Lecture 6: Classification of Formulas

1 Review of Previous Material

1.1 Multiple Viewpoints of Semantics

Semantics is the relationship between formulas and truth values. Previously, we gave three different but equivalent definitions for the formal semantics of propositional logic. We defined a *satisfaction relation* $\models : \subseteq 2^{PROP} \times \text{FORM}$ such that

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

1.2 Types of Formulas

For any formula φ, by definition, we have $\emptyset \subseteq models(\varphi) \subseteq 2^{\text{PROP}}$. So φ can be put in one of three categories:

- φ is *valid* or a *tautology* if for every $\tau \in 2^{AP(\varphi)}$, $\tau \models \varphi$.
- φ is *unsatisfiable* or a *contradiction* or for every $\tau \in 2^{AP(\varphi)}$, $\tau \not\models \varphi$.
- φ is *satisfiable* if there exists $\tau \in 2^{AP(\varphi)}$ such that $\tau \models \varphi$.

For example, $((a \wedge (a \rightarrow b)) \rightarrow b)$ is a tautology, $(a \wedge \neg a)$ is a contradiction, and $(a \wedge b)$ is satisfiable, but not a tautology. Note that every valid formula is also satisfiable. So the class of valid formulas is a subclass of satisfiable formulas.

There is an intuitive relationship between the different classes:

Lemma 1.

1. *ψ is valid $\iff$ $(\neg\psi)$ is not satisfiable.*
2. *ψ is not valid $\iff$ $(\neg\psi)$ is satisfiable.*
3. *ψ is satisfiable $\iff$ $(\neg\psi)$ is not valid.*

These can be easily proved simply from the definitions.

2 Lifting $\models$

In the previous lectures, the satisfaction relation $\models$ was defined over the set $2^{Prop} \times$ FORM, i.e. it was restricted to a *single* truth assignment and a *single* formula. Now we extend this definition to sets. We will write $T \models \Phi$ where T is a set of truth assignments and Φ is a set of formulas. That is $T \subseteq 2^{Prop}$ and $\Phi \subseteq$ FORM.

To define the semantics of $\models$ over sets we use a standard construction called *universal lifting*. If applied to a relation $R \subseteq A \times B$, this construction defines another relation $R' \subseteq 2^A \times 2^B$ such that

$$R'(P, Q) \iff \forall p \in P, q \in Q : R(p, q),$$

i.e. two sets are related in R' iff each pair of elements is related in R.

There are other type of lifting such as *existential lifting*, which means two sets A and B are related in R' iff there exists a pair of elements in A and B which are related in R. For $\models$, we only use universal lifting.

Definition 1. *For a set $T \in 2^{2^{\mathrm{PROP}}}$ and a set $\Phi \in 2^{\mathrm{FORM}}$, $T \models \Phi$ if for every $\tau \in T$ and every $\varphi \in \Phi$, $\tau \models \varphi$.*

This says that a set truth assignments satisfies a set of formulas if all assignments in the set satisfy all formulas.

3 Lifting *models*

Recall that every formula φ is associated with a set of truth assignments $models(\varphi) \subseteq 2^{Prop}$ such that $\tau \models \varphi$ iff $\tau \in models(\varphi)$. We can extend this definition to sets as well so that $models$ becomes a function that takes sets of formulas to sets of truth assignments, that is, $models : 2^{\mathrm{FORM}} \to 2^{2^{Prop}}$.

We must first decide what it means for a truth assignment τ to satisfy *a set of* formulas Φ. We interpret $\Phi = \{\varphi_1, \varphi_2, \ldots\}$ as a big conjunction $\varphi_1 \wedge \varphi_2 \wedge \ldots$ of its elements; in this case, τ satisfies Φ only if it satisfies *each* φ_i. This is called the *conjunctive interpretation* of commas.

Definition 2. *For $\Phi \in 2^{\mathrm{FORM}}$,*

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

An interesting point here is to consider what the value of $models(\Phi)$ should be when Φ is empty. By set-theoretic convention, an empty intersection is defined to be the universal set. Thus, by convention, $models(\emptyset) = 2^{\mathrm{PROP}}$.

With the above definition of $models$, we can obtain a lifted version of the result that $\tau \models \varphi$ iff $\tau \in models(\varphi)$ as follows:

Lemma 2. *For $T \in 2^{2^{Prop}}$ and $\Phi \in 2^{\mathrm{FORM}}$,*

$$T \models \Phi \text{ iff } T \subseteq models(\Phi)$$

Note that the membership operator ($\in$) from the single-element definition of $\models$ has been replaced with a subset operator ($\subseteq$).

