COMP 409, Lecture 3

January 23, 2002

1 Semantics

The lecture started with stating that there are two aspects or levels of language (or logic) - *syntax* and *semantics*. While syntax deals with the form or structure of the language, it is semantics that adds meaning to the form.

The *semantical domain* (also know as *domain of discourse*) is the set 2^{Prop} where *Prop* is the set of all atomic propositions. *Semantics* is the relation between $Form$, the set of all formulae and 2^{Prop}. We call this relation *satisfaction*, represented by $\models$, where $\models \subseteq 2^{Prop} \times Form$.

Earlier the focus was on the issue of logical syntax and the Unique Readability Theorem was proved. This lecture describes 3 different views of semantics and shows that they are equivalent. In propositional logic there are only two values (truth values) *true* and *false* or *1* and *0*. Formulas represent the syntax. What are the structures to which these formulas refer?

Definition 1 *A truth assignment, τ, is an element of* 2^{Prop}.

There are two ways to think of truth assignments:

1. 2^{Prop} can be thought of as the power set of all propositions, and a truth assignment X is an element of it, that is, $X \subseteq$ *Prop*.

2. Or, we can think of 2^{Prop} as the set of all functions from $Prop$ to the set $\{0,1\}$. Then, a truth assignment is a map τ : *Prop* $\mapsto \{0,1\}$.

These notions are equivalent since given $X \subseteq$ *Prop*, we can define τ_X : *Prop* $\mapsto \{0,1\}$ by

$$\tau_X(p) = \begin{cases} 1 & \text{if } p \in X \\ 0 & \text{if } p \notin X \end{cases}$$

which is the characteristic function of X. Conversely, given $\tau : Prop \mapsto \{0,1\}$, we can take $X_\tau = \{p|\tau(p) = 1\}$.

Let's consider now three different perspectives of semantics:

1. philosopher's view
2. electrical engineer view
3. software engineer view

2 Philosopher's view

For a philosopher, semantics is a binary relation $\models$ between structures and formulas. $\tau \models \varphi$ means:

1. τ satisfies φ or
2. τ is true of φ or
3. φ holds at τ or
4. τ is a model of φ.

Example:
object τ: table
formular φ: this table is round
$\tau \models \varphi$ can be either true of false

Definition 2 $\models \subseteq (2^{Prop} \times Form)$ *is a binary relation, where the left side has a truth assignment and the right side has a formula.* $\models$ *is called the* satisfaction relation, semantical relation, *or the* truth relation. *We shall define it inductively:*

- $\tau \models p$ for some proposition p if $\tau(p) = 1$, meaning that p holds if p is true
- $\tau \models \neg\varphi$ if it is not the case that $\tau \models \varphi$, that is, if $\tau \not\models \varphi$ (Note: this is so only in 2-valued world)
- $\tau \models \varphi \vee \psi$ if $\tau \models \varphi$ or $\tau \models \psi$
- $\tau \models \varphi \wedge \psi$ if $\tau \models \varphi$ and $\tau \models \psi$
- $\tau \models \varphi \rightarrow \psi$ if $\tau \models \varphi$ implies $\tau \models \psi$, i.e. if $\tau \not\models \varphi$ or $\tau \models \psi$
- $\tau \models \varphi \leftrightarrow \psi$ if $\tau \models \varphi$ iff $\tau \models \psi$

It is important to realize that the semantics is well defined because of the Unique Readability Theorem. Consider for example the clauses:

- $\tau \models (\varphi_1 \wedge \varphi_2)$ iff $\tau \models \varphi_1$ and $\tau \models \varphi_2$
- $\tau \models (\varphi_1 \vee \varphi_2)$ iff $\tau \models \varphi_1$ or $\tau \models \varphi_2$

These clauses rely on the fact that, given a formula, we can talk about its primary connective and its immediate subformula(s) without ambiguity. Without unique readability, we would not have well-defined semantics. Suppose for example that we do not require parentheses, and consider the "formula" $p \wedge q \vee r$.

