Lecture 10. Statement Logic as a word algebra on the set of atomic statements. Lindenbaum algebra.

Reading: Previously distributed extract, "Boolean algebras" (pp.126-139) from Partee (1979) *Fundamentals of Mathematics for Linguists*. The part about Lindenbaum algebras (without the name) is subsection 5, pp. 133-134.

The algebraic parts of the handout are based on the PtMW textbook and two other sources:

Cohn P.M. Universal algebra, Harper and Row. New York, Evanston and London, 1965.

Burstall R.M. and J.A. Goguen, Algebras, Theories and Freeness: An Introduction for Computer Scientists. In: M. Broy and G. Schmidt (eds.) Theoretical Foundations of Programming Methodology, Reidel, 1982, pp. 329 – 349.

Note: Lecture 10 in 2004 corresponds to lectures 8 and 9 in 2001. Homework 11 in 2004 corresponds to homework 9 in 2001. [We are omitting the earlier homework 8, on congruences.]

0. Preliminary notes.

Let us return to the algebras considered in Lecture 4 (7). We considered the homomorphism f: **Mod4** $\rightarrow$ **Mod2**. Are there any homomorphisms from **Mod2** to **Mod4**? We show that there are none. Suppose that h: **Mod2** $\rightarrow$ **Mod4** is such a homomorphism. Then, by definition of a homomorphism, $h(\textbf{zero}_{\textbf{Mod2}}) = \textbf{zero}_{\textbf{Mod4}}$, i.e. $h(0) = 0$, $h(\textbf{one}_{\textbf{Mod2}}) = \textbf{one}_{\textbf{Mod4}}$, i.e. $h(1) = 1$. So far it is OK. But by the same definition $h(\textbf{one}_{\textbf{Mod2}} + \textbf{one}_{\textbf{Mod2}})$ should be equal to $h(\textbf{one}_{\textbf{Mod2}}) + h(\textbf{one}_{\textbf{Mod2}}) = \textbf{one}_{\textbf{Mod4}} + \textbf{one}_{\textbf{Mod4}} = 2$. On the other hand, in **Mod2** we have $\textbf{one}_{\textbf{Mod2}} + \textbf{one}_{\textbf{Mod2}} = 0$ and $h(0) = 0$. So we have inconsistency $h(\textbf{one}_{\textbf{Mod2}} + \textbf{one}_{\textbf{Mod2}}) = 2$ and $h(\textbf{one}_{\textbf{Mod2}} + \textbf{one}_{\textbf{Mod2}}) = 0$, i.e., $h(0) = 0$ and $h(0) = 2$. So there is no homomorphism from **Mod2** to **Mod4.**

One of the properties of these two algebras is that the same result can be obtained in many different ways. For example, in **Mod2** we have $1 + 1 = 0 + 0 = 0$. These algebras are not *free*: some nontrivial equalities hold in them. This is the reason why homomorphisms of one Ω-algebra into another are in some cases impossible. Below we will consider Ω-algebras which can be homomorphically mapped to any Ω-algebra.

1. Freedom for algebras. Word algebras, initial algebras.

1.1. Word algebras without variables.

For any signature Ω there is a particularly interesting algebra called the *word algebra* which we denote as $\mathbf{W}_\Omega$. Actually, we have considered word algebras when we have defined the syntax of Statement Logic (and Predicate Logic). Let us do it now using the algebraic terminology.

The elements of the word algebra (of its carrier) are "syntactical" expressions: they are words (strings) "built" from the symbols of the given signature as letters of the alphabet (with the addition of brackets and sometimes other punctuation symbols). The basic expressions are the 0-ary symbols; strictly speaking, they are one-letter words. And the operations produce larger expressions (larger words) from smaller ones using operation symbols and brackets. Let us see what this means with an example.

But first of all note that there are several traditions for constructing such expressions: one is the more general *prefix* form which is suitable for operations of any arity, another is the *infix* form which is more traditional in arithmetic and logic for familiar binary operations. Illustration: "$(3 + 5)$" is in infix form; "$+(3,5)$" is in prefix form. In the arithmetic and logical examples below we will use infix form for binary operations.

Example. Let us consider the signature $\Omega_{\text{Numb}} = \{\mathbf{zero}, \mathbf{one}, +, \times\}$ and the word algebra $\mathbf{W}_{\Omega\text{Numb}}$. Its carrier $W_{\Omega\text{Numb}}$ consist of words formed by symbols of the signature Ω_{Numb} considered as *letters* and brackets '(' and ')'. The carrier $W_{\Omega\text{Numb}}$ is defined by induction:

Basis of induction: The *one-letter words* **zero** and **one** belong to $W_{\Omega\text{Numb}}$.

