An Algebraic Perspective on Propositional Logic: The First Steps Towards Montague Grammar [1]

- Many of the key ideas underlying Montague's general semantic program stem from an 'algebraic' perspective on logic and its formal semantics.
- In these notes, we will begin developing such an algebraic perspective.
- **However, it will require us to first make some superficial revisions to our syntax and semantics for PL and FOL.**

1. Prolegomena, Part 1: A New Presentation of PL

Recall our formal definition of a 'WFF' in Propositional Logic, below:

(1) **The Definition of a 'Well-Formed Formula' (WFF) of PL**
The set of 'well-formed formulae' of PL, WFF, is the smallest set such that:
a. If φ is a proposition letter, then $\varphi \in$ WFF
b. If $\varphi, \psi \in$ WFF, then
1. $\sim\varphi \in$ WFF
2. $(\varphi \,\&\, \psi) \in$ WFF
3. $(\varphi \vee \psi) \in$ WFF
4. $(\varphi \rightarrow \psi) \in$ WFF

(2) **A More Precise Restatement of the Conditions in (1b)**

a. If φ is a string in the vocabulary of PL which is in the set WFF, then the following is also in WFF: **The result of concatenating the symbol '~' and the string φ**

b. If φ and ψ are strings in the vocabulary of PL which are in the set WFF, then the following is also in WFF: **The result of concatenating '(', φ, '&', ψ and ')'.**

c. If φ and ψ are strings in the vocabulary of PL which are in the set WFF, then the following is also in WFF: **The result of concatenating '(', φ, 'v', ψ and ')'.**

d. If φ and ψ are strings in the vocabulary of PL which are in the set WFF, then the following is also in WFF: **The result of concatenating '(', φ, '→', ψ and ')'.**

[1] These notes are based upon material in the following required readings: Partee *et al.* (1993) Chapter 9, Chapter 13 pp. 331-336. In addition, students are highly encouraged to begin reading Halvorsen & Ladusaw (1979) and Dowty *et al.* (1981) Chapter 8, and to start gradually working through Montague's original "Universal Grammar" paper.

Just for fun – but with big implications for later – let's introduce the 'syntactic operations' below:

(3) **Syntactic Operations Over the Vocabulary of PL**
The following operations (functions) take as input either strings of symbols over the vocabulary of PL or pairs of such strings:

a. The Operation 'Not'
Not(φ) = **The result of concatenating the symbol '~' and the string φ**

b. The Operation 'And'
And(φ,ψ) = **The result of concatenating '(', φ, '&', ψ and ')'**

c. The Operation 'Or'
Or(φ,ψ) = **The result of concatenating '(', φ, '∨', ψ and ')'**

d. The Operation 'If'
If(φ,ψ) = **The result of concatenating '(', φ, '→', ψ and ')'**

With all this in place, we can offer the following equivalent restatement of the syntax of PL:

(4) **The Definition of a 'Well-Formed Formula' (WFF) of PL**
The set of 'well-formed formulae' of PL, WFF, is the smallest set such that:

a. If φ is a proposition letter, then φ ∈ WFF

b. If φ ∈ WFF, then Not(φ) ∈ WFF

c. If φ, ψ ∈ WFF, then And(φ,ψ) ∈ WFF

d. If φ, ψ ∈ WFF, then Or(φ,ψ) ∈ WFF

e. If φ, ψ ∈ WFF, then If(φ,ψ) ∈ WFF

(5) **Vocabulary**
Let A be a set, and *f* be an n-ary function. We say that *A is* ***closed under*** *f* if

(i) Every $<a_1, \ldots, a_n> \in A^n$ is in the domain of *f*

(ii) If $<a_1, \ldots, a_n> \in A^n$, then $f(<a_1, \ldots, a_n>) \in A$

(6) **Observation**
The set WFF defined in (4) is closed under the operations Not, And, Or, and If.

