COSC 410 Lecture 2
Metalogic

Willem Labuschagne
University of Otago

2016

Introduction

There are two things you can do with logic: apply it or prove things about it. If this were a paper about AI, we would apply logic. Since it's a paper about logic, we will prove things about logic. This is called doing *metalogic*.

What do we want to prove? Well, let's consider what makes a logic what it is. As we saw last time, a logic has two aspects: *syntax* and *semantics*.

Syntax is about the design of a language to represent information symbolically. A language is defined by a grammar (syntax) that tells us which strings make sense.

One way to change your logic is to change the grammar of the logic language. In lecture 1 we introduced a particular type of logic called classical propositional logic. Recall that the logic language L_A is generated from a nonempty set A of atomic sentences by means of the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. Changing A is a small change that gives us a different language of the same kind (i.e. having the same grammar). When we get to *epistemic* logic and *temporal* logic and *first-order* logic, we'll expand the set of connectives, thus changing the syntax quite radically, and the languages we end up with have very different powers of expression.

Syntax is about what is inside the logic language. Semantics is about what is outside the language: what the sentences talk about. Syntax and semantics are supposed to march in step.

Think of an agent explaining to someone in German where to buy a nice cup of coffee in London while pointing out the route on a map. The German words are the syntactic symbols of the language, the map is the semantic model underlying the language level. It had better be a map of London and not of Glasgow or Berlin.

What draws syntax and semantics together in logic? At the heart of every logic is a definition of *satisfaction*, which connects syntax and semantics by telling us which sentences are true in (i.e. satisfied by) which states. Once we have the concept of satisfaction, we can define the crucially important concept of *entailment*, which captures the way in which we are modelling the idea of "thinking that goes right".

So the second way to change your logic is to change the concept of entailment.

This is what we'll do when we get to *nonmonotonic* logic, for example. We'll try to make entailment reflect more closely the way people actually think when they're thinking adaptively about everyday situations.

So there are many kinds of logic. The best way to understand each kind of logic is to prove things about it, in other words to do metalogic. In this lecture we do metalogic about classical propositional logic. When we prove things about a logic, we either try to show that something is a property the logic always possesses, or show that something is not a property always possessed by this kind of logic.

The 3 card system

In the traffic system example of Lecture 1, we assumed that the set S of states was the same as the set W_A of truth assignments, i.e. $S = W_A$. The following example shows that we sometimes want to have $S \neq W_A$. In other words, the property $S = W_A$ happened to hold for the traffic system but does not always hold when we model real systems in classical propositional logic.

Instead of driving a car, our agent is now a card player. The 3 card system is about three players, say player 1, player 2, and player 3, and a pack of cards containing only a red card, a green card, and a blue card. The states of the system are the various possible deals, where each player must get exactly one of the cards.

In order to build a language for representing information about the 3 card system, we could start with the following set of atomic sentences:

$$A = \{r_1, r_2, r_3, g_1, g_2, g_3, b_1, b_2, b_3\}.$$

Think of these atoms as abbreviations of the following assertions:

r_1: The red card is dealt to player 1.

r_2: The red card is dealt to player 2.

r_3: The red card is dealt to player 3.

g_1: The green card is dealt to player 1.

g_2: The green card is dealt to player 2.

g_3: The green card is dealt to player 3.

b_1: The blue card is dealt to player 1.

b_2: The blue card is dealt to player 2.

b_3: The blue card is dealt to player 3.

From these atoms we can build complex sentences in the usual way by making use of the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.

The set W_A consists of all the functions $f : A \longrightarrow \{0, 1\}$, and so there are $2^9 = 512$ functions in W_A. To see this, note that each truth assignment has 9 possible inputs from A, and each input has 2 possible outputs, so that there are $2 \times 2 \times \ldots \times 2$ (9 times) ways to assign one of two outputs to each possible input.

Does this mean the 3 card system has 512 different states? No, most of these 512 truth assignments are spurious and do not represent legitimate deals. There are only 6 different way to deal the three cards to three players according to the rule that each player gets a different card, i.e. only 6 realisable states of the 3 Card System. To see this, note that there are 3 ways for a card to be dealt to the first player, and for each way to deal a card to the first player there are then 2 ways for a card to be dealt to the second player, and finally just one way to deal the single remaining card to the third player, giving $3 \times 2 \times 1 = 6$ ways to deal the cards.