Induction step: if the words t and s belong to $W_{\Omega\text{Numb}}$, then the words $(t + s)$ and $(t \times s)$ belong to $W_{\Omega\text{Numb}}$.

No other words belongs to $W_{\Omega\text{Numb}}$.

So the basic expressions of $W_{\Omega\text{Numb}}$ are the one-letter words **zero** and **one.** The complex expressions are produced from the simpler ones considered as operands and a symbol of operations and brackets: (**zero** + **zero**), (**zero** + **one**), (**zero** × **zero**), etc. For example, applying operation × to operands (**zero** + **zero**) and **zero** we get the word ((**zero** + **zero**) × **zero**) as the result. Expressions of word algebras are also called *terms*.

The **operations** of the word algebra $\mathbf{W}_{\Omega\text{Numb}}$ are defined in a natural way:
zero = '**zero**'
one = '**one**'
'**zero**' + '**zero**' = '(**zero** + **zero**)'
'**zero**' + '**one**' = '(**zero** + **one**)'
…
'(**zero** + **zero**)' × '**zero**' = '((**zero** + **zero**) × **zero**)', etc.

In the "table" above we use quotation signs as metasymbols to mark expressions as operands and results of operations.

The most important property of the word algebra is that a given result can be obtained in only one way; the terms themselves show us what operations must have been used to obtain them.

General case. For the signature Ω the word algebra $\mathbf{W}_\Omega$ can be defined in the same way.

The carrier W_Ω (we use the prefix form here in the definition):
1) if ω is a constant (0-ary symbol) from Ω then the one-letter word ω belongs to W_Ω.
2) if ω is an n-ary symbol from Ω and $t_1, ..., t_n$ are words from W_Ω then the word $\omega(t_1, ..., t_n)$ belongs to W_Ω.
3) There are no other elements in W_Ω (than defined by (1) and (2)).

Words from W_Ω are also called *terms* of W_Ω.

The operations:
1) every 0-ary operation ω marks the one-letter word ω of W_Ω.

2) if ω is a symbol of n-ary operation from Ω, and $t_1, ..., t_n$ are words from W_Ω, then the word $\omega(t_1, ..., t_n)$ is the result of the application of the operation ω to the n-tuple $<t_1, ..., t_n>$.

Note that in a case when a signature Ω does not contain 0-ary symbols, the word algebra described above will be empty. Below we will consider a more general notion of word algebra.

1.2. Word algebras and homomorphisms. Initial algebra.

Let us show that for any Ω_{Numb}-algebra **A** there exists a unique homomorphism f: $\mathbf{W}_{\Omega\text{Numb}} \to \mathbf{A}$. The function f can be defined by induction:

1) f maps the words **zero** and **one** from $W_{\Omega\text{Numb}}$ to the elements of the carrier A of the algebra **A** marked by corresponding constants; (for example to 0 and 1 in **Mod4**).

2) for any words t and s from $W_{\Omega\text{Numb}}$ we put $f(t + s) = f(t) + f(s)$ and $f(t \times s) = f(t) \times f(s)$.

Example. The homomorphism f: $\mathbf{W}_{\Omega\text{Numb}} \to \mathbf{Mod4}$ maps **zero** and **one** to 0 and 1. We have, for example, $f(\mathbf{zero} + \mathbf{zero}) = 0 + 0 = 0$ or $f((\mathbf{one} + \mathbf{one}) \times ((\mathbf{one} + \mathbf{one}) + \mathbf{one}))) = f(\mathbf{one} + \mathbf{one}) \times f((\mathbf{one} + \mathbf{one}) + \mathbf{one})) = = (f(\mathbf{one}) + f(\mathbf{one})) \times (f(\mathbf{one} + \mathbf{one}) + f(\mathbf{one})) = (1 + 1) \times ((f(\mathbf{one}) + f(\mathbf{one})) + 1) = (2 \times 3) = 2$.

In the general case for any signature Ω containing constants and any Ω-algebra **A** there exists a unique homomorphism f: $\mathbf{W}\Omega \to \mathbf{A}$.
Each basic expression of the word algebra (0-ary symbol of Ω) is mapped onto the corresponding element of the carrier A of **A**. Since the operations of the word algebra correspond to the operations of **A** in the obvious way, the rest of the homomorphism is determined in the same obvious way.