(7) **Abstracting Even Further**
If we wanted to – and we will want to later on – we could schematically represent each of the 'syntactic rules' in (4b-e) as the following tuples:

b. < Not, <WFF>, WFF>
'The result of applying Not to a member of WFF is a WFF'

c. < And, <WFF, WFF>, WFF>
'The result of applying And to a pair consisting of a WFF and a WFF is a WFF'

d. <Or, <WFF, WFF>, WFF>
'The result of applying Or to a pair consisting of a WFF and a WFF is a WFF'

e. <If, <WFF, WFF>, WFF>
'The result of applying If to a pair consisting of a WFF and a WFF is a WFF'

(8) **Key Observation**

- In this way of describing the syntax of PL, we importantly distinguish between:
  (i) Syntactic *Operations*
  (ii) Syntactic *Rules*

- Syntactic Operations (Neg, And, Or, If) freely apply to any strings of symbols, and output strings that aren't necessarily part of the language we want to define.

- Syntactic Rules (7b-e) make use of the syntactic operations to define **categories** of strings in the vocabulary, *i.e.*, **syntactic categories** (e.g. 'WFF')

This will be of fundamental importance to the Montague Grammar architecture…

Now, recall our formal definition of a 'valuation' for Propositional Logic, below:

(9) **Definition of a 'Valuation' for Propositional Logic**
A valuation V is a function from the well-formed formulae of PL to the set of truth-values {1,0} (V: WFF → {0,1}) such that if $\varphi, \psi \in$ WFF, then

a. $V(\sim\varphi) = 1$ *iff* $V(\varphi) = 0$
b. $V((\varphi \,\&\, \psi)) = 1$ *iff* $V(\varphi) = 1$ and $V(\psi) = 1$
c. $V((\varphi \vee \psi)) = 1$ *iff* $V(\varphi) = 1$ or $V(\psi) = 1$
d. $V((\varphi \rightarrow \psi)) =1$ *iff* $V(\varphi) = 0$ or $V(\psi) = 1$

Again, just for fun – but with big implications for later – let's introduce the following 'semantic operations':

(10) **Semantic Operations Over {1,0}**
The following operations (functions) take as input elements of either {1,0} or $\{1,0\}^2$:

a. The Operation 'Neg'
Neg = { <1,0>, <0,1>}

b. The Operation 'Conj'
Conj = { <<1,1>,1>, <<1,0>,0>, <<0,1>,0>, <<0,0>,0> }

c. The Operation 'Disj'
Disj = { <<1,1>,1>, <<1,0>,1>, <<0,1>,1>, <<0,0>,0> }

d. The Operation 'Imp'
Imp = { <<1,1>,1>, <<1,0>,0>, <<0,1>,1>, <<0,0>,1> }

(11) **Observation:** {1,0} is closed under Neg, Conj, Disj, Imp

We can use the semantic operations in (10) and the syntactic operations in (3) to provide the following re-statement of what a valuation is…

(12) **Definition of a 'Valuation' for Propositional Logic**

A function V: WFF → {0,1} is a valuation if it satisfies the conditions below:

a. If φ = Not(ψ), then V(φ) = Neg(V(ψ))

b. If φ = And(ψ,χ), then V(φ) = Conj(V(ψ),V(χ))

c. If φ = Or(ψ,χ), then V(φ) = Disj(V(ψ),V(χ))

d. If φ = If(ψ,χ), then V(φ) = Imp(V(ψ),V(χ))

Confirm For Yourself:
The new definition in (12) entails the key conditions of our earlier definition in (9):

a. V(~φ) = 1 *iff* V(φ) = 0
b. V((φ & ψ)) = 1 *iff* V(φ) = 1 and V(ψ) = 1
c. V((φ ∨ ψ)) = 1 *iff* V(φ) = 1 or V(ψ) = 1
d. V((φ → ψ)) =1 *iff* V(φ) = 0 or V(ψ) = 1

2. Prolegomena, Part 2: A New Presentation of FOL

We'll also slightly alter our presentation of FOL in a manner similar to what we just did for PL...