Proof. $T \models \varphi$
iff $\forall \tau \in T$, $\varphi \in \Phi$, we have $\tau \in models(\varphi)$
iff $\forall \tau \in T$, we have $\tau \in \bigcap_{\varphi \in \Phi} models(\varphi)$
iff $\forall \tau \in T$, we have $\tau \in models(\Phi)$
iff $T \subseteq models(\Phi)$. □

4 Logical Implication

Observe that the function *models* maps a a set of formulas to a set of truth assignments, while $\models$ relates sets of truth assignments to sets of formulas. We can combine the two to overload $\models$ yet again and define it as a binary relation on sets of formulas. This relation, $\models \subseteq 2^{\mathrm{FORM}} \times 2^{\mathrm{FORM}}$, is called *logical implication.*

Definition 3. *For* $\Phi, \Psi \in 2^{FORM}$,

$$\Phi \models \Psi \ \ \textit{if} \ \ models(\Phi) \subseteq models(\Psi)$$

By notational convention, when Ψ and Φ are singleton sets (i.e. contain a single formula $\varphi \in \Phi$ and $\psi \in \Psi$) we write $\varphi \models \psi$ instead of $\{\varphi\} \models \{\psi\}$. Note that the set of models of the left set of formulas may be equal to the set of models of the right set of formulas. For example, this is true for $\{p, q\} \models \{(p \wedge q)\}$. Every truth assignment that satisfies the formulas p and q also satisfies $(p \wedge q)$ and vice versa. However, there are also cases where $models(\Phi)$ is a strict subset of $models(\Psi)$, e.g. in the case of $\{p, q\} \models \{(p \vee q)\}$.

The name 'logical implication' already hints that there is a strong relationship between this interpretation of $\models$ and the $\rightarrow$ operator. If $\Phi \models \Psi$, then Φ represents a set of 'claims', and Ψ is true whenever the claims hold – regardless of the truth assignment.

4.1 Logical Implication vs. Material Implication

Logical implication is a very fundamental idea in logic. This relation represents a relationship between formulas where if you accept one set of formulas as true then the others will also be true, regardless of the truth assignment.

It is instructive to look at the relationship between *logical implication*, as defined above, and *material implication* which is the 'implies' operator $\rightarrow$.

Recall the definition of logical implication

- $\psi \models \varphi$: if $\tau \models \psi$ then $\tau \models \varphi, \forall\, \tau \in 2^{Prop}$

Look at the case where $\psi \rightarrow \varphi$ is valid. This means $\tau \models \psi \rightarrow \tau \models \varphi$ for all $\tau \in 2^{Prop}$. But this is the same as saying if $\tau \models \psi$ then $\tau \models \varphi$. But this is just the same as logical implication!

So we see that logical implication happens exactly when material implication holds regardless of the current set of facts. This leads to the following lemma:

Lemma 3. $\varphi \models \psi \iff \varphi \to \psi$ *is valid.*

In other words, logical implication is the validity of material implication.

4.2 Another meaning of $\models$

What does it mean to say that $\emptyset \models \Psi$, where Ψ is a set of formulas? The empty set has no "type", so it could be an empty set of formulas or an empty set of assignments.

- Suppose that $\emptyset$ is an element of $2^{2^{\mathrm{PROP}}}$. Since the empty set is a subset of every set, in particular $\emptyset \subseteq models(\Phi)$. In this case $\emptyset \models \Psi$ is trivially true. Obviously this interpretation is not very interesting.

- The second alternative is to consider $\emptyset$ as an element of 2^{FORM}. By definition, the relation $\emptyset \models \Phi$ holds iff $models(\emptyset) \subseteq models(\Phi)$. But what is $models(\emptyset)$? The definition of $models$ tells us that $models(\emptyset) = \cap models(\varphi)$ for all $\varphi \in \emptyset$. By set-theoretic convention, intersection builds from the top-down, that is, we start from the universal set and throw out all elements which are not present in every set whose intersection is being taken. So in the case of the empty intersection, no elements are thrown out and the empty intersection equals the universal set. So, by convention, $models(\emptyset) = 2^{\mathrm{PROP}}$. In other words, $\emptyset \models \Psi$ holds iff $models(\Phi) = 2^{\mathrm{PROP}}$, which is another way of saying that Φ is valid. Thus, $\emptyset \models \Phi$ means that Φ is valid.

The second interpretation is the one that is customarily used. This gives us another meaning for $\models$. Moreover, we can drop the $\emptyset$ from the left-hand side, thereby making $\models$ into an unary operator. We formalize this idea with the next definition.

