Semantics

We are given (*D*,*), where *D* is a non-empty set in an interpretation of *L*, and * is an assignment with respect to *D*. We understand α to be an arbitrary well formed formula of *L*. Suppose that *s* determines the truth value for α, i.e. *s* satisfies α if true and does not satisfy α if false. Further, suppose that "α is true on (for, in)(*D,* *)" or "α is false on (for, in)(*D*, *)". Lastly, α is logically true or α is logically false.

If we have truth functional logic, we can then define satisfaction using truth functions. *L* does in fact meet this requirement, as α is either logically true or false and this interpretation of α in *L* satisfies the formula (for, in)(*D*,*). Satisfaction in this case means that *s* is able to show truth value of the wff using a sequence of objects (in this instance a complete formula) from the domain *D* of our interpretation. Essentially, our interpretation of the semantic system allows us to derive models about the "real world"; here *L* is a simple boolean system (or so appears at first glance). We interpret the terms α as members of the domain *D*, and then the formula "α is true on (for, in)(*D,* *)" has an actual truth value.

All the above is a complicated series of statements that attempt to define how the logic system *L* supplied is implemented. If I were to simplify the above remarks, I would begin by stating that that Satisfaction, Interpretation and Logical Truth are all inter-related and necessary for a system of logic to be complete and usable. Satisfaction in the sense that we're using it here means that given a set of “things” (*D* is never defined to be anything other than a non-empty set, which is perfectly acceptable since a complete system of logic presupposes nothing about the type of data we're working with) coupled with a function that assigns values to those things should give us truth values for each assignment, and the assignments are “satisfied” if true and not satisfied if false. This concept is predicated upon the interpretation of the wffs we use in the system. Interpretation is just the meaning we understand certain wffs to have when applied to the system *L*. This meaning can be, and usually is, something simple, such as “the value of this wff in the system is true”. We don't necessarily have to know (as shown earlier) exactly what the wff “really” means to use the system and derive valid conclusions. We do, however, must know the logical truth values behind the wffs in order to determine the end result using the system. Supposing α was true, for instance, we would then interpret this truth value using *L* (assign it a value in the domain) using function *s* to come up with this truth value, which in turn would show if α is satisfied.

An example of how the above system works would go something like this: We are given a statement that is a wff in *L*. The meaning of the statement α is unknown or uncared about at this time, but we need to know if it is satisfied in *L*. We determine if it is satisfied by using function *s* to satisfy α. We then interpret α in the system *L*, mapping some value from the domain (as I stated earlier, *L* appears to be a boolean system, which means only T and F are in the domain) to it. In this case I'll map T to α as that seems appropriate given the way *L* was described. This mapping now has a truth value itself, and in this case it is true. Finally, we interpret the mapping and truth value we derived to come up with a statement that has an actual truth value for the meaning of α. Perhaps α was something like “2+2=4”. We can now interpret this statement to mean “it is true that 2+2=4e” since we have come up with a logical truth about it.

This essay has a sort of rambling behavior, but I think the key concepts are made clear. Semantics is the interpretation of symbols, the important part of logic. While we can create systems that only tell us things about those particular symbols which are defined within itself, this tells us nothing useful. “2+2=4” has a meaning syntactically for certain, and in fact we can do mathematics all day without ever understanding what 2 even is. But it is only when we understand the symbol 2 to mean “two objects” that math has any real use to us, and the same goes for logic. The interpretation of the symbols themselves in an application to the “Real World” is what makes logic a useful tool in determining things. Computing, for instance, would be non-existent without logic, but it is only in the semantic interpretation of $A \cup B$ that we can come up with a circuit that must have both A and B pathways on to be true. I'm inclined to say that purely syntactical approaches are rather boring. An example of this would be the program I included to do the second section of this test. It essentially produces lists of numbers. How exciting! It is only after I have used semantics, mapping those numbers to cake positions that the witches moved from and to, does it have any value to me. No longer do I simply see a series like [1,2,4,5,3], but instead see, “Hey, witch 3 moved to cake 5.” That is the importance of semantic representations, and the whole purpose to do logic.