- Recall our earlier definition below:

(13) **The Definition of a 'Well-Formed Formula' (WFF) in FOL**
The set of 'well-formed formulae' of FOL, WFF, is the smallest set such that:

a. 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

b. If $\varphi, \psi \in$ WFF, then
   1. $\sim\varphi \in$ WFF
   2. $(\varphi \;\&\; \psi) \in$ WFF
   3. $(\varphi \vee \psi) \in$ WFF
   4. $(\varphi \rightarrow \psi) \in$ WFF

c. If $\varphi \in$ WFF and υ is a variable, then
   1. $\forall\upsilon\varphi \in$ WFF
   2. $\exists\upsilon\varphi \in$ WFF

For our first minor change, we'll slightly alter the syntax of the atomic formulae.

(14) **Slightly Altered Syntax for Atomic Formulae**
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 $(\ldots(\varphi\alpha_1)\ldots\alpha_n) \in$ WFF

<u>Illustration:</u> Earlier Syntax: Faxby
New Syntax: ((((Fa)x)b)y)

Now, let's expand the syntactic operations in (3) by adding the following…

(15) **Syntactic Operations Over the Vocabulary of FOL**

a. <u>The Operation 'Concat'</u>
Concat(φ,ψ) = **The result of concatenating '(', φ, ψ, and ')'**

b. <u>The Operation 'All'</u>
All(φ,ψ) = **The result of concatenating '$\forall$', φ, and ψ**

c. <u>The Operation 'Ext'</u>
Ext(φ,ψ) = **The result of concatenating '$\exists$', φ, and ψ**

With these syntactic operations, we can restate our definition of WFF in FOL as follows:

(16) **The Definition of a 'Well-Formed Formula' (WFF) in FOL**

The set of 'well-formed formulae' of PL, WFF, is the smallest set such that:

a. 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 Concat(…(Concat(Concat(φ,α_1), α_2), …, α_n) $\in$ WFF

b. If $\varphi \in$ WFF, then Not(φ) $\in$ WFF

c. If $\varphi, \psi \in$ WFF, then And(φ,ψ) $\in$ WFF

d. If $\varphi, \psi \in$ WFF, then Or(φ,ψ) $\in$ WFF

e. If $\varphi, \psi \in$ WFF, then If(φ,ψ) $\in$ WFF

f. If $\varphi \in$ WFF and υ is a variable, then Ext(υ,φ) $\in$ WFF

g. If $\varphi \in$ WFF and υ is a variable, then All(υ,φ) $\in$ WFF

We could also use our schematic notation in (7) to provide the following even more compact presentation of the rules in (16b)-(16g).[2]

(17) **Abstract Representation of Rules (16b)-(16g)**

b. < Not, <WFF>, WFF>
'The result of applying Not to a member of WFF is a WFF'

c. < And, <WFF, WFF>, WFF>
'The result of applying And to a pair consisting of a WFF and a WFF is a WFF'

d. <Or, <WFF, WFF>, WFF>
'The result of applying Or to a pair consisting of a WFF and a WFF is a WFF'

e. <If, <WFF, WFF>, WFF>
'The result of applying If to a pair consisting of a WFF and a WFF is a WFF'

f. <Ext, <VAR, WFF>, WFF>
'The result of applying Ext to a pair consisting of a VAR and a WFF is a WFF'

g. <All, <VAR, WFF>, WFF>
'The result of applying All to a pair consisting of a VAR and a WFF is a WFF'

[2] Rule (16a) won't be schematically representable in this way until we've made one last change, to come later.

Given our new notation for atomic formulae and our semantic operations in (10), we'll make a slight change to our definitions of 'model' and 'valuation w.r.t. a model and variable assignment'

(18) **Definition of a 'Model' for First Order Logic**

A model $\mathcal{M}$ is a pair <D, I> consisting of:

a. A *non-empty* set D, called the 'domain of $\mathcal{M}$'

b. A function I, whose domain is the individual constants and predicate letters, and whose range satisfies the following conditions:

(i) If α is an individual constant, then $I(\alpha) \in D$
(ii) If Φ is an *n*-ary predicate letter, then $I(\Phi)$ is **the curried characteristic function of an n-ary relation** $R \subseteq D^n$

Note:
We are now interpreting predicate letters not as subsets of D or relations on D, but as **the curried characteristic functions** of such sets and relations.

(19) **Valuation of FOL, Relative to a Model and a Variable Assignment**

Let $\mathcal{M}$ be a model <D,I> and g be a variable assignment (based on $\mathcal{M}$). Then the 'valuation based on M and g' ($V_{M,g}$) is a function whose domain is the set of FOL formulae, whose range is {0,1}, and which satisfies the conditions below:

(i) If φ = Concat(…(Concat(Φ,α_1), …, α_n), then
$V_{M,g}(\varphi) = I(\Phi)([[\alpha_1]]^{M,g})\ldots([[\alpha_n]]^{M,g})$ [3]

(ii) If φ = Not(ψ), then $V_{M,g}(\varphi)$ = Neg($V_{M,g}(\psi)$)

(iii) If φ = And(ψ,χ), then $V_{M,g}(\varphi)$ = Conj($V_{M,g}(\psi)$, $V_{M,g}(\chi)$)

(iv) If φ = Or(ψ,χ), then $V_{M,g}(\varphi)$ = Disj($V_{M,g}(\psi)$, $V_{M,g}(\chi)$)

(v) If φ = If(ψ,χ), then $V_{M,g}(\varphi)$ = Imp($V_{M,g}(\psi)$, $V_{M,g}(\chi)$)

(vi) If φ = Ext(υ,φ), then $V_{M,g}(\varphi) = 1$ *iff* there is an $a \in D$ such that $V_{M,g(v/a)}(\psi) = 1$

(vii) If φ = All(υ,φ), then $V_{M,g}(\varphi) = 1$ *iff* for every $a \in D$, $V_{M,g(v/a)}(\psi) = 1$

[3] Note that if $I(\Phi)$ is the curried characteristic function of R, then this is equivalent to saying $< [[\alpha_1]]^{M,g}, \ldots, [[\alpha_n]]^{M,g}> \in R$, as in our original definition of a valuation based on *M* and g.

(20) **Illustration of the New Definition**

a. <u>Illustration of the Definition of a Model:</u>
The pair FOL < { Angelika, Seth, Rajesh }, I > is a model of FOL, where I consists of at least the following mappings: [4]

I(a) = Angelika I(b) = Seth I(c) = Rajesh

I(F) = { <Angelika, 0>, <Seth, 0>, <Rajesh,0> }

$$
I(P) = \left[\begin{array}{lcl}
\text{Angelika} & \rightarrow & \left[\begin{array}{l} \text{Angelika} \rightarrow 0 \\ \text{Rajesh} \rightarrow 1 \\ \text{Seth} \rightarrow 1 \end{array} \right] \\
\text{Rajesh} & \rightarrow & \left[\begin{array}{l} \text{Angelika} \rightarrow 0 \\ \text{Rajesh} \rightarrow 0 \\ \text{Seth} \rightarrow 1 \end{array} \right] \\
\text{Seth} & \rightarrow & \left[\begin{array}{l} \text{Angelika} \rightarrow 0 \\ \text{Rajesh} \rightarrow 0 \\ \text{Seth} \rightarrow 0 \end{array} \right]
\end{array} \right]
$$

b. <u>Illustration of the Definition of a Valuation:</u>
Let $\mathcal{M}$ be the model (partially) defined in (20a) and g be any variable assignment based on $\mathcal{M}$

- $V_{M,g}$(~((Pb)c)) = (by definition of Not and Concat)
- $V_{M,g}$(Not(Concat(Concat(P,b), c))) = (by (19ii))
- Neg($V_{M,g}$(Concat(Concat(P,b), c))) = (by (19i))
- Neg(I(P)($[[b]]^{M,g}$)($[[c]]^{M,g}$)) = (by definition of $[[.]]^{M,g}$)
- Neg(I(P)(I(b))(I(c))) = (by definition of I)
- Neg(I(P)(Seth)(Rajesh)) = (by definition of I(P))
- Neg(0) = (by definition of Neg)
- 1

[4] Note that we are again basically interpreting ‘F’ as “is French” and ‘P’ as “is older than”.

3. Algebras and Morphisms: The Key Concepts

Various advances in algebra in the 18th and 19th century lead mathematicians to develop a highly general and abstract definition of what a system of 'algebra' is.

(21) **Definition of an Algebra**

An *algebra* is a tuple $< A, f_1, \dots f_n >$ consisting of a set A together with one or more operations (functions) $f_1, \dots , f_n$, where A is *closed* under each of $f_1, \dots , f_n$.

- Note: The operations $f_1, \dots , f_n$ don't have to be of the same arity, but they must be of some finite arity.

(22) **Illustrations of Algebras**

a. $< \mathbb{N}, +, \times >$
   - The natural numbers are *closed* under addition and multiplication

Note: This is not an algebra: $< \mathbb{N}, +, \times, - >$, since $\mathbb{N}$ isn't closed under subtraction

b. $< \mathbb{Z}, +, \times, - >$
   - The integers are *closed* under addition, multiplication, and subtraction

Note: This is not an algebra: $< \mathbb{Z}, +, \times, -, \div >$, since $\mathbb{Z}$ isn't closed under division.

c. $< \mathbb{Q}, +, \times, -, \div >$
   - The rationals are *closed* under addition, multiplication, subtraction, division

d. $< WFF_{PL}$, Not, And, Or, If $>$
   - The WFF of PL are *closed* under Not, And, Or, and If (see (6))

e. $< \{1,0\}$, Neg, Conj, Disj, Imp $>$
   - The set $\{1,0\}$ is *closed* under Neg, Conj, Disj, Imp (see (11))

f. $< \{1,0\}$, Conj, Disj $>$

g. $< \{\{a\}, \varnothing\}, \cap, \cup >$
   - The set $\{\{a\}, \varnothing\}$ is closed under intersection and union

$\{a\} \cup \{a\} = \{a\}$	$\{a\} \cap \{a\} = \{a\}$
$\{a\} \cup \varnothing = \{a\}$	$\{a\} \cap \varnothing = \varnothing$
$\varnothing \cup \{a\} = \{a\}$	$\varnothing \cap \{a\} = \varnothing$
$\varnothing \cup \varnothing = \varnothing$	$\varnothing \cap \varnothing = \varnothing$

(23) **Key Observation**
The algebras < {1,0}, Conj, Disj > and < {{a}, ∅}, ∩, ∪ > are intuitively 'similar'

< {1,0}, Conj, Disj >	< {{a}, ∅}, ∩, ∪ >
Conj(1,1) = 1	∩({a},{a}) = {a}
Conj(1,0) = 0	∩({a}, ∅) = ∅
Conj(0,1) = 0	∩(∅, {a}) = ∅
Conj(0,0) = 0	∩(∅, ∅) = ∅
Disj(1,1) = 1	∪ ({a},{a}) = {a}
Disj(1,0) = 1	∪ ({a},∅) = {a}
Disj(0,1) = 1	∪ (∅, {a}) = {a}
Disj(0,0) = 0	∪ (∅, ∅) = ∅

This intuitive notion of 'similarity' is captured in the following notion of 'isomorphism'.

(24) **Isomorphic and Isomorphism**
Let **A** <A, f_1, …, f_n> and **B** <B, g_1, …, g_n> be algebras. We say that **A** and **B** are *isomorphic* if the following hold:

(i) There is a one-to-one correspondence between the operations f_1, …, f_n and g_1, …, g_n

(ii) Each f_i is of the same arity as g_i

(iii) There is a bijection h: A→ B with the following key property:

For every operation f_i in **A**, $h(f_i (a_1, \ldots a_m)) = g_i (h(a_1), \ldots h(a_n))$

If **A** and **B** are isomorphic, then the bijection h: A→B is an *isomorphism*.

Note: Informally, (iii) states that that:

If you apply bijection h to the result of f_i applied to a_1 through a_m, you get the same thing in B as you'd get if
You applied the bijection h to a_1 through a_m to get a sequence b_1 through b_m
You then applied the corresponding function g_i in **B** to b_1 through b_m

Thus, if $f_i (a_1, \ldots a_m) = a$, then $g_i (h(a_1), \ldots h(a_n)) = h(a)$

- And so we see how the isomorphism h entails that the structure of the algebra **A** is 'mirrored' in the algebra **B**

(25) **Illustration**

The algebras < {1,0}, Conj, Disj > and < {{a}, ∅}, ∩, ∪ > are isomorphic.

Demontration: Assume the correspondence Conj ~ ∩ and Disj ~ ∪
Consider the bijection h = { < 1, {a} >, < 0, ∅ > }

Checking For Conj ~ ∩

h(Conj(1,1)) = h(1) = {a} = ∩({a},{a}) = **∩(h(1), h(1))**
h(Conj(1,0)) = h(0) = ∅ = ∩({a}, ∅) = **∩(h(1), h(0))**
h(Conj(0,1)) = h(0) = ∅ = ∩(∅, {a}) = **∩(h(0), h(1))**
h(Conj(0,0)) = h(0) = ∅ = ∩(∅, ∅) = **∩(h(0), h(0))**

Checking For Disj ~ ∪

h(Disj(1,1)) = h(1) = {a} = ∪ ({a},{a}) = **∪ (h(1), h(1))**
h(Disj(1,0)) = h(1) = {a} = ∪ ({a}, ∅) = **∪ (h(1), h(0))**
h(Disj(0,1)) = h(1) = {a} = ∪ (∅, {a}) = **∪ (h(0), h(1))**
h(Disj(0,0)) = h(0) = ∅ = ∪ (∅, ∅) = **∪ (h(0), h(0))**

Thus, the function h: A→ B is an *isomorphism* from **A** to **B.**

In addition to the notion of 'isomorphism' in (24), there's also a weaker notion of 'similarity' between algebras, that of 'homomorphism'.

(26) **Homomorphic and Homomorphism**

Let **A** <A, f_1, …, f_n> and **B** <B, g_1, …, g_n> be algebras. We say that **A** and **B** are *homomorphic* if the following hold:

(i) There is a one-to-one correspondence between the operations f_1, …, f_n and g_1, …, g_n

(ii) Each f_i is of the same arity as g_i

(iii) There is a function h: A→ B with the following key property:

For every operation f_i in **A**, h(f_i (a_1, … a_m)) = g_i (h(a_1), … h(a_n))

If **A** and **B** are homomorphic, then the bijection h: A→B is an *homomorphism*.

Note:
A homomorphism differs from an isomorphism in that the former need not be an injection or a surjection; it can map two different things in A to the same thing in B, *or* there could be things in B that nothing in A gets mapped to…

(27) **Key Illustration**
The algebras < WFF_{PL} , Not, And, Or, If > and < {1,0}, Neg, Conj, Disj, Imp > are homomorphic.

Demonstration:
Assume the correspondence Not ~ Neg, And ~ Conj, Or ~ Disj, and If ~ Imp.
Let V be *any valuation*, as defined in (12).

- Note, V is a function from WFF_{PL} to {1,0}.
- Thus, we need only check the key correspondence property in (26iii)

Checking For Not ~ Neg:	$V(Not(\psi))$	=	$Neg(V(\psi))$	(by (12a))
Checking For And ~ Conj:	$V(And(\psi,\chi))$	=	$Conj(V(\psi),V(\chi))$	(by (12b))
Checking For Or ~ Disj:	$V(Or(\psi,\chi))$	=	$Disj(V(\psi),V(\chi))$	(by (12c))
Checking for If ~ Imp:	$V(If(\psi,\chi))$	=	$Imp(V(\psi),V(\chi))$	(by (12c))

---

4. Semantics and Homomorphism

(28) **Key Consequence of (27)**

A valuation V: $WFF_{PL} \rightarrow$ {1,0} is a homomorphism from < WFF_{PL} , Not, And, Or, If > ('syntactic algebra) to < {1,0}, Neg, Conj, Disj, Imp > ('semantic algebra')

One of the crucial insights of Montague was to generalize from (28) in the following way…

(29) **An 'Interpretation' of Propositional Logic (To Be Revised)**

Let the language of PL be the algebra *L* = < WFF_{PL} , Not, And, Or, If >. An *interpretation of PL* is a structure < B, f_1, f_2, f_3, f_4, h> such that:

a. < B, f_1, f_2, f_3, f_4 > is an algebra

b. h is a homomorphism from *L* to < B, f_1, f_2, f_3, f_4 >

(30) **Key Consequence:**
If V is a valuation, then < {1,0}, Neg, Conj, Disj, Imp, V > is an interpretation of PL.

However, as shown below, the notion of 'interpretation' is more general than that of 'valuation'

(31) **An Interpretation That is Not Based on Valuations**

Let h: $WFF_{PL} \rightarrow P(\{a,b,c\})$ be a function from WFF_{PL} to the powerset of {a,b,c} with the following key properties: [5]

a. $h(\sim\varphi) = h(Not(\varphi)) = h(\varphi)'$

b. $h((\varphi \,\&\, \psi)) = h(And(\varphi, \psi)) = \cap(\, h(\varphi), h(\psi)\,)$

c. $h((\varphi \vee \psi)) = h(Or(\varphi, \psi)) = \cup(\, h(\varphi), h(\psi)\,)$

d. $h((\varphi \rightarrow \psi)) = h(If(\varphi, \psi)) = IMP(\, h(\varphi), h(\psi)\,)$
- Where $IMP(A,B) = A' \cup B$

Illustration:
If h(p) = {b,c} and h(q) = {a,b}, then h((p&q)) = {b}, and h((p∨q)) = {a,b,c}

Key Consequence:
The structure < *P*({a,b,c}), ′, ∩, ∪, IMP, h > is an interpretation of PL.

- < *P*({a,b,c}), ′, ∩, ∪, IMP > is an algebra

- h is a homomorphism from *L* to < *P*({a,b,c}), ′, ∩, ∪, IMP >
  - h is a function from WFF_{PL} to *P*({a,b,c})
  - The assumptions in (31a-d) show that it has the homomorphism property (26iii)

We could view an interpretation like < *P*({a,b,c}), ′, ∩, ∪, IMP, h > as being one in which the WFFs of PL are mapped to the 'possible worlds' in {a,b,c} where they are true…

Finally, we can (try to) generalize the notion of 'interpretation' in (29) to any language…

(32) **Generalizing 'Interpretation' for Any Language (To Be Revised)**

Let $L = \langle A, f_1, \ldots, f_n \rangle$ be a 'syntactic algebra' for a given language. An *interpretation* of L is a structure $\langle B, g_1, \ldots, g_n, j \rangle$ such that:

a. $\langle B, g_1, \ldots, g_n \rangle$ is an algebra.
b. For any i, f_i and g_i have the same arity.
c. j is a function from the 'lexical items' (*e.g.* 'non-logical constants') in A to B.

If **B** = $\langle B, g_1, \ldots, g_n, j \rangle$ is an interpretation of L, then the *meaning assignment* for L determined by **B** is the unique homomorphism h from L to $\langle B, g_1, \ldots, g_n \rangle$ such that $j \subseteq h$.

