5 Basics of Modal Semantics

5.1 Formal Semantics Generally

Semantics is the study of meaning.

Formal semantics is the study of the meaning or truth conditions of statements in an artificial language (or idealized form of a natural language) using mathematically defined or mathematically precise models.

It should perhaps be noted that these words are not always used consistently. Compare 'biology'.

5.2 Truth Tables and Truth Trees

The formal semantics of extensional or truth-functional propositional logic is more or less exhaused by the study of truth tables.

(Nothing to the meaning of a statement in PL besides its truth values is relevant to the truth/falsity of larger statements of which it forms a part.)

All the important logical concepts for propositional logic can be defined in terms of truth-tables:

- An argument, $A_1, A_2, \ldots, A_n$ /∴ B, is **logically valid** iff there is no row on their combined truth table for which $A_1, A_2, \ldots, A_n$ are true but B is false.
- A truth-table row on which $A_1, A_2, \ldots, A_n$ are all true but B is false is a **counterexample** to the argument $A_1, A_2, \ldots, A_n$ /∴ B.
- An individual statement A is said to be a **logically valid**, or a **tautology** iff it is true for every row of its truth table.
- A series of statements $A_1, A_2, \ldots, A_n$ is said to be **satisfiable** iff there is a row on their combined truth table on which all are true.
- Two statements A and B are said to be **logically equivalent** iff they have the same truth value on every row of their combined truth table.
- And so on, etc.

The notion of "row of a truth table" can be made mathematically precise by defining it as a function assigning exactly one a member of the set {T, F} to each statement letter found in the language of propositional logic. An alternative name for this would be a **truth-value assignment**.

In more complex logical systems, a way of specifying the possible semantic values of the terminology used in it that makes it possible to determine the truth value of any statement is called a **model** or **interpretation**.

The notion of truth in formal semantics is often relativized to "truth on/in a given model", just as in propositional logic we can speak of truth on a given row of a truth table. Logical concepts (logical validity, equivalence, etc.) are then typically defined in terms of the distribution of truth values for *all possible models*.

Since the notion of a truth-value assignment is also important for understanding models in propositional modal logics, it is worth introducing some abbreviated methods for working with truth-value assignments.

A full truth table shows all possible truth-value assignments for a given statement or argument, e.g.:

(	p	$\rightarrow$	q	)	$\vee$	(	r	$\rightarrow$	p	)
	T	T	T		T		T	T	T	
	T	T	T		T		F	T	T	
	T	F	F		T		T	T	T	
	T	F	F		T		F	T	T	
	F	T	T		T		T	F	F	
	F	T	T		T		F	T	F	
	F	T	F		T		T	F	F	
	F	T	F		T		F	T	F	

We can see that the above statement is logically valid. A less cumbersome method of showing this, however, would be to attempt to construct a row on which the above is false, and show this cannot be done. We'd start by filling in an F under the main operator:

(	p	$\rightarrow$	q	)	$\vee$	(	r	$\rightarrow$	p	)
					F					

Because disjunctions are only false when both disjuncts are false, we get:

(	p	$\rightarrow$	q	)	$\vee$	(	r	$\rightarrow$	p	)
		F			F			F		

And by the rules governing '→':

(	p	$\rightarrow$	q	)	$\vee$	(	r	$\rightarrow$	p	)
	T	F	F		F		T	F	F	
	↑								↑	

We have p assigned both T and F on the same truth-table row, which is impossible. The obvious conclusion would be that no truth-table row can make the above statement false, so it must be logically valid.

The same process can be used to show that something isn't logically valid if the process continues until the entire row is filled out, but nothing is given inconsistent assignments:

(	p	&	q	)	→	(	p	→	r	)
	T	T	T		F		T	F	F	

This technique could be used to show that an argument is valid or invalid by starting with assigning T to each premise and an F to the conclusion, and attempting to fill in the rest of the row until it is completed successfully (providing a counterexample) or one is forced into an inconsistent assignment (in which case the argument is valid).

