Philosophical Issues in Quantified Modal Logic

J. Stanley

Handout 1: Propositional Modal Logic

Lewis and Langford, Symbolic Logic, Chapter VI, The Logistic Calculus of Unanalyzed Propositions:

> As indicated in the preceding chapter, we have here the further purpose to develop a calculus based upon a meaning of ‘implies’ such that “p implies q” will be synonymous with “q is deducible from p.” The relation of material implication, which figures in most logistic calculuses of propositions, does not accord with this usual meaning of ‘implies’. It leads to those paradoxes such as “A false proposition implies every proposition” and “A true proposition is implied by any” which have been set forth in Chapter IV…As we shall see, it is entirely possible so to develop the calculus of propositions that it accords with the usual meaning of ‘implies’…(p. 122-3)

Lewis defined his deducibility relation ‘›’ in terms of ‘◊’:

p›q = ~◊(p & ~q) (“It is false that it is possible that p should be true and q false”).

p›q = □(p →q)

Quine, “Three Grades of Modal Involvement”:

> Modal logic received special impetus years ago from a confused reading of ‘→’, the material ‘if-then’, as ‘implies’: a confusion of the material conditional with the relation of implication. Properly, whereas ‘→’ or ‘if-then’ connects statements, ‘implies’ is a verb which connects names *of* statements and thus expresses a relation of the named statements. Carelessness over the distinction of use and mention having allowed this intrusion of ‘implies’ as a reading of ‘→’, the protest thereupon arose that ‘→’ in its material sense was too weak to do justice to ‘implies’, which connotes something like logical implication. Accordingly, an effort was made to repair the discrepancy by introducing an improved substitute for ‘→’, written ‘ ‘ and called strict implication. The initial failure to distinguish use and mention persisted; so ‘ ‘, though read ‘implies’ and motivated by connotations of the word ‘implies’ functioned actually not as a verb but as a statement connective, a much strengthened ‘if-then’. Finally, in recognition of the fact that logical implication is validity of the material conditional, a validity operator ‘nec’ was adopted to implement the definition of ‘p q’ as ‘nec(p→q)’. Since ‘ ‘ had been left at the level of a statement connective, ‘nec’ in turn was of course rendered as an operator directly attachable to statements—whereas ‘is valid’, properly, is a verb attachable to a name of a statement and expressing an attribute of the statement named.

This is a good description of what is going on in C.I. Lewis's work.

Three Grades of Modal Involvement (pp. 158-9).

Lewis's 'Calculus of Strict Implication', System 1 (pp. 123-126)

Lewis's system S2 (p. 166): add to the seven postulates

◊(p & q) › ◊p & ◊q

(In S1, one cannot derive S2, which is clearly problematic)

Lewis provided 5 systems of strict implication. The weaker systems are seriously problematic; no one has ever provided an interpretation for S1.

<u>Normal Modal Systems</u>

The system K (p. 25, Hughes and Cresswell):

Axioms: all valid well-formed formulae (wffs) of propositional calculus

PC If ψ is a valid wff of PC, then ψ is an axiom

K: $L(p \rightarrow q) \rightarrow (Lp \rightarrow Lq)$

Three primitive rules:

US (The Rule of Uniform Substitution): The result of uniformly replacing any variable or variables $p_1 \ldots p_n$ in a theorem by any wff $\beta_1 \ldots \beta_n$ respectively is itself a theorem.
MP (The Rule of Modus Ponens): If ψ and $\psi \rightarrow \chi$ are theorems, then so is χ
N (The Rule of Necessitation): If ψ is a therem, so is $L\psi$

A *normal system* of propositional modal logic is a class S of wff of modal propositional logic which contains all PC-valid wff and K, and has the property that if ψ and χ are in S then so is anything obtainable from them by the use of US, MP, and N.

We will see why this is the right definition of a normal system when we turn to the semantics.

Some other normal modal systems:

T: $Lp \rightarrow p$

The system T is the system K together with the T axiom

4: $Lp \rightarrow LLp$

The system S4 is T together with the 4 axiom

5: $Mp \rightarrow LMp$

The system S5 is T together with the 5 axiom

4 is a theorem of S5 (Hughes and Cresswell, p. 58)

B: $p \rightarrow LMp$

The system B is T together with the B axiom

The semantics for propositional modal logic

The problem of finding a semantical interpretation for “L” and “M” (or “□” and “◊”).

“L” and “M”, like “~”, syntactically are sentence-operators. We do not need to vary from the truth-table to interpret “~”. So perhaps we can use truth-tables to interpret “L” and “M”.

Φ	$\sim\Phi$
T	F
F	T

Φ	$L\Phi$
?	?
?	?

Problem: “L” and “M” are not truth-functional on their natural interpretations as “It is necessary that” and “It is possible that”. For example, “snow is white” is “either snow is white or it is not the case that snow is white” are both true. But replacing one for the other under “It is necessary that” does not preserve the truth-value of the whole sentence:

It is necessary that snow is white or it is not the case that snow is white. (True)
It is necessary that snow is white. (False)

Similarly, both “Kerry won the 2004 election” and “Kerry won the 2004 election and Kerry didn’t win the 2004 election” are false. But:

It could have been that Kerry won the 2004 election. (True)
It could have been that Kerry won the 2004 election and Kerry did not win the 2004 election. (False)

Compositionality: The semantic value of the whole sentence is a function of the semantic value of its parts.