[5] That such a function exists can be shown by giving a concrete example. Students familiar with modal propositional logic will be familiar with many such concrete examples.

(33) **Semantics and Homomorphism: The Intuitive Relation**

a. The Principle of Compositionality
The meaning of a complex expression is a 'function of' (i) the meanings of its component expressions and (ii) their mode of syntactic composition.

b. Key Observation:
(33a) basically says that interpretation is homomorphism from syntax to meaning

- To get the meaning of a complex expression φ, you determine the syntactic rule SynR that derives it from its component expressions ψ, χ

- Then, you compute the meanings $[[\psi]]$, $[[\chi]]$

- Finally, you input $[[\psi]]$, $[[\chi]]$ into a semantic rule SemR that 'corresponds' to the syntactic rule SynR

  - Thus, $[[\ \text{SynR}(\psi,\chi)\ ]] \quad = \quad \text{SemR}(\ [[\psi]], [[\chi]]\ )$

Syntax: *Semantics*

φ (= SynR(ψ, χ)) [[φ]] (= SemR([[ψ]], [[χ]]))

ψ χ [[ψ]] [[χ]]

(34) **Semantics and Homomorphism: Another Possible Connection**
Suppose that interpretation is homomorphism from syntax to meaning.

- It follows that if a language has a semantics, there is a 'similarity' (homomorphism) between its structure (syntactic algebra) and mind-external reality (semantic algebra)

a. Picture Theory of Meaning (Early Wittgenstein):
A meaningful expression 'pictures' a state of affairs / atomic fact.
(There is a structural similarity between the structure of the expression and the 'logical facts' of the world.)

b. Correspondence Theory of Truth (Russell, *et multia alia*)
For a statement to be true, it must have a structural isomorphism with the state of affairs that makes it true.
" 'A cat is on the mat' is true *iff* there is in the world a cat and a mat, and the cat is related to the mat in virtue of being on it. If any of the three pieces is missing… then the statement is false."

5. Moving Beyond Propositional Logic: The Key Challenges

If the definition in (32) is truly viable, then all we need to do to provide a semantics for a language *L* is the following:

- Characterize the language L as a 'syntactic algebra' $< A, f_1, \ldots, f_n >$
- Find a structure $< B, g_1, \ldots, g_n, j>$ that satisfies the key properties of being an 'interpretation' of the language.

Thus, in order for the general semantic program embodied in (32) to be viable, we must be able to characterize every (semantically interpreted) language as a syntactic algebra…

(35) **Key Problem: First Order Logic!**

- Given our presentation in (16), the key syntactic operations forming the WFFs of FOL are: Concat, Not, And, Or, If, Ext, All.
- However, the structure $< WFF_{FOL}$, Concat, Not, And, Or, If, Ext, All $>$ is *not* an algebra!
    - WFF_{FOL} is *not* closed under Concat, Ext, and All
        - Concat(Pa , ((Rb)c)) = (Pa((Rb)c)) $\notin WFF_{FOL}$
        - Ext(Pa, ((Rb)c)) = ∃Pa((Rb)c) $\notin WFF_{FOL}$
        - All(Pa, ((Rb)c)) = ∀Pa((Rb)c) $\notin WFF_{FOL}$

<u>Main Issue:</u>
In PL, applying any of the syntactic operations to a WFF creates a WFF. This just isn't the case for FOL

<u>The Solution:</u>
We'll need to augment what we mean by an 'algebraic characterization' of a language, and thus the definition in (32).

(36) **Another Challenge Raised by FOL**
Even once we resolve the issue in (35), to pursue the general program outlined in (32), **we'll need to find some 'semantic operations' that 'correspond' to Ext and All**

- Finding such operations is what logicians mean by 'algebraization of FOL'.
- Doing this is far from trivial; Montague's solution is not commonly used and difficult