Definition. **I** is an *initial algebra* in signature Ω (an *initial* Ω-algebra), iff for any Ω-algebra **A** there is a unique homomorphism f: $I \to A$.

Every (non-empty) word algebra $\mathbf{W}_\Omega$ is by definition is an initial algebra.

More examples.

Consider the signature Ω_{Nt} = {**zero**, **suc**}, where **zero** is a 0-ary symbol and **suc** is a unary one.

Consider four Ω_{Nt} -algebras: **Nt**, **Mod3**, **NtNt** and the word algebra **W**$_{\Omega Nt}$:

Nt	Nt = {0, 1, 2, 3, 4, ...}	**zero** = 0
		suc(0) = 1, **suc**(1) = 2, **suc**(2) = 3, etc.
Mod3	Mod3 = {0, 1, 2}	**zero** = 0
		suc(0) = 1, **suc**(1) = 2, **suc**(2) = 0
NtNt	NtNt = {0', 1', 2',...	**zero** = 0
	0'', 1'', 2'', ...}	**suc**(0') = 1', **suc**(1') = 2', ...
		suc(0'') = 1'', **suc**(1'') = 2'',....
W$_{\Omega Nt}$	W$_{\Omega Nt}$ = {'zero', 'suc(zero)', 'suc(suc(zero))', 'suc(suc(suc(zero)))', ... }	
	Operations:	**zero** = 'zero'
		suc('zero') = 'suc(zero)'
		suc('suc(zero)') = 'suc(suc(zero))', etc.

> [Note: often the quotation marks are omitted on the elements of W$_{\Omega Nt}$; the context is sufficient to resolve the potential ambiguity.]

It is easy to see that the word algebra **W**$_{\Omega Nt}$ is an initial algebra.

Is **NtNt** an initial algebra? Let us consider its homomorphisms to **Nt.** In every homomorphism **zero** should go to **zero**, so 0' goes to 0, 1' to 1, etc. But where should we map 0''? It can go to 0, or to 1, or to 2, or to any other element . Giving a value for 0'' will determine the value for 1'', 2'', etc. So there are many possible homomorphisms from **NtNt** to **Nt**. Since there is not a unique homomorphism from **NtNt** to **Nt**, **NtNt** cannot be an initial algebra.

And it is easy to show that there are no homomorphisms from **Mod3** to **Nt**. [Show it]. So **Mod3** also is not an initial algebra.

But we can show that **Nt** is an initial algebra. Every element of its carrier is a value of some term of the word algebra **W**$_{\Omega Nt}$. On the other hand, different terms have different values. So there exists an isomorphism i: **Nt** $\rightarrow$ **W**$_{\Omega Nt}$. For any Ω_{Nt} –algebra **A** the unique homomorphism h: **Nt** $\rightarrow$ **A** is defined as a composition $h = f \circ i$ where f: **W**$_{\Omega Nt}$ $\rightarrow$ **A** is the unique homomorphism of the word algebra **W**$_{\Omega Nt}$ to **A.**

Theorem. If **I** and **I'** are both initial Ω-algebras then they are isomorphic.

2. Word algebra on a set. Free algebra.

2.1. Word algebra on a set of variables.

Here we consider a slightly more general concept: the word algebra in the signature Ω on a given set X of variables. We denote such an algebra by $\mathbf{W}_\Omega(X)$.

Given a signature Ω and the set of variables X, let us define the set Term of *terms* over Ω and X. The definition will be recursive:

1. Every variable $x \in X$ is a term.

2. If $t_1, ..., t_n$ are terms and $\omega \in \Omega(n)$, then $\omega(t_1, ..., t_n)$ is a term.

Note that by our definition 0-ary symbols from Ω (if any) are terms (by the second rule, when $n = 0$). So all the terms which we considered earlier (terms without variables) remain terms in the new definition. But now we have also terms with variables.

Let us again consider terms as words in the alphabet. This time the alphabet consists of variables from X, operators from Ω, and brackets and comma, considered as symbols. The set Term of terms (considered as words) will be the carrier of the algebra $\mathbf{W}\Omega(X)$.