This process gets messy however, if there are multiple ways for a certain statement or substatement to have a certain truth value. Consider:

(	p	∨	q	)	↔	(	q	→	p	)
					F					

Here there are *two* ways for the biconditional to be false: for the left to be true and the right to be false, or the left to be false while the right is true:

(	p	∨	q	)	↔	(	q	→	p	)
		T			F			F		
		F			F			T		

Continuing the process, we'll see that we'd need to "split" the rows again, since there are multiple ways for the disjunction to be true in the first row, and there are multiple ways for the conditional to be true in the second row. Some of these possibilities will lead to inconsistent assignments; some will not. The result would be a rather messy table.

Truth trees, (also known as semantic trees or semantic tableaux) represent a way to do what amounts to the same thing but mess-free.

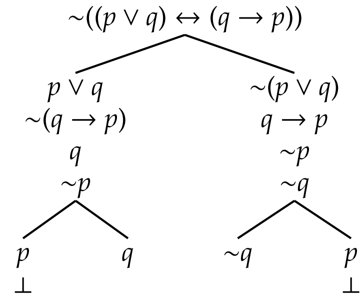
Rather than writing 'T' or 'F' under the main operator, we begin by simply writing a statement or its negation.

If the assumption of the statement's truth or falsity requires parts of it to be definitely true or false, we write those results underneath.

If the assumption of its truth or falsity can be realized in more than one way, we create separate "branches" for each possibility.

We apply the same method to the results on each branch until we get both an atomic statement and its negation on the same branch, which is the same as getting an inconsistency in the purported assignment, or we successfully identify an assignment consistent with our hypothesis. Branches leading to inconsistencies are considered "closed", and are marked with the absurdity symbol, '⊥'.

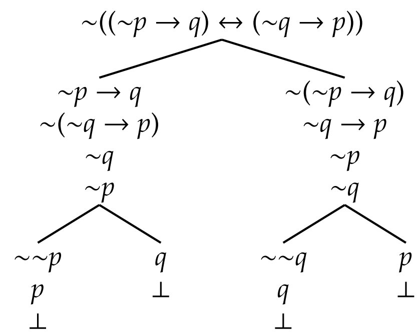
Here is the truth tree for our previous example:



If a completed tree contains 'open' branches (those not closed by '⊥'), it represents a truth-value assignment consistent with the original assumption(s).

Above, the two open branches in the middle each represents a truth value assignment showing that "$(p \vee q) \leftrightarrow (q \rightarrow p)$" is not a tautology. The middle left branch shows that this statement is false when q is true and p is false. The middle right branch shows that it is false when p and q are both false.

On the other hand, if we begin a truth tree with the negation of a tautology, all branches will close:



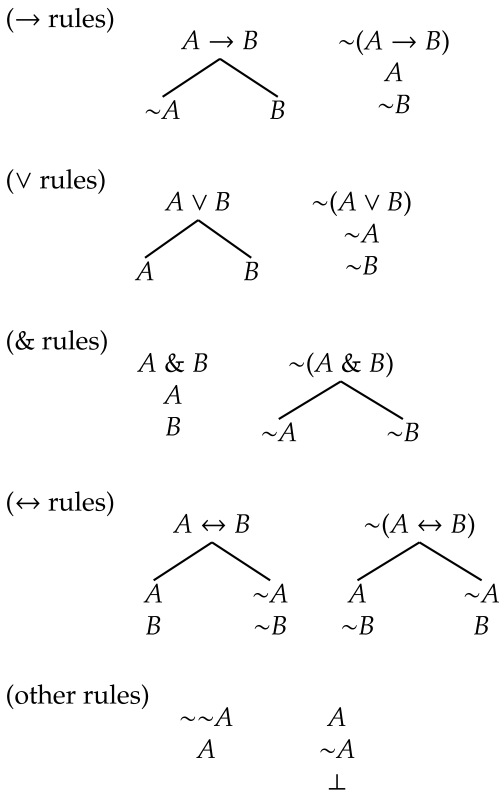
To complete a truth tree, we must apply the appropriate rule to each statement on the tree. The rule is applied to all open branches (if any) under the statement.

Once a branch closes, we may cease working on the tree.

