3.33. Logical Duality

1. Duals and Duality. Here we return to striking parallels noticed in our exploration of the formal language, the better to understand how different bits of this language pair up as 'twins' or 'mirror images' of one another. Armed with a table of such twin items, we will come to recognize the fundamental symmetries running through the entire formal language and its semantics. These parallels not only yield a deeper understanding of the formal language, but also reveal a network of 'shortcuts' useful for proving things by and about the language

As a simple illustration, consider a potential misinterpretation of the semantic rules for molecular sentences.[1] Suppose an alien visiting our planet misreads the "1" in our truth tables to mean *False*, and the "0" to mean *True*. Take the semantic rule for conjunctions as an example.

Conjunction Rule:

●	▲	(● ∧ ▲)
1	1	1
1	0	0
0	1	0
0	0	0

We illustrate the misreading in two steps. First, every "1" is read as meaning **false**. (To avoid confusion, we represent *false* from the alien's point of view by the word "False," rather than the traditional "0".)

●	▲	(● ∧ ▲)
False	False	False
False	0	0
0	False	0
0	0	0

[1] Borrowing an example from Kleene (1967/2002: 23-24).

And every “0” is read as meaning **true**.

●	▲	(●∧▲)
False	**False**	**False**
False	True	True
True	False	True
True	True	True

As the boldfaced first valuation shows, this misreading makes the whole sentence *False* only when both its parts are *False*. Since that is the semantic rule for disjunctions, **misreading “1” as *false* and “0” as *true* amounts to reading “∧” sentences as disjunctions**.

The alien will misread the semantic rule for disjunctions in the same way.

Disjunction Rule:

●	▲	(●∨▲)
1	1	1
1	0	1
0	1	1
0	0	0

“1” is read as *false*.

●	▲	(●∨▲)
False	False	False
False	0	False
0	False	False
0	0	0

And “0” is read as *true*.

●	▲	(●∨▲)
False	False	False
False	True	False
True	False	False
True	**True**	**True**

As the last, boldfaced valuation emphasizes, on this misreading a "$\vee$" sentence is only true when both its parts are true. **Misreading "1" as *false* and "0" as *true* amounts to reading "$\vee$" sentences as conjunctions**.

Though Truth and Falsehood are clearly paired as 'opposites' in our bivalent semantics, we now see something more: by having Truth and Falsehood switch places systematically (as in the alien misreading), conjunctions and disjunctions are likewise revealed as semantic 'mirror images' of one another.

Such 'mirror images' are called **duals**. So conjunction and disjunction are dual types of sentences.

Pressing the 'mirror image' metaphor further highlights another point about duals. Note that the mirror image of a mirror image is just the original image again. For example: the mirror image of this page of text is the text left-right reversed. But taking the mirror image *of that switched text* yields the original text again.

Likewise with duals: starting with the conjunction truth table and taking its dual yields the disjunction truth table. But as we've seen, the dual of the disjunction table is just the conjunction table again. In general: **the dual of the dual is just the original**. (In technical jargon: duality is **involuntary**.)

We construct a list of such 'mirror images' pairs.

True	**False**
Conjunction	**Disjunction**

An immediate application of this table comes in a parallel between the semantic rules for conjunction and disjunction.[2]

> A **conjunction** is only **true** when both its parts are **true**.
> A **disjunction** is only **false** when both its parts are **false**.

[2] Noted earlier in "*3.15. Formal Semantics: Disjunctions*".

We don’t need to remember two different semantic rules here. Armed with the table of duals and one of the semantic rules, we can extract the other rule by systematically terms with their duals.

True	**False**
Conjunction	**Disjunction**

	conjunction		**true**		**true**
A		is only		when both its parts are	
	disjunction		**false**		**false**

What duality reveals is an underlying rule applying **symmetrically** to both conjunction and disjunction.

Consider next the dual of the negation.

Negation Rule

●	~●
1	0
0	1

Again we read every “1” as false.

●	~●
False	0
0	False

And we read every “0” as true.

●	~●
False	True
True	False

This is a sentence which is false when its (one) part is true, and true when that part is false. But that’s just the truth table for negation again. So: **negation is its own dual**.

True	**False**
Conjunction	**Disjunction**

Negation

Here again, we can remember just half of the semantic rule, and extract the other half by duality.

	true		**negation**		**false**
When a sentence is		its		is	
	false		**negation**		**true**

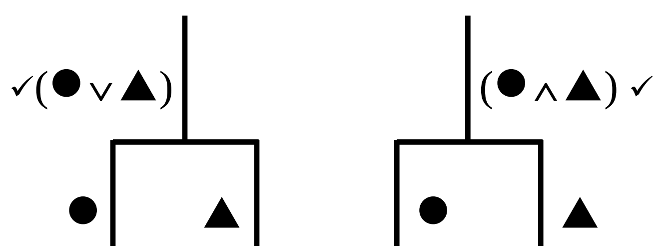
(As a mirror image analogy, imagine a shape which is perfectly left-right symmetrical. Taking the mirror image of that shape yields that shape again.)

And while our first examples of duality were presented in truth table form – systematically swapping True and False in truth table format – the semantic rules in truth tree notation offer a visually striking illustration of duality in terms of left/right symmetry.

Since left of the line means True and right means False, we systematically switch True and False by moving all sentences on the left to the right, and all sentences on the right to the left. Applying this transformation to the true conjunction yields its dual.

The sentence on the right – a sentence false only when both its parts are false – is the truth tree rule for a false **disjunction**. **The dual of a true conjunction is a false disjunction**.

The same True-False interchange shows that **a true disjunction is the dual of a false conjunction**.