We define operations on the set Term of such words in a natural way:

If $t_1, ..., t_n$ are terms and $\omega \in \Omega(n)$, then the term $\omega(t_1, ..., t_n)$ is the result of the application of the operation ω to the n-tuple $<t_1, ..., t_n>$.
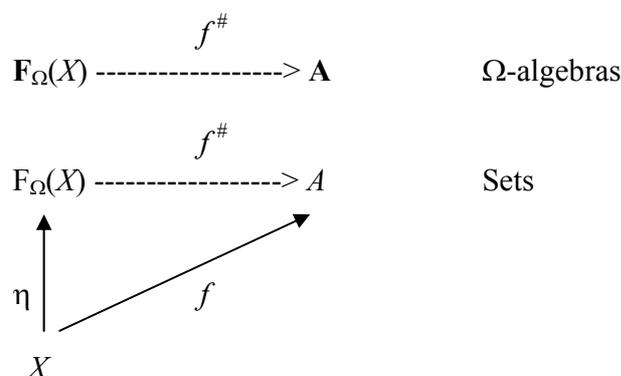
2.2. Homomorphisms from a word algebra on a set. Free algebra.

It is not difficult to show that for any Ω-algebra $\mathbf{A}$ we have many homomorphisms from $\mathbf{W}_\Omega(X)$ to $\mathbf{A}$. To get such a homomorphism we should map in some way the set of variables X to the carrier A of $\mathbf{A}$. And we will get our homomorphism as the extension of this mapping. But we first define the abstract notion of *free* algebra with the help of this property of homomorphisms.

Definition. By a *free Ω-algebra on a set of variables X* we mean an Ω-algebra $\mathbf{F}_\Omega(X)$ with a function $\eta \colon X \to \mathbf{F}_\Omega(X)$, such that for any Ω-algebra $\mathbf{A}$ any function $f \colon X \to A$ extends uniquely to a homomorphism $f^{\#} \colon \mathbf{F}_\Omega(X) \to \mathbf{A}$. By "$f^{\#}$ extends f" we mean that $f^{\#} \circ \eta = f$.

More intuitively, if $f^{\#}$ extends f, then $f^{\#}$ is identical to f wherever f is defined. And in the simplest case, η is something very close to an identity mapping (see below.)

We can illustrate the definition of free algebra $\mathbf{F}_\Omega(X)$ on a set X by a diagram:

$$f^{\#}$$
$$\mathbf{F}_\Omega(X) \text{----------------->} \mathbf{A} \qquad\qquad \Omega\text{-algebras}$$

$$f^{\#}$$
$$\mathrm{F}_\Omega(X) \text{----------------->} A \qquad\qquad \text{Sets}$$

$$\eta \Big\uparrow \qquad\qquad f$$

$$X$$

It is easy to see that the word algebra $\mathbf{W}_\Omega(X)$ is a free algebra on X. In this case the function η is the obvious one: it maps every variable x in X to the one-letter word 'x' in Term. This is almost an identity mapping, except for the fact that x in X is just an element of a set, whereas the one-letter word 'x' in the word algebra is an element of the carrier of an algebra – they look like the same thing, but they have different ontological status.

Example. Consider the word algebra $\mathbf{W}_{\Omega\ \text{Numb}}(X)$ in the signature Ω_Numb on the set of variables X = $\{x,y\}$. In this case the set Term consists of terms x, y, **zero**, **one**, (**zero** + **zero**), (x + **zero**),…, ($y \times$ (**one** + x)), etc. (Strictly speaking we should use prefix form for terms but we will be flexible for uniformity with previous examples and traditions of arithmetic).

Let us consider the Ω_Numb-algebra **Mod4** and homomorphisms from $\mathbf{W}_{\Omega\ \text{Numb}}(X)$ to **Mod4**. Every such homomorphism is defined by a function $f\colon X \to \text{Mod4}$ (Mod4 = $\{0, 1, 2, 3\}$) i.e. an evaluation of variables from X in the set Mod4 (i.e. in $\{0, 1, 2, 3\}$). And $f^{\#}$ is an evaluation function giving a value to every term from $\mathbf{W}_{\Omega\ \text{Numb}}(X)$. Naturally, $f^{\#}$ ('x') = $f(x)$, i.e., $f^{\#} \circ \eta = f$.

If, for example, $f(x) = 3$ and $f(y) = 2$ then $f^{\#}(y \times (\text{\textbf{one}} + x)) = f^{\#}(y) \times f^{\#}(\text{\textbf{one}} + x) = f(y) \times (f^{\#}(\text{\textbf{one}}) + f^{\#}(x)) = 2 \times (1 + f(x)) = 2 \times (1 + 3) = 2 \times 0 = 0$.