What about the other 506 truth assignments? They are perfectly good truth assignments, but they don't correspond to states of the 3 Card System. They can't happen unless the rules are broken. We basically don't want to waste our time talking about them. We only want to talk about the states that can happen within the rules of the game.

Now, we could use binary strings to talk about the states, but we would need strings with nine bits, which is enough to make your eyes cross and your hair fall out. It takes a bit of effort to get clear the difference between 110100101 and 110101001. There must be an easier way.

Let's instead denote the 6 states by rgb, rbg, grb, gbr, brg, bgr where rgb stands for the state in which player 1 gets the red card, player 2 the green, and player 3 the blue card, etc. (Note that the little strings like rgb are names or labels we've given to states; the strings are not sentences of the language L_A.)

The state we have labelled rgb corresponds to the truth assignment f given by

$$\begin{aligned} f(r_1) &= 1 \\ f(r_2) &= 0 \\ f(r_3) &= 0 \\ f(g_1) &= 0 \\ f(g_2) &= 1 \\ f(g_3) &= 0 \\ f(b_1) &= 0 \\ f(b_2) &= 0 \\ f(b_3) &= 1. \end{aligned}$$

In other words, f corresponds to the state we could call 100010001 if we use binary strings, but choose to call rgb because that's more easily comprehended.

It should be clear what truth assignments the remaining states correspond to.

Since our set of states is

$$S = \{rgb, rbg, grb, gbr, brg, bgr\}.$$

we now have a simple example of a system for which $S \neq W_A$.

However, although $S \neq W_A$, we must not think that the sets S and W_A are disconnected. The definition of satisfaction requires us to know whether an atomic sentence is true or false in a state. If states are just truth assignments, then that's easy. If not, then states have to be associated with truth assignments. This may be done by a *labelling function* $V : S \longrightarrow W_A$ (the V stands for "valuation" because V takes a state and gives a valuation of which atomic sentences are true in that state).

In the case of the 3 card system, the labelling function $V : S \longrightarrow W_A$ is the obvious function sending rgb to the truth assignment f that makes only the atoms r_1, g_2, and b_3 true; sending rbg to the truth assignment making only r_1, g_3, and b_2 true; sending grb to the truth assignment making only r_2, g_1, and b_3 true, and so forth.

What does the definition of satisfaction look like when we allow $S \neq W_A$?

Definition 1 *(Satisfaction for a semantics with S, W_A, and labelling function $V : S \longrightarrow W_A$) Let $\alpha \in L_A$ and let s be any state in S. Then s satisfies α if and only if one of the following is the case:*

- $\alpha \in A$ *and* $V(s)(\alpha) = 1$
- $\alpha = \neg\beta$ *and s fails to satisfy* β
- $\alpha = \beta \wedge \gamma$ *and s satisfies both β and* γ
- $\alpha = \beta \vee \gamma$ *and s satisfies at least one of β and* γ
- $\alpha = \beta \rightarrow \gamma$ *and s satisfies γ or fails to satisfy* β
- $\alpha = \beta \leftrightarrow \gamma$ *and s satisfies either both β and γ or else neither.*

We see that the only change is the way in which we discover whether a state s satisfies an atomic sentence α from A. We must first find the truth assignment corresponding to the state s by calculating $V(s)$. Since $V : S \longrightarrow W_A$, feeding V a state s as input will result in a truth assignment $V(s)$ in W_A as output. Now we can apply this truth assignment to the atomic sentence α and see what truth value we get.

Proof strategies

To show that when we use propositional logic we do not always have to take $S = W_A$, it was sufficient to give an example of a system for which we clearly want to have $S \neq W_A$. Broadly speaking, whenever you need to show that something is not always the case, or show that some sort of situation can exist, then it is enough to provide an example of the right kind.

In contrast, if you want to show that something is indeed **always** the case, or want to show that some sort of situation can **never** exist, then providing an example will not do. Instead, you have to give a general proof. And when you write out a proof, you need to do it in such a way that someone who reads your proof can make sense of it, understanding not just the details but also the overall structure of your argument. In other words, your proof strategy should be familiar and comprehensible to other researchers.

