COMP 409, Lecture Notes #6

February 4, 2003

1 The *models* relation

Last time we saw:

$$models(\varphi) \subseteq 2^{AP(\varphi)}$$

$$models(\Phi) = \bigcap_{\varphi \in \Phi} models(\varphi)$$

($\varphi \in Form$, $\Phi \subseteq Form$)

Note the intersection in the second equation; each additional φ restricts possibilities.

We also saw $\models$, a relation between truth assignments and formulas.

Definition 1 $\tau \models \varphi \Leftrightarrow \tau \in models(\varphi)$

$\models$ is also used to represent the following relations:

$$T \models \Phi \Leftrightarrow T \subseteq models(\Phi)$$

($T \subseteq 2^{Prop}$)

$$\Psi \models \Phi \Leftrightarrow models(\Psi) \subseteq models(\Phi)$$

The first is a satisfaction relation; the second is a logical implication. Now $\models$ is heavily overloaded; you have to look at context to know which one is being used.

Let's consider the properties of the last relation, $\models \subseteq 2^{Form} \times 2^{Form}$:

1. Reflexive: $\Phi \models \Psi \Rightarrow \Psi \models \Phi$
2. Transitive: $(\Phi \models \Psi) \wedge (\Psi \models \Theta) \Rightarrow \Phi \models \Theta$

These properties mean that $\models$ defines a *partial order* (or *weak partial order*) on 2^{Form}. (NB: It's not a *total order* because $\models$ doesn't apply to every pair $\in 2^{Form} \times 2^{Form}$. For example, $\{p\} \not\models \{q\}$, $\{q\} \not\models \{p\}$.)

2 Monotonicity

Definition 2 *f is monotone with respect to $\sqsubseteq$ if $a \sqsubseteq b \Rightarrow f(a) \sqsubseteq f(b)$.*
f is antimonotone with respect to $\sqsubseteq$ if $a \sqsubseteq b \Rightarrow f(b) \sqsubseteq f(a)$.

For example, $\leq$ is monotone on the right. If $a \leq b$, increasing b preserves this inequality. $\leq$ is also antimonotone on the left; decreasing a preserves the inequality. (This is easy to prove by transitivity.) Similarly, $\subseteq$ is monotone on the right and antimonotone on the left.

What about $\models$?

Lemma 1 *$models$ is antimonotone with respect to $\subseteq$.*

Proof sketch: look at the definition of $models(\Phi)$ at the top. Because of the $\bigcap$, each new φ restricts the right-hand side to be a smaller set, so adding propositions to Φ makes $models(\Phi)$ smaller.

Corollary 1 *$\models$ is monotone on the left and antimonotone on the right.*

Philosophically, the "monotone on the left" part means that every new fact can only strengthen a conclusion. This is contrary to common sense reasoning; in the real world, new facts can cause us to withdraw premises. For example, we might say "If Tweety is a bird, then Tweety can fly." But if we later learn that Tweety is a penguin, we must withdraw that conclusion. Non-monotone logic is a whole different branch of logic that we won't get into now.

3 Logical Equivalence

Definition 3 $\Psi \models\!=\!| \Phi \Leftrightarrow models(\Psi) = models(\Phi)$.

$\models\!=\!|$ is is a *logical equivalence relation.* In plain English, logical equivalence is like saying, "Let me put it in different words." Equivalence relations are exactly those relations that are reflexive, transitive, and symmetric.

We can also say $\psi \models\!=\!| \varphi \Leftrightarrow models(\psi) = models(\varphi)$. For example, $(p \wedge q) \models\!=\!| (q \wedge p)$.

Logical equivalence relations partition their domain into *equivalence classes* in which all elements are logically equivalent. Some special equivalence classes:

- $models((p \wedge \neg p)) = \emptyset$ (*contradiction*)
- $models((p \vee \neg p)) = 2^{Prop}$ (*tautology, valid*)

φ is a tautology iff $\tau \models \varphi$ for all $\tau \in 2^{Prop}$. This is also written $\models \varphi$.

4 Validity and Satisfiability

- Validity: φ is valid if $\models \varphi$.
- Satisfiability: φ is satisfiable if there is a truth assignment that satisfies φ.

There's a duality here:

- φ is valid iff $\neg\varphi$ is not satisfiable.
- φ is satisfiable iff $\neg\varphi$ is not valid.

Lemma 2 *$\varphi \models \psi$ iff $\models \varphi \Rightarrow \psi$.*
$\varphi \models\!= | \psi$ iff $\models \varphi \Leftrightarrow \psi$.

Being computer scientists, we'll mostly focus on satisfiability rather than validity.

5 Constraint Satisfaction

1. Let $\varphi = (((p \rightarrow (q \rightarrow r)) \wedge (p \rightarrow q)) \wedge (p \wedge (\neg r)))$. Is φ satisfiable?
2. Does $3x^2 + 4x + 2 = 0$ have a real solution?

In both of these cases, we are looking for variables that satisfy constraints. This is a common problem-solving paradigm known as *constraint satisfaction*. There are many variants:

1. quadratic equations
2. linear equations
3. linear inequalities
4. cubic equations
5. quartic equations
6. quintic equations
7. differential equations

These are mostly continuous math. In computer science, we more often solve problems in discrete domains (e.g. satisfiability). A good example is *graph coloring*.

5.1 Graph coloring

A *graph* is defined as a set of vertices and a set of edges $G = (V, E)$. A *k-coloring* is a mapping $c : V \to \{0, \ldots, k-1\}$. We want to color the graph so that no two adjacent vertices have the same color, i.e. there is no monochromatic edge. Formally, $(u, v) \in E \Rightarrow c(u) \neq c(v)$.

(Aside: the famous 4-color theorem which was proved recently states that every planar graph is 4-colorable.)

There are two graph coloring problems we are likely to want to solve:

1. *Colorability*: given G and k, is G k-colorable?
2. *Optimization*: find the smallest k such that G is k-colorable.

Let's look at the colorability problem for small values of k.

- 1-colorability: If we only have one color, then clearly no graph with more than 2 vertices is 1-colorable.
- 2-colorability: There is an easy algorithm. Start at some arbitrary vertex v and color it black. Color all vertices adjacent to v white. Continue until the graph is colored or you encounter a vertex that cannot be colored.
- 3-colorability: This one is hard.

What if we want to express the 3-colorability problem as a satisfiability problem? Recall that $G = (V, E)$. Our goal is to generate a formula φ_G such that φ_G is satisfiable iff G is 3-colorable. Our propositions are as follows:

$$Prop = \{P_{v,c} : v \in V, c \in \{\mathtt{R}, \mathtt{G}, \mathtt{B}\}\}$$

Example: $P_{v,\mathtt{R}}$ is true iff vertex v is colored red.

Next time we'll write down the constraints.