And the mirror image of a true negation is a false negation (and vice versa). **Negation is its own dual**.

2. Methods of Duality. We review several different methods for finding duals.

First Method: True/False Swap. Our approach to finding the dual of a sentence type has been through the semantics, beginning with the alien's systematic misreading of the 1/0 notation. Let us call that approach the "**True/False Swap Method**" for finding a dual.

> **True/False Swap Method**: To find the dual of a given truth table, systematically switch True and False through that truth table.

Since it begins and ends with truth tables, this method allows us to speak meaningfully about the **dual of a truth table**. The method applies to other things – such as sentences, or types of sentences – only by their association with truth tables.

So with a pair of sentences in hand, the True/False Swap method allows us to determine whether they are duals: build the truth table for one sentence, and see whether the True/False Swap Method yields from that the truth table for the other sentence.

For instance, to determine whether “(~P ∨ Q)” and “(~P ∧ Q)” are duals, we first build the truth table for one of them – say, “(~P ∨ Q)”.

P	**Q**	**~P**	**(~P ∨ Q)**
1	1	0	1
1	0	0	0
0	1	1	1
0	0	1	1

True/False Swap yields the dual of this truth table: a table **only true when “P” is false and “Q” is true**.

P	**Q**	…	**Dual of “(~P ∨ Q)”**
False	False		False
False	**True**		**True**
True	False		False
True	True		False

And that is the truth table for “(~P ∧ Q)”.

P	**Q**	**~P**	**(~P ∧ Q)**
1	1	0	0
1	0	0	0
0	**1**	1	**1**
0	0	1	0

Since “(~P ∨ Q)” and “(~P ∧ Q)” take dual truth tables, we conclude that the two sentences are indeed duals.

But having sentence duality ride on the coattails of truth table duality comes with certain limitations. Specifically: beginning with one sentence, the True/False Swap Method doesn’t succeed in isolating a *single* sentence as its dual.

For example: given "(~P ∨ Q)" and its corresponding truth table, and executing True /False Swap on that truth table will yield a dual truth table with only one true valuation: where "P" is false and "Q" is true.

But while "(~P ∧ Q)" does take indeed that dual truth table, so do an infinity of other sentences – e.g., the double negation of "(~P ∧ Q)," the quadruple negation of "(~P ∧ Q)," and so on. Beginning with one sentence, the True/False Swap Method doesn't find **the** dual of that sentence, but rather an infinite number of duals of that sentence. If we wish to speak of *the* (unique) dual of a sentence – as we speak meaningfully of *the* (unique) dual of a truth table – the True/False Swap Method falls short.

Second Method: Tilde Insertion. Two observations about Truth/False Swaps point the way toward a different method of finding duals of sentences.

Note ***first*** that in performing the True/False Swap, we only need attend to truth tables for the sentence letters (on the left end of the truth table) and the finished sentence (on the right). That's why, in our last example, we only focused on the sentence letters, and the final truth table on the right.

P	**Q**	…	**Dual of "(~P ∨ Q)"**
False	False		False
False	True		True
True	False		False
True	True		False

The intermediate steps were just a means of calculating that final truth table, and so acted like a ladder we could kick away once we'd climbed it.

Note ***second*** that changing True into False and False into True is the characteristic semantic behavior of **negation**.

Putting those two points together: running a True/False Swap on the sentence letter truth tables, and on the truth table for the whole sentence (on the right) is equivalent to **negating each of the sentence letters, and the whole sentence**. This forms the **Tilde Insertion Method** of finding a dual.

> **Tilde Insertion Method**: to find a dual of a given sentence, insert a tilde before each sentence letter, and before the whole sentence.

Worth noting right away is that the Tilde Insertion Method doesn’t require us to construct any truth tables at all. (This will make the Tilde Insertion Method a useful way of finding duals of sentences in later chapters where the semantics moves beyond truth tables.)

Return, for instance, to our previous example, Tilde Insertion yields the following dual sentence.

(~P ∨ Q)	Original Sentence
$\underline{\sim}(\sim\underline{\sim}P \vee \underline{\sim}Q)$	Result of Tilde Insertion

And simplification by way of Double Negation and DeMorgan’s Law does yield the sentence “(~P ∧ Q)”.

(~P ∨ Q)	Original Sentence
~(~~P ∨ ~Q)	Tilde Insertion
~(P ∨ ~Q)	Double Negation
(~P ∧ ~~Q)	DeMorgan’s Law
(~P ∧ Q)	Double Negation

But note: this approach recognizes “(~P ∧ Q)” as a dual of “(~P ∧ Q)” only after these simplification steps.

Now if our concept of dual sentence here allows for different degree of simplification, then "~(~~P ∨ ~Q)," "~(P ∨ ~Q)," "(~P ∧ ~~Q)," and "(~P ∧ Q)" will all count as duals of the original sentence – and so will not provide any one sentence as **the** dual of "(~P ∨ Q)".

Whereas without such simplification, only "~(~~P ∨ ~Q)" – and not the simpler "(~P ∧ Q)" – counts as **the** dual of "(~P ∨ Q)".

Third Method: Connective Swap. To keep duals of sentences as simple as possible, while still assigning each sentence exactly one dual, we thus turn to a method which builds duals of sentences by first specifying **duals of connectives**.

Using the table of duals already establish, we state for each connective its dual connective.

True	**False**
Conjunction	**Disjunction**

Negation

Following this approach, the **dual of a wedge is a vel** (and vice versa) while the **tilde is a self-dual**. To get the dual of a given sentence, we then replace every connective in that sentence with its dual. Concretely, that means replacing each wedge by a vel and each vel by a wedge. (Since tilde is its own dual, replