

Lecture 9. Statement Logic as a word algebra on the set of atomic statements. Lindenbaum algebra.

1. Statement Logic as a word algebra on the set of atomic statements.....	1
2. Lindenbaum algebra.....	1
Homework 9.....	2

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.

1. 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 $Atom = \{p, q, \dots\}$ of atomic statements as the algebra **Form** in the signature $\Omega_{BA} = \{0, 1, \neg, \vee, \wedge\}$. Now we can see that **Form** is the word algebra $\mathbf{W}_{\Omega_{BA}}(Atom)$ where the set of atomic formulas $Atom = \{p, q, \dots\}$ 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 $\sigma: \mathbf{Form} \rightarrow \mathbf{T}$.

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 $f: Atom \rightarrow \{0,1\}$ evaluating atomic statements we have the homomorphism of the word algebra $f^\#: \mathbf{W}_{\Omega_{BA}}(Atom) \rightarrow \mathbf{T}$, 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.

2. Lindenbaum algebra.

The word algebra $\mathbf{W}_{\Omega_{BA}}(Atom)$ in the signature $\Omega_{BA} = \{0, 1, \neg, \vee, \wedge\}$ is a free algebra, so it 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 $Term = \mathbf{W}_{\Omega_{BA}}(Atom)$: 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 $\phi \Leftrightarrow \psi$.

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 $\mathbf{W}_{\Omega_{BA}}(Atom)$ 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 $\phi_1 \Leftrightarrow \psi_1$ and $\phi_2 \Leftrightarrow \psi_2$, then $\neg\phi_1 \Leftrightarrow \neg\psi_1$, $(\phi_1 \vee \phi_2) \Leftrightarrow (\psi_1 \vee \psi_2)$.

Consider the quotient algebra $\mathbf{W}_{\Omega_{BA}}(Atom)/\Leftrightarrow$ by this congruence. As we know (see Lecture 7) the carrier of the quotient algebra is the set of equivalence classes of the set $\mathbf{W}_{\Omega_{BA}}(Atom)$. The operations on these classes are defined in a natural way: $\neg[[\phi]] = [[\neg\phi]]$, $[[\phi]] \vee [[\psi]] = [[\phi \vee \psi]]$.

[See for details “Extract from Partee (1979)”]

We can define the congruence \Leftrightarrow in another, more algebraic, way. As we saw in Section 1, 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 $\bigcap_f \mathbf{Ker} f$ 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 $\Leftrightarrow = \bigcap_f \mathbf{Ker} f$. Really, equivalence classes by the congruence $\bigcap_f \mathbf{Ker} f$ 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 eight equivalence classes.]

Homework 9.

1. Go back to exercises 1 and 2 of homework 8.2. If you did those successfully, you convinced yourself that the equivalence relation $\equiv_{\text{Mod } n}$ “equality modulo n ” is a congruence on \mathbf{Nat} , for any n . Review the argument.

a. Now construct the quotient algebra $\mathbf{Nat}/\equiv_{\text{Mod } n}$ 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 $\mathbf{Nat}/\equiv_{\text{Mod } 4}$ is isomorphic to the algebra $\mathbf{Mod}4$.

2. Do the exercise suggested at the very end of this handout.