Not, And}\}$ as defined in (5) on the handout 'Montague's General Theory of Semantics'.

a. Let E be a set of entities, $\mathscr{B}\ \langle B, G_\gamma, f\rangle_{\gamma\in\Gamma}$ where $\Gamma = \{\text{Concat, Not, And}\}$ be a Fregean interpretation for FOL-NoQ based on E, and g be a meaning assignment based on $\mathscr{B}$.

Let $\mathscr{M}$ be a model $\langle E, I\rangle$ where $I = f$.

Claim:
$\forall\varphi \in C_t$, $[\![\varphi]\!]^M = g(\varphi)$.

Base step: $n = 0$
If φ is an atomic formula of FOL-NoQ and contains no logical constants ($n = 0$), then φ is of the form $(\ldots(\psi\alpha_1)\ldots\alpha_m)$ where ψ is an m-ary predicate letter and each of $\alpha_1\ldots\alpha_m$ is either an individual constant or a variable.

1. $[\![\varphi]\!]^M =$ (by form of φ)
2. $[\![(\ldots(\psi\alpha_1)\ldots\alpha_m)]\!]^M =$ (by definition of F_{Concat})
3. $[\![F_{\text{Concat}}(\ldots F_{\text{Concat}}(\psi,\alpha_1),\ldots,\alpha_m)]\!]^M =$ (by definition of $[\![\,]\!]^M$)
4. $I(\psi)([\![\alpha_1]\!]^M)\ldots([\![\alpha_m]\!]^M) =$ (by definition of $[\![\,]\!]^M$)
5. $I(\psi)(I(\alpha_1))\ldots(I(\alpha_m)) =$ (by $I = f$)
6. $f(\psi)(f(\alpha_1))\ldots(f(\alpha_m)) =$ (by $f \subseteq g$)
7. $g(\psi)(g(\alpha_1))\ldots(g(\alpha_m)) =$ (by definition of G_{Concat})
8. $G_{\text{Concat}}(\ldots G_{\text{Concat}}(g(\psi), g(\alpha_1)),\ldots, g(\alpha_m))$ (by homomorphism property of g)
9. $g(F_{\text{Concat}}(\ldots F_{\text{Concat}}(\psi,\alpha_1),\ldots,\alpha_m)) =$ (by definition of F_{Concat})
10. $g((\ldots(\psi\alpha_1)\ldots\alpha_m))$ (by form of φ)
11. $g(\varphi)$

Therefore, if φ is an atomic formula of FOL-NoQ and contains no logical constants ($n = 0$), then $[\![\varphi]\!]^M = g(\varphi)$.

Induction step:
Let $n \in \mathbb{N}$ be such that for all $m < n$, if φ contains m logical constants, then $[\![\varphi]\!]^M = g(\varphi)$.

Suppose that φ contains n logical constants. There are two possible cases to consider:

i. φ is of the form $(\psi\wedge\chi)$ where ψ contains $m < n$ logical constants and χ contains $j < n$ logical constants.

1. $[\![\varphi]\!]^M = 1$ *iff* (by form of φ)
2. $[\![(\psi\wedge\chi)]\!]^M = 1$ *iff* (by definition of $[\![\,]\!]^M$)
3. $[\![\psi]\!]^M = 1$ and $[\![\chi]\!]^M = 1$

4. Therefore, given the definition of G_{And} on MGTS (7):
   $G_{\text{And}}(\llbracket\psi\rrbracket^M, \llbracket\chi\rrbracket^M) = 1$ *iff* (by induction assumption)
5. $G_{\text{And}}(g(\psi), g(\chi)) = 1$ *iff* (by homomorphism property of g)
6. $g(F_{\text{Concat}}(\psi, \chi)) = 1$ *iff* (by definition of F_{Concat})
7. $g((\psi \wedge \chi)) = 1$ *iff* (by form of φ)
8. $g(\varphi) = 1$

Therefore, $\llbracket\varphi\rrbracket^M = g(\varphi)$.

ii. φ is of the form $\neg\psi$ where ψ contains $n-1$ logical constants.

1. $\llbracket\varphi\rrbracket^M = 1$ *iff* (by form of φ)
2. $\llbracket\neg\psi\rrbracket^M = 1$ *iff* (by definition of $\llbracket\,\rrbracket^M$)
3. $\llbracket\psi\rrbracket^M = 0$
4. Therefore, given the definition of G_{Not} on MGTS (7):
   $G_{\text{Not}}(\llbracket\psi\rrbracket^M) = 1$ *iff* (by induction assumption)
5. $G_{\text{Not}}(g(\psi)) = 1$ *iff* (by homomorphism property of g)
6. $g(F_{\text{Not}}(\psi)) = 1$ *iff* (by definition of F_{Not})
7. $g(\neg\psi) = 1$ *iff* (by form of φ)
8. $g(\varphi) = 1$

Therefore, $\llbracket\varphi\rrbracket^M = g(\varphi)$.

Therefore, by strong induction, $\forall n \in \mathbb{N}$, if φ has n logical constants, then $\llbracket\varphi\rrbracket^M = g(\varphi)$.

b. Let $\mathscr{M}$ be a model $\langle D, I\rangle$. Let $\mathscr{B}$ $\langle B, G_\gamma, f\rangle_{\gamma\in\Gamma}$ where $\Gamma = \{\text{Concat, Not, And}\}$ be a Fregean interpretation for FOL-NoQ based on D where $f = I$.

The proof in (5a) also proves that $\llbracket\varphi\rrbracket^M = g(\varphi)$ with the assumptions above. The difference here is that the Fregean interpretation $\mathscr{B}$ is based on the model $\mathscr{M}$ in that the model's domain serves as the set of entities on which the Fregean interpretation is based and the model's interpretation function serves as the interpretation's function from $\bigcup_{\delta\in\Delta} X_\delta$ into B. The key equalities of $I = f$ and $D = E$, where D is the model's domain and E is the set of entities on which the Fregean interpretation is based, hold in (5a) and (5b) (and = is of course cumulative).