There are 4 proof strategies that we will use: *direct* proof, proof by *contradiction*, *vacuous* proof, and proof by *induction*. (Induction will be discussed in lecture 3.)

Think of a proof strategy as a way to set out your argument that makes clear to others what you are trying to do.

Direct proof

Almost all the things you might want to prove will have the general form "If X is the case, then Y is also the case". In direct proof, we begin by assuming that X is the case and show step by step that Y is the case. Direct proof keeps going in the same direction, forwards, one foot in front of the other.

To illustrate, consider how we might use direct proof to prove that if a and b are odd integers then $a + b$ is an even integer.

Proof. Suppose a and b are odd integers.

Then a and b both leave a remainder of 1 when divided by 2.

So $a = 2k + 1$ and $b = 2m + 1$ for some integers k and m,

i.e. $a + b = 2k + 1 + 2m + 1 = 2k + 2m + 2 = 2(k + m + 1)$.

Now $2(k + m + 1)$ is divisible by 2 without a nonzero remainder.

Hence $a + b$ is an even integer. ∎

More examples of direct proof will be given later.

Proof by contradiction

Proof by contradiction (also called *reductio ad absurdum*) is different from direct proof because instead of going in one direction all the time it involves picking one direction at a fork in the road, reaching a dead end, and then going back to the other tine of the fork. Proof by contradiction is often useful if you've tried direct proof and got stuck because the next step wasn't clear.

The idea is to prove "If X is the case, then Y is the case" by assuming X is the case and then considering the two possibilities for Y: that Y is true, and that Y is false. We deliberately assume Y is false, and try to show that this leads to a contradiction.

Another way to think of it: proof by contradiction starts off like direct proof with departure from X, then pauses to look at a split into two directions. One of them we think is the good direction, because we think it will take us where we want to go. The other is, we suspect, a bad direction, likely to take us to a dead end. To finish the proof we follow the bad direction until we verify that it leads to dead end, i.e. until we find a contradiction. A contradiction means something has gone seriously wrong, so finding a contradiction justifies eliminating the bad direction and leaves us with the other (good) one.

To illustrate, consider how we might use proof by contradiction to prove that if a^2 is an even integer then a is even.

Proof. Suppose a^2 is an even integer.

Now there are two possibilities for a: either a is even or a is odd. (Clearly, this is the fork in the road.)

We want to show that a is even, so let's find a reason to eliminate the case where a is odd.

If a were odd, then $a = 2m + 1$ for some integer m.

But then $a^2 = (2m + 1)(2m + 1) = 4m^2 + 4m + 1 = 2(2m^2 + 2m) + 1$

which leaves a remainder of 1 when divided by 2,

i.e. a^2 would be odd.

But this contradicts what we already know, namely that a^2 is even.

We therefore eliminate the case that a is odd.

Hence a is even. ■

(In case you are wondering whether an integer can be both even and odd, the answer is no, as you'll see below.)

Vacuous proof

Suppose you want to show that "All things having property X also have property Y". Basically what you want to show is that one can never find a thing with property X that lacks property Y, i.e. you want to show that one can never hope to find a *counter-example*. And occasionally we can show this easily by demonstrating that there are no things having property X at all.

To illustrate, we give a vacuous proof that for every integer a which is both even and odd, $a^2 > 113$.

Proof. There are no integers a that are both even and odd.

To see this, note that if a is both even and odd then dividing a by 2 leaves a remainder of 0 (since a is even) but also leaves a non-zero remainder of 1 (since a is odd). This is impossible.

Since there are no integers a having the two properties of being both even and odd, there are no integers having the two properties of being even and being odd but failing to satisfy the inequality $a^2 > 113$. ■

Model theory

The time has come to apply our proof strategies. We will restrict our attention to metalogical properties derived from our definition of satisfaction and, resting on that, the notions of model, entailment, and equivalence. Hence we speak of our results as being *model-theoretic.*

In what follows we shall assume that the language L_A is arbitrary, so that all we know about A is that it is a nonempty set — we do not assume that $A = \{p, q\}$, although that is one of the possibilities for A, and in fact we do not insist that A be finite, i.e. we allow the possibility that $A = \{p_0, p_1, p_2, \ldots\}$. We further assume that we have a semantics for L_A consisting of a set S of states, the set W_A of truth assignments, and a labelling function $V : S \longrightarrow W_A$, in other words everything needed for the definition of satisfaction to work even when we have $S \neq W_A$.