Definition 4 ($\models$ as validity)**.** *Given $\Psi \in 2^{\mathrm{FORM}}$, we use $\models \Psi$ to denote that each formula in Ψ is valid. That is,*

$$\models \Psi \textit{ if } models(\Psi) = 2^{\mathrm{PROP}}$$

This also gives us a shorter way of stating Lemma **??**:

$$\varphi \models \psi \text{ iff } \models (\varphi \to \psi)$$

The overloading of $\models$ is an example of polymorphism. The types of its arguments determine the meaning of the operator.

4.3 Logical Equivalence

When $\Phi \models \Psi$ and $\Psi \models \Phi$ we have the special case of logical equivalence. From the definition of $\models$ we can infer that $models(\Phi) = models(\Psi)$. This is written as $\Phi {\models\!\!\!=\!\!\!\mid} \Psi$. Intuitively, Φ and Ψ convey the same information.

Definition 5 (Logical Equivalence)**.** $\Phi \models\dashv \Psi$ *iff* $\Phi \models \Psi$ *and* $\Psi \models \Phi$.

It is easy to infer that the relation $\models\dashv$ is reflexive, symmetric and transitive.

$\models\dashv$ partitions 2^{FORM} into equivalence classes. There are two special equivalence classes:

- The equivalence class of $(p \vee (\neg p))$ contains all valid formulas or tautologies.
- The equivalence class of $(p \vee (\neg p))$ contains all unsatisfiable formulas or contradictions.

5 Properties of $\models$

5.1 Reflexivity, Symmetry, and Transitivity

There are three fundamental properties that each relation is tested for: *reflexivity, symmetry*, and *transitivity*. We will examine these three properties for $\models$.

1. Reflexivity: $\models$ is reflexive, because $models(\Phi) \subseteq models(\Phi)$.
2. Symmetry: $\models$ is not symmetric. To see why this is the case, consider $\Phi = \{p \wedge q\}$ and $\Psi = \{p\}$. Clearly $\Phi \models \Psi$ but the reverse is not true: $\Psi \not\models \Phi$. $\models$ is not antisymmetric either, since $\subseteq$ is not antisymmetric and $\models$ is defined by set containment between sets of truth assignments.
3. Transitivity: $\models$ is transitive, because if $\Phi \models \Psi$ and $\Psi \models \Theta$, then $models(\Phi) \subseteq models(\Psi) \subseteq models(\Theta)$. Thus $\Phi \models \Theta$.

A relation that is reflexive and transitive, but not symmetric, is called a *partial order* or *weak partial order*. From the above we can see that $\models$ is a weak partial order, because it is reflexive, not symmetric and transitive. Another weak partial order is $\subseteq$.

5.2 Monotonicity

Definition 6. *Monotonicity*

- *A function f is monotone if $x \leq y \Rightarrow f(x) \leq f(y)$.*
- *A function f is anti-monotone if $x \leq y \Rightarrow f(y) \leq f(x)$*

When we are dealing with relations we also distinguish between *left* and *right* monotonicity, according to the following definitions:

- Left Monotonicity: $xRy \wedge x' \geq x \Rightarrow x'Ry$
- Right Monotonicity: $xRy \wedge y' \geq y \Rightarrow xRy'$
- Left Anti-Monotonicity: $xRy \wedge x' \leq x \Rightarrow x'Ry$

- Right Anti-Monotonicity: $xRy \wedge y' \leq y \Rightarrow xRy'$

For example, the algebraic operator "$\leq$" is left anti-monotonic and right monotonic, while the operator "$\geq$" is left monotonic and right anti-monotonic.

To determine the monotonicity of $\models$ we first need to consider the monotonicity of "$\subseteq$". Intuitively, if $x \subseteq y$, we can remove items from x and add items to y without disrupting the validity of the statement. This means that $\subseteq$ is left anti-monotonic and right monotonic

Similar observations can be made for $models(\Phi) = \cap models(\varphi)$ for all $\varphi \in \Phi$. If $\Phi \subseteq \Phi'$ then $models(\Phi') \subseteq models(\Phi)$. The intuition is that adding more things to the set can only make the intersection smaller; more constraints make for fewer conclusions. Thus, $models$ is anti-monotone.

Now recall that $\models$ is a combination of $\subseteq$ and $models(\Phi)$. When we combine a left anti-monotonic, right monotonic relation with an anti-monotone operator, the operator reverses the monotonicity of the relation. Thus,

- $\models$ is left monotonic
- $\models$ is right anti-monotonic

Informally, this states that we can always take away consequences or add assumptions and validity will still hold. In common-sense reasoning, this is not always the case. For example, if we know that "Tweety is a bird", we infer that"Tweety can fly". But if we are now told that "Tweety is a penguin", then we withdraw the conclusion that Tweety can fly. It turns out that in day-today reasoning adding more assumptions doesn't necessarily preserve the logical implication. This phenomenon is called *non-monotonic reasoning* and it is an important research area in Logic.