If the semantic value of sentences were truth-values, we would not be able to compute the semantic value of sentences containing “It is necessary that” and “It is possible that” from the semantic values of their parts.

Perhaps the problem is that we have *too few truth-values*. This at any rate was the reaction of Jan Łukasiewicz in 1918 (1920?)

Łukasiewicz: three truth-values, 1, .5 , 0

&	1	.5	0	v	1	.5	0	→	1	.5	0	~	L	M
1	1	.5	0		1	1	1		1	.5	0	0	1	1
.5	.5	.5	0		1	.5	.5		1	1	.5	.5	0	1
0	0	0	0		1	.5	0		1	1	1	1	0	0

As Łukasiewicz points out, this semantics does not validate (~p→p) → p (when p is .5, the whole formula ends up being .5). But it is even worse for modality. Where p is .5, ~p is .5. So where p is .5, p & ~p is .5. But then M(p & ~p) is 1! This is surely a terrible result.

It's a standard objection to certain three-valued truth-tables that they make "p & ~p" .5 (or gap) when p is .5 or (or gap). There is an intuition that "p & ~p" should always be 0. But the situation gets much worse when we treat "M" as yielding 1 when given an argument that is .5.

Maybe we need *four* truth-values: Necessarily True (NT), Necessarily False (NF), Contingently True (CT), and Contingently False (CF).

(1) NT = LΦ
(2) CT = Φ & M~Φ
(3) CF = ~Φ & MΦ
(4) NF = L~Φ

Φ	LΦ	~Φ
1	1	4
2	4	3
3	4	2
4	4	1

But what about "&"?

&	1	2	3	4
1	1	2	3	4
2	2	2	3/4	4
3	3	3/4	3	4
4	4	4	4	4

What about CT & CF? We've got the old 'p & ~p' problem again. If p is contingently true, then ~p is contingently false. If CT & CF = CF, then by the definition of CF, we have:

~(p & ~p) & M(p & ~p)

and by conjunction elimination, we have the bad consequence that led us from 3-valued semantics to 4-valued semantics: M(p & ~p)

So CF & CT must be NF. But that can't be right, for a host of reasons. There are many conjunctions of contingent falsehoods and contingent truths that are not necessarily false.

So it doesn't look like adding more truth-values should help. So let's look at a different approach.

Carnap on state-descriptions (from Meaning and Necessity):

> A class of sentences in S1, which contains for every atomic sentence either this sentence or its negation, but not both, and no other sentences, is called a state-description in S1, because it obviously gives a complete description of a possible state of the universe of individuals with respect to all properties and relations expressed by predicates of the system. Thus the state-descriptions represent Leibniz' possible worlds or Wittgenstein's possible states of affairs.
>
> It is easily possible to lay down semantical rules which determine for every sentence in S1 whether or not it *holds in* a given *state-description*. That a sentence holds in a state-description means, in non-technical terms, that it would be true if the state-description (that is, all sentences belonging to it) were true. A few examples will suffice to show the nature of these rules: (1) an atomic sentence holds in a given state-description if and only if it belongs to it; (2) ~Φ holds in a given state description if and only if Φ does not hold in it; (3) Φ v ψ holds in a state description if and only if either Φ holds in it or ψ holds in it…

A sentence is necessary ("L-true") if and only if it holds in every state-description.

We are not going to identify possible worlds with state-descriptions; we will be neutral about what they are in the semantics we provide.

The basic semantics we will use is given on pp. 38-39 of Hughes and Cresswell.

A wff Φ is valid on a frame <W,R> iff, for every model <W, R, V> based on <W, R>, and for every w in W, V(Φ,w) = 1

A wff is K-valid iff it is valid on every frame.

Note that we have placed no restrictions on the frames. As it turns out, K is sound and complete with respect to the class of all frames.

Soundness: everything provable is something you want to prove (every theorem is semantically valid).

Completeness: every semantically valid wff in the desired class of frames is provable in the system.

But before we discuss these metatheoretical results, let's pause to see that the axioms of the stronger systems T, B, S4, and S5 are not K-valid.

T is not K-valid. Let W = {w, w'}, let R = {<w,w'>}, and let V(p, w) = 0 and V(p, w') = 1. This is a model <W, R, V> that invalidates T. For V(Lp, w) = 1 and V(p, w) = 0. So, V(Lp → p, w) = 0.

Note that if R had contained the pairs <w,w> and <w',w'>, then we would not be able to construct a model based on the frame that would have invalidated T. In short, if R had been a *reflexive* relation on W, we would not have been able to invalidate T. It is because <w,w> is not in R that we were able to construct this invalidating model (since w can't see itself, and it itself is a world in which p is false).

4 is not K-valid. Let W = {w, w', w''}, let R = {<w,w>, <w,w'>, <w',w''>}, and let V(p, w) = 1, V(p, w') = 1, and V(p, w'') = 0. This is a model <W, R, V> that invalidates 4. For V(Lp, w) = 1, and V(LLp, w) = 0.

Note that if R also contained the pair <w,w''> then we would not have been able to invalidate 4, because then V(Lp, w) would have been 0. Note that then, the frame <W, R> would have then been *transitive* (more specifically, the relation R on W would have been transitive). We are able to construct invalidating models for 4 using intransitive frames.

Brief homework (to be brought to section on Monday):

Produce a model that shows the following to be invalid on some frame:

(1) MLp → (p → Lp)

Produce a model that shows the following to be invalid on some reflexive and transitive frame:

(2) p → LMp
(3) Mp → L(p v Mp)