For every sentence $\alpha \in L_A$, we have a set of models $Mod(\alpha)$. Let's get to know this set of models better. It will be helpful to recall some elementary set theory.

Remark 1 *If X and Y are sets, then $X \cap Y$ is the intersection of X and Y, which is the set of all x such that x is in X as well as in Y. Similarly, $X \cup Y$ is the union of X and Y, which is the set of all x that belong to X or to Y or to both.*

Remark 2 *Two sets are equal if they have exactly the same members.*

Theorem 2 $Mod(\varphi \wedge \psi) = Mod(\varphi) \cap Mod(\psi)$.

Proof. First we use direct proof to show that if x belongs to $Mod(\varphi \wedge \psi)$ then x also belongs to $Mod(\varphi) \cap Mod(\psi)$.

Then we use direct proof to show that if x belongs to $Mod(\varphi) \cap Mod(\psi)$ then x also belongs to $Mod(\varphi \wedge \psi)$.

This demonstrates that the two sets $Mod(\varphi \wedge \psi)$ and $Mod(\varphi) \cap Mod(\psi)$ have the same members, and thus the two sets are equal.

Let $x \in Mod(\varphi \wedge \psi)$.

Then x satisfies $\varphi \wedge \psi$.

So x satisfies φ and x satisfies ψ.

So $x \in Mod(\varphi)$ and $x \in Mod(\psi)$.

Hence $x \in Mod(\varphi) \cap Mod(\psi)$.

Since x was chosen arbitrarily, it follows that

every member of $Mod(\varphi \wedge \psi)$ belongs to $Mod(\varphi) \cap Mod(\psi)$.

Conversely, let $x \in Mod(\varphi) \cap Mod(\psi)$.

Then $x \in Mod(\varphi)$ and $x \in Mod(\psi)$.

So x satisfies φ and x satisfies ψ.

So x satisfies $\varphi \wedge \psi$.

Hence $x \in Mod(\varphi \wedge \psi)$.

Since x was arbitrary, every member of $Mod(\varphi) \cap Mod(\psi)$ belongs to $Mod(\varphi \wedge \psi)$. ■

In lecture 1, we used the concept of model to define classical entailment. Let's get to know the entailment relation $\models$ a bit better.

Theorem 3 $\models$ *is reflexive: for all* $\alpha \in L_A$, $\alpha \models \alpha$.

Proof. We use direct proof.

Let $\alpha \in L_A$.

Suppose x satisfies α.

Then x also satisfies α (strange as it may seem).

Hence $\alpha \models \alpha$. ■

That was almost too easy. Let's try another property of $\models$.

Theorem 4 $\models$ *is monotonic: for all* $\alpha, \beta, \gamma \in L_A$, *if* $\alpha \models \beta$ *then* $\alpha \wedge \varphi \models \beta$.

Proof. We use proof by contradiction, not because we have to but just for fun.

Suppose $\alpha \models \beta$.

Then every x satisfying α also satisfies β.

In other words, every model of α satisfies β. (Remember this fact.)

Now there are two possible cases: either $\alpha \wedge \varphi \models \beta$ or $\alpha \wedge \varphi \not\models \beta$.

Suppose $\alpha \wedge \varphi \not\models \beta$.

Then there is some state x that satisfies $\alpha \wedge \varphi$ but that does not satisfy β.

But since this x satisfies $\alpha \wedge \varphi$, we know that this x satisfies α.

And we know every model of α satisfies β. (The fact we had to remember above.)

Thus we have a contradiction: x both satisfies and fails to satisfy β.

This will not do, and so we may eliminate the case $\alpha \wedge \varphi \not\models \beta$.

Thus we are left with $\alpha \wedge \varphi \models \beta$. ■

Next, let's do one with vacuous proof.

Theorem 5 $\models$ *is explosive: for all* $\beta \in L_A$, $\alpha \wedge \neg\alpha \models \beta$.

(A contradiction entails every sentence.)

Proof. No model of $\alpha \wedge \neg\alpha$ fails to satisfy β

because there are no models of $\alpha \wedge \neg\alpha$. ■

Closely related to entailment is equivalence. Let's get to know equivalence better.

Theorem 6 $\varphi \equiv \psi$ *if and only if* $\varphi \models \psi$ *and* $\psi \models \varphi$.

Proof. We use direct proof in each direction.

Suppose $\varphi \equiv \psi$.

Then $Mod(\varphi) = Mod(\psi)$.

Hence $Mod(\varphi)$ and $Mod(\psi)$ have the same members

i.e. every x satisfying φ also satisfies ψ

so that $\varphi \models \psi$,

and every x satisfying ψ also satisfies φ

so that $\psi \models \varphi$.

Conversely, suppose $\varphi \models \psi$ and $\psi \models \varphi$.

Then every x in $Mod(\varphi)$ also lives in $Mod(\psi)$ since $\varphi \models \psi$,

and every x in $Mod(\psi)$ also lives in $Mod(\varphi)$ since $\psi \models \varphi$.

Hence $Mod(\varphi) = Mod(\psi)$.

Thus $\varphi \equiv \psi$. ■

Equivalence ($\equiv$) goes in two directions whereas entailment ($\models$) goes in one direction. So it is usual for proofs about equivalences to have two parts, in opposite directions.

Theorem 7 *For all* $\varphi, \psi \in L_A$, $(\varphi \to \psi) \equiv (\neg\varphi \vee \psi)$.

Proof. Suppose x is in $Mod(\varphi \to \psi)$

i.e. x satisfies $\varphi \to \psi$.

Then x fails to satisfy φ or x satisfies ψ

i.e. x satisfies $\neg\varphi$ or x satisfies ψ

i.e. x satisfies $\neg\varphi \vee \psi$

i.e. x is in $Mod(\neg\varphi \vee \psi)$.

Conversely suppose x is in $Mod(\neg\varphi \vee \psi)$

i.e. x satisfies $\neg\varphi \vee \psi$.

Then x satisfies $\neg\varphi$ or x satisfies ψ

i.e. x fails to satisfy φ or x satisfies ψ

i.e. x satisfies $\varphi \to \psi$

i.e. x is in $Mod(\varphi \to \psi)$.

Combining the two halves, we have that

$Mod(\varphi \to \psi) = Mod(\neg\varphi \vee \psi)$

and thus that $(\varphi \to \psi) \equiv (\neg\varphi \vee \psi)$. ■

Exercises

Quiz: The quiz question for lecture 3 will come from exercise 3 below.

1. Show that
   - $Mod(\neg\varphi) = S - Mod(\varphi)$ where $S - Mod(\varphi)$ is the set of all members of S that do not belong to $Mod(\varphi)$.
   - $Mod(\varphi \vee \psi) = Mod(\varphi) \cup Mod(\psi)$.
2. Show that
   - $\models$ allows right-weakening: if $\alpha \models \beta$ and φ is any sentence, then $\alpha \models (\beta \vee \varphi)$.
   - $\models$ allows conjunction of conclusions: if $\alpha \models \beta$ and $\alpha \models \varphi$ then $\alpha \models (\beta \wedge \varphi)$.
   - $\models$ allows disjunction of premisses: if $\alpha \models \beta$ and $\varphi \models \beta$ then $(\alpha \vee \varphi) \models \beta$.
   - $\models$ is transitive: if $\alpha \models \beta$ and $\beta \models \varphi$ then $\alpha \models \varphi$.
   - $\models$ is contrapositive: if $\alpha \models \beta$ then $\neg\beta \models \neg\alpha$.
   - (Very important) $\models$ is the global form of $\to$ in the following sense:

     $\alpha \models \beta$ if and only if every state in S satisfies $\alpha \to \beta$.
3. Show that for all $\varphi, \psi \in L_A$ the following hold:
   - $\varphi \equiv \neg\neg\varphi$

- $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$
- $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
- $\neg(\varphi \leftrightarrow \psi) \equiv (\varphi \vee \psi) \wedge \neg(\varphi \wedge \psi)$
- $\varphi \equiv \psi$ if and only if every state satisfies $\varphi \leftrightarrow \psi$.

Remark 3 *Logic is not just about technical results. You should spend some time thinking about the concepts and imagining what you would say if your Aunt Maud asked you an obvious question such as: "So what's logic about, then?"*