3. Statement Logic as a word algebra on the set of atomic statements.

In lecture 5 we considered the set of formulas of SL with the set *Atom* = $\{p, q, ...\}$ of atomic statements as the algebra **Form** in the signature Ω_BA = $\{0, 1, \neg, \vee, \wedge\}$. Now we can see that **Form** is the word algebra $\mathbf{W}_{\Omega\text{BA}}(Atom)$ where the set of atomic formulas *Atom* = $\{p, q, ...\}$ is the set of variables of this word algebra..

We considered also the Ω_BA-algebra **T** on the set $\{0,1\}$ of truth values. We described the semantics of Statement logic as a homomorphism $\sigma :$ **Form** $\to$ **T.**

Now we can consider this homomorphism as an example of the general construction of a homomorphism from a word algebra to some other algebra. For any mapping $f\colon$ Atom $\to \{0,1\}$ evaluating atomic statements we have the homomorphism of the word algebra $f^{\#}$: $\mathbf{W}_{\Omega\text{BA}}(Atom) \to$ **T**, extending the mapping f, which gives us the truth values of formulas (for a given evaluation of atomic statements by the evaluation function f).

So the semantics of SL is a family of homomorphisms of this kind.

4. Lindenbaum algebra.

The word algebra $\mathbf{W}_{\Omega\text{BA}}(Atom)$ in the signature Ω_BA = $\{0, 1, \neg, \vee, \wedge\}$ is a free algebra, so it is not a Boolean algebra. (You showed this in Homework 6, when you showed that **Form** is not a Boolean algebra.) But we can "transform" it into a Boolean algebra by considering the equivalence relations on it and constructing appropriate equivalence classes to serve as the elements of the carrier of a Boolean algebra.

In Lecture 5 we considered "logical equivalence" on the set of formulas, i.e. on the set of Term = $\mathbf{W}_{\Omega\text{BA}}(Atom)$: formulas φ and ψ are *logically equivalent* if they have the same truth value for any possible assignment of truth values (evaluations) to their atomic parts. We denoted logical equivalence between formulas φ and ψ as $\varphi \Leftrightarrow \psi$.

We considered also truth tables for formulas. We know that we can view the truth table for formulas φ and ψ containing the same atomic statements (variables) as functions of the type $D_t{}^n \rightarrow D_t$ where n is the number of variables in these formulas. Formulas φ and ψ are logically equivalent iff their truth tables represent the same functions. (If formulas contain different variables we can "add" variables to the truth tables to "equalize" the sets of variables in the tables).

It is easy to see that logical equivalence on formulas is a equivalence relation on the set of all formulas, i.e. on the carrier $W_{\Omega BA}$ (*Atom*) of the word algebra $\mathbf{W}_{\Omega BA}(Atom)$. Let us denote this equivalence by $\Leftrightarrow$. We can show that the equivalence $\Leftrightarrow$ is a congruence on this algebra. It agrees with the operations of the signature Ω_{BA}. Really, it is easy to verify that if $\varphi_1 \Leftrightarrow \psi_1$ and $\varphi_2 \Leftrightarrow \psi_2$, then $\neg\varphi_1 \Leftrightarrow \neg\psi_1$, $(\varphi_1 \vee \varphi_2) \Leftrightarrow (\psi_1 \vee \psi_2)$, etc.

Consider the quotient algebra $\mathbf{W}_{\Omega BA}(Atom)/\Leftrightarrow$ by this congruence. As we know (see Lecture 4) the carrier of the quotient algebra is the set of equivalence classes of the set $W_{\Omega BA}(Atom)$. The operations on these classes are defined in a natural way: $\neg[[\varphi]] = [[\neg\varphi]]$, $[[\varphi]] \vee [[\psi]] = [[\varphi \vee \psi]]$. [See for details "Extract from Partee (1979)"]