For each binary statement connective, we have two truth tree rules, one for statements of that form, one for negations thereof. We also have a rule for double negations, and for closing branches.

Besides their role closing branches, there are no special rules for dealing with atomic statements or their negations. These are simply used instead to determine the relevant truth-value assignment should any branches remain open.

Here are the rules:

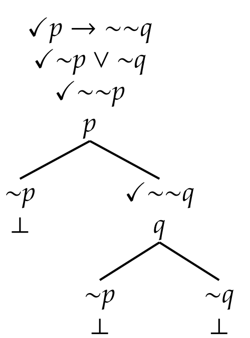


Hopefully, it is obvious that these match the truth-table rules.

To avoid dealing with too many branches, it is generally easiest to apply non-branching rules *before* branching rules.

It is also helpful to place checkmarks (✓) next to each statement as you apply the rule to it to ensure that none is forgotten.

To establish the validity of the argument: $p \rightarrow {\sim}{\sim}q$, ${\sim}p \vee {\sim}q$ /∴ ${\sim}p$, we show that the tree assuming its premises true and its conclusion false has no open branches:



5.3 Modal Models

We might consider expanding the formal semantics for extensional propositional logic by adding truth-table rules to cover modal operators.

Taking rows of a truth table themselves as representing distinct possible worlds, it is natural to put a T under a '□' operator iff there is T under the main operator to which '□' is applied is true not only on the row in question, but on *all* rows:

□	(	p	→	p	)	→	□	(	p	→	q	)
T		T	T	T		F	F		T	T	T	
T		T	T	T		F	F		T	F	F	
T		F	T	F		F	F		F	T	T	
T		F	T	F		F	F		F	T	F	

For example, the reason we have a T on the first row in the first column is not just because there is a T in the third column in the first row, but because there is a T in *every* row in that column. We have an F on the first row under the second '□' because of the F in the *second* row two columns over.

However, this approaches faces some limitations:

- It is only suitable if '□' is interpreted as meaning 'it is a tautological that...', not for other, weaker conceptions of necessity.
- It takes all combinations of truth values for atomic statements to be possible.
- It does not allow what is necessary to differ from one context of evaluation to another.

Suppose for example that '*p*' above means "Barney is purple" and '*q*' means "Barney is colored". In that case, the second row, where *p* is true but *q* is false, might not be regarded as a legitimate possibility according to our standards of evaluation in the actual world (the first row).

One thought might be to draw arrows from the first row to those rows that do represent genuine alternative possibilities for it:

□	(	p	→	p	)	→	□	(	p	→	q	)
T		T	T	T		T	T		T	T	T	
T		T	T	T		?	?		T	F	F	
T		F	T	F		?	?		F	T	T	
T		F	T	F		?	?		F	T	F	

In that case we get a T in the first row under the second box, since $p \rightarrow q$ is true for every row of the table that is considered possible on the first row.

This way of thinking of things is at least *very close* to what is going on in our more mathematically precise definitions below.

The arrows in such a table represent what we shall call **the accessibility relation**, i.e., the relation between one world or context of evaluation and those other worlds or contexts of evaluation that are relevant to the truth conditions of modal statements in the first world or context of evaluation.

However, using an expanded truth-table method such as this still has limitations:

- It does not allow us to represent worlds or contexts of evaluation that agree on their truth-value assignment for atomic statements but differ in their accessibility status.
- If we made similar arrows for every row, the result would be a tangled mess. (Indeed, we shall mainly make use of trees rather tables from now on because of this.)

It is useful then to give some more precise definitions for use in modal semantics.

A **K-model** is an triple $\langle \mathbf{W}, \mathbf{R}, \mathbf{a} \rangle$, where:

- **W** is a set of different contexts of evaluations, **w**, **v**, **u**, **w′**, . . ., etc.. These might be different possible worlds, or moments of time, or states of belief, depending on the particular undertanding of modality involved. The precise nature or metaphysical status of the members of **W** is not important; indeed, the only thing we assume is that for each **w** in **W**, each atomic statement of the language can be evaluated as T or F (but not both).
- **R** is a relation (the accessibility relation) defined over the members of **W**. We write **wRv** to mean that this relation holds between **w** and **v** (or that **v** is accessible from **w**).
- **a** is a function, definable in terms of **W** and **R** that assigns exactly one member of the set {T, F} to each well formed statement of the language of modal propositional logic for each member of **W**. It is subject to further restrictions noted below.