Let τ be a truth assignment such that $\tau(p) = 0$, $\tau(q) = 1$, and $\tau(r) = 1$. Since the example formula does not have a unique primary connective, its meaning may be defined in the following two ways:

- $\tau \models p \wedge q \vee r$ iff $\tau \models p$ and $\tau \models q \vee r$. Since $\tau \not\models p$, we have that $\tau \not\models p \wedge q \vee r$.
- $\tau \models p \wedge q \vee r$ iff $\tau \models p \wedge q$ and $\tau \models r$. Since $\tau \models r$, $\tau \models p \wedge q \vee r$.

According to the above example, a formula could have two different truth values if it did not have a unique primary connective.

A similar kind of ambiguity exists also in natural languages. Consider the following English phrase: "John hates Jim and he likes Jill". Who is he? It is ambiguous.

This points out an important feature of this definition, the *compositionality* of $\models$. An example is $\tau \models (\varphi \wedge \psi)$. All that is needed is information about $\tau \models \varphi$ and $\tau \models \psi$ and those answers are independent of each other. This is alot like modularity in programming. An older term for this is *referential transparency.*

3 Electrical Engineer view

To an electrical engineer the truth assignment is simply a mapping of voltages on a wire: $\tau : Prop \rightarrow \{0,1\}$

Operations are carried out by gates, which represent logical connectives. A k-ary gate g is a mapping $g : \{0,1\}^k \rightarrow \{0,1\}$. Here are some examples:

- $\neg : \{0,1\} \rightarrow \{0,1\}$
  - $\neg(0) = 1$ and $\neg(1) = 0$
- $\wedge : \{0,1\}^2 \rightarrow \{0,1\}$
  - $\wedge(0,0) = \wedge(1,0) = \wedge(0,1) = 0$ and $\wedge(1,1) = 1$
- $\vee : \{0,1\}^2 \rightarrow \{0,1\}$
  - $\vee(1,1) = \vee(1,0) = \vee(0,1) = 1$ and $\vee(0,0) = 0$
- $\rightarrow: \{0,1\}^2 \rightarrow \{0,1\}$
  - $\rightarrow(1,1) = \rightarrow(0,0) = \rightarrow(0,1) = 1$ and $\rightarrow(1,0) = 0$
- $\leftrightarrow: \{0,1\}^2 \rightarrow \{0,1\}$
  - $\leftrightarrow(1,1) = \leftrightarrow(0,0) = 1$ and $\leftrightarrow(1,0) = \leftrightarrow(0,1) = 0$

We can now think of a formula as a circuit, which maps truth assignments to Boolean values: $\varphi : 2^{Prop} \rightarrow \{0,1\}$

Definition 3 *Let $p \in Prop, \tau \in 2^{Prop}$. Then the semantics is defined according to the following rules:*

- $p(\tau) = \tau(p)$ *(meaning of a wire)*
- $(\neg\varphi)(\tau) = \neg(\varphi(\tau))$
- $(\varphi \circ \psi)(\tau) = \circ(\varphi(\tau), \psi(\tau))$

Lemma 1 *Let $\varphi \in Form$ and $\tau \in 2^{Prop}$, then $\tau \models \varphi$ iff $\varphi(\tau) = 1$.*

Proof: Proof by structural induction:

- base: $\tau \models \varphi$ iff $\tau(p) = 1$ iff $p(\tau) = 1$
- induction:
  1. $\neg$: $\tau \models (\neg\varphi)$ iff $\tau \not\models \varphi$ iff $\varphi(\tau) = 0$ iff $\neg(\varphi(\tau)) = 1$ iff $(\neg\varphi)(\tau) = 1$
  2. $\wedge$: $\tau \models (\varphi \wedge \psi)$ iff $(\tau \models \varphi) \wedge (\tau \models \psi)$ iff $\varphi(\tau) = 1$ and $\psi(\tau) = 1$ iff $\wedge(\varphi(\tau), \psi(\tau))$ iff $(\varphi \wedge \psi)(\tau) = 1$
  3. $\vee$: the same as for 2.
  4. $\rightarrow$: $\tau \models (\varphi \rightarrow \psi)$ iff $(\tau \not\models \varphi)$ or $(\tau \models \psi)$ iff $\varphi(\tau) = 0$ or $\psi(\tau) = 1$ iff $\rightarrow (\varphi(\tau), \psi(\tau))$ iff $(\varphi \rightarrow \psi)(\tau) = 1$

∎

4 Software Engineer view

We have seen the philosophical view and the electrical engineer's view of semantics of formulae. Now lets look at how a software engineer views the same thing.

A software engineer describes a set of worlds (or truth assignments) in which a given formula is true. This mapping from formula to sets of truth assignments is called $models$, where $models : Form \rightarrow 2^{2^{Prop}}$.

Definition 4 *Let φ be a formula, then $models(\varphi)$ is defined as follows:*

1. $\varphi = p : models(p) = \{\tau : \tau(p) = 1\}$, *where $p \in Prop$*
2. $\varphi = (\neg\theta) : models(\neg\theta) = 2^{Prop} - models(\theta)$
3. $\varphi = (\theta \wedge \psi) : models(\theta \wedge \psi) = models(\theta) \cap models(\psi)$
4. $\varphi = (\theta \vee \psi) : models(\theta \vee \psi) = models(\theta) \cup models(\psi)$
5. $\varphi = (\theta \rightarrow \psi) : models(\theta \rightarrow \psi) = (2^{Prop} - models(\theta)) \cup models(\psi)$

6. $\varphi = (\theta \leftrightarrow \psi)$: $models(\theta \leftrightarrow \psi) = ((2^{Prop} - models(\theta)) \cap (2^{Prop} - models(\psi))) \cup (models(\theta) \cap models(\psi))$

Lemma 2 *Let $\varphi \in Form$ and $\tau \in 2^{Prop}$, then*
$\tau \in models$ iff $\varphi(\tau) = 1$ (iff $\tau \not\models \varphi$)

Proof:

1. $\tau \in models(p)$ iff $\tau(p) = 1$ iff $p(\tau)$
2. $\tau \in models((\neg\psi))$ iff $\tau \notin models(\psi)$ iff $\psi(\tau) = 0$ iff $\neg(\psi(\tau)) = 1$ iff $(\neg\psi)(\tau) = 1$

∎

φ is the characteristic function of the set $models(\varphi)$.

5 Relevance Lemma

Propositions that do not occur in a formula have no impact on its truth value.

We have to formalize this, but first let us formalize the *occurrence* of a proposition in a formula.

Definition 5 *For $\varphi \in Form$, define $AP(\varphi)$ to be the set of atomic propositions that occur in φ.*

1. $AP(p) = p, where\ p \in Prop.$
2. $AP((\neg\theta)) = AP(\theta), where\ \theta \in Form.$
3. $AP((\theta \circ \psi)) = AP(\theta) \cup AP(\psi), where\ \theta, \psi \in Form.$

Definition 6 *Let $X \subseteq Prop$, and τ_1 and τ_2 be truth assignments. Then τ_1 and τ_2 agree on X if $\tau_1(p) = \tau_2(p)$ for all $p \in X$, denoted by $\tau_1 \mid_X = \tau_2 \mid_X$.*

Lemma 3 *(Relevance lemma)*
Let $\varphi \in Form$ and $\tau_1, \tau_2 \in 2^{Prop}$. If $\tau_1 \mid_{AP(\varphi)} = \tau_2 \mid_{AP(\varphi)}$ then $\tau_1 \models \varphi$ iff $\tau_2 \models \varphi$ and $\varphi(\tau_1) = 1$ iff $\varphi(\tau_2) = 1$.

Proof: (by induction)

1. $\varphi = p, p \in Prop$:
   $\varphi(\tau_1) = \tau_1(p) = \tau_2(p) = \varphi(\tau_2)$.
2. $\varphi = (\neg\theta)$:
   $\varphi(\tau_1) = \neg(\theta(\tau_1)) = \neg(\theta(\tau_2)) = \varphi(\tau_2)$.
3. $\varphi = (\theta \circ \psi)$:
   $\varphi(\tau_1) = \circ(\theta(\tau_1), \psi(\tau_1)) = \circ(\theta(\tau_2), \psi(\tau_2)) = \varphi(\tau_2)$.

∎