We can define the congruence $\Leftrightarrow$ in another, more algebraic, way. As we saw in Section 2, every mapping f: Atom $\rightarrow \{0,1\}$ evaluating atomic statements defines the homomorphism $f^{\#}$: $\mathbf{W}_{\Omega BA}(Atom) \rightarrow \mathbf{T}$ of the word algebra $\mathbf{W}_{\Omega BA}(Atom)$ onto the algebra $\mathbf{T}$. The homomorphism $f^{\#}$ gives us the truth values of formulas for the evaluation of atomic statements given by evaluation function f. The homomorphism $f^{\#}$: $\mathbf{W}_{\Omega BA}(Atom) \rightarrow \mathbf{T}$ defines a congruence on the algebra $\mathbf{W}_{\Omega BA}(Atom)$: two formulas are equivalent iff they have the same truth value for mapping f. Let us consider all such mappings, i.e. all possible evaluations of the set Atom of our variables, and all congruences given by these evaluations. Every congruence $\mathbf{Ker}\,f$ given by the evaluation f is a set of ordered pairs of equivalent formulas. Consider the intersection $\cap_f \mathbf{Ker}\,f$ of all these congruences. Intersection of congruences on an algebra is a congruence on this algebra. It is not very difficult to show that for the congruence $\Leftrightarrow$ defined above we have $\Leftrightarrow = \cap_f \mathbf{Ker}\,f$. Really, equivalence classes by the congruence $\cap_f \mathbf{Ker}\,f$ contain formulas which are equivalent for every evaluation f. [Exercise: Go through this argument with an example, using a version of statement logic with just two sentential constants *p* and *q*. You should end up with 16 equivalence classes.]

Homework 11.

1. [In 2001, this question started by saying: "Go back to exercises 1 and 2 of homework 8. If you did those successfully, you convinced yourself that the equivalence relation $\equiv_{\text{Mod } n}$ "equality modulo *n*" is a congruence on **Nat**, for any *n*. Review the argument." In 2004, we are not assigning what was then homework 8. Instead, we leave this problem 1 here as the most challenging problem of this homework; if you want to try it, begin by studying what is on the website of 2001 as Homework 8 and the student solutions posted there.]

a. Now construct the quotient algebra **Nat**/$\equiv_{\text{Mod } n}$ corresponding to this congruence. (Pick an *n*, e.g. "equality modulo 4", as in exercise 1 of homework 8.2.)

b. Show that the quotient algebra **Nat**/$\equiv_{\text{Mod4}}$ is isomorphic to the algebra **Mod4.**

2. Do the exercise suggested at the very end of this handout. Namely, go carefully through the argument that logical equivalence is a congruence on the algebra $\mathbf{W}_{\Omega BA}$(*Atom*), using as an example a version of statement logic with just two sentential constants *p* and *q*. You should end up with 16 equivalence classes.

3. In the handout from Partee (1979) on Boolean algebras, there is a section showing that we can make systems of Venn diagrams into Boolean algebras if we define things carefully. As it's done there, there's a different Boolean algebra for each configuration of circles, with the various elements of a given algebra corresponding to the different possible shadings of the diagrams. Let's consider, for instance, the Venn diagrams with two overlapping circles (as on p. 130 and on 132 of that handout), and let's construct an algebra called **Venn16** (the reason for this name will be seen in one of the questions below.) We can also consider algebras **Venn4** with just one circle**, Venn256** with three overlapping circles drawn as on p. 131, and others corresponding to how many circles we draw and how they do or don't overlap; they are also discussed in that handout. And let's apply some of the things we've been learning to **Venn16** and its cousins. Some of the questions below are very simple and are just a review of terminology; some of them will take some work. You can do any subset of them.

3.0. **Venn16** is an algebra on what signature?

3.1. What is the carrier Venn16 of **Venn16**? What is its cardinality, i.e. how many members does it contain? What are the operations of **Venn16**? (Illustrate them.)

3.2. Does **Venn16** have any subalgebras? If yes, show at least one, and show that it is a subalgebra.

3.3. If you can find a subalgebra of **Venn16,** can you find a homomorphism from **Venn16** to a subalgebra?

3.4. There should be a unique homomorphism from the word algebra $\mathbf{W}_{\Omega Bool}$ to **Venn16**, since $\mathbf{W}_{\Omega Bool}$ is an initial algebra. Find it. [Or is this false because **Venn16** involves some "variables" and is not just built from a zero and a one?]

3.5. Find a homomorphism from **Venn16** to **Venn4**, and give the kernel *ker* of that homomorphism.

3.6. Building on 3.5, specify the quotient algebra **Venn16**/*ker*. Can you name two other Boolean algebras to which **Venn16**/*ker* is isomorphic, one of them in the **Venn** family of algebras and one of them a power set Boolean algebra?

3.7. ... Illustrate more of the algebraic concepts from the past lectures in this **Venn** family and in relations of these algebras to other Boolean algebras.

4. Show the correspondence of Venn16 and Lindenbaum Algebra on Statement Logic with just {p,q}.

5. Review discussion of Venn algebras (in old Partee book) with 2^1, 2^3, 2^5, ... Can you find any way to modify the Lindenbaum Algebras to come up with corresponding algebras with equivalent classes of formulas?