Do not be overly concerned with the description of K-models or other modal models as set theoretic entities like "ordered triples". One does not need to know a lot of set theory to follow our reasoning regarding them. The important thing is that a K-model provides three things: a collection of worlds (or other evaluation contexts) to consider, an accessibility relation, and an *evaluation function* that determines which statements are true for which worlds on that model.

We shall use the notation "$\mathbf{a_w}(A)$" as a name of the truth-value that **a** assigns to A for world **w**. (This notation is part of the metalanguage, the language we use to discuss our modal systems, not something we are using within those systems themselves.) To count as a K-model, the **a** of a given model must obey the following restrictions for each world **w** in **W**:

For any atomic statement P, $\mathbf{a_w}(P) = \text{T}$ or F depending on the nature of **w**.

$\mathbf{a_w}(\bot) = \text{F}$

$$\mathbf{a_w}(A \rightarrow B) = \begin{cases} \text{T} & \text{if } \mathbf{a_w}(A) = \text{F or } \mathbf{a_w}(B) = \text{T} \\ \text{F} & \text{otherwise} \end{cases}$$

which, together with our definitions, also yield:

$$\mathbf{a_w}(\sim A) = \begin{cases} \text{T} & \text{if } \mathbf{a_w}(A) = \text{F} \\ \text{F} & \text{if } \mathbf{a_w}(A) = \text{T} \end{cases}$$

$$\mathbf{a_w}(A \mathbin{\&} B) = \begin{cases} \text{T} & \text{if } \mathbf{a_w}(A) = \text{T and } \mathbf{a_w}(B) = \text{T} \\ \text{F} & \text{otherwise} \end{cases}$$

$$\mathbf{a_w}(A \vee B) = \begin{cases} \text{T} & \text{if } \mathbf{a_w}(A) = \text{T or } \mathbf{a_w}(B) = \text{T} \\ \text{F} & \text{otherwise} \end{cases}$$

$$\mathbf{a_w}(A \leftrightarrow B) = \begin{cases} \text{T} & \text{if } \mathbf{a_w}(A) = \mathbf{a_w}(B) \\ \text{F} & \text{otherwise} \end{cases}$$

Which, of course agrees with truth-table and tree rules. For statements involving modal operators:

$$\mathbf{a_w}(\Box A) = \begin{cases} \text{T} & \text{if } \mathbf{a_v}(A) = \text{T for } \textit{every } \mathbf{v} \\ & \quad \text{in } \mathbf{W} \text{ such that } \mathbf{wRv} \\ \text{F} & \text{otherwise} \end{cases}$$

by the definition of '$\Diamond$', this gives

$$\mathbf{a_w}(\Diamond A) = \begin{cases} \text{T} & \text{if } \mathbf{a_v}(A) = \text{T for } \textit{some } \mathbf{v} \\ & \quad \text{in } \mathbf{W} \text{ such that } \mathbf{wRv} \\ \text{F} & \text{otherwise} \end{cases}$$

One might worry that the above provisos would make any definition of **a** illegitimate, since it seems to be defined in terms of itself. In fact, this is what is often called a recursive definition. The value of $\mathbf{a_w}$ is defined non-circularly for atomic statements and '$\bot$', and the value of more complicated statements are defined in terms of simpler ones. Since no statement can be infinitely complex, this is a satisfactory definition.

Further definitions are in order:

An argument, $A_1, A_2, \ldots, A_n$ /∴ B is **K-valid** iff there is *no* K-model, $\langle \mathbf{W}, \mathbf{R}, \mathbf{a} \rangle$, and world **w** in **W**, for which $\mathbf{a_w}(A_1) = \text{T}$ and $\mathbf{a_w}(A_2) = \text{T}$ and . . . and $\mathbf{a_w}(A_n) = \text{T}$ but $\mathbf{a_w}(B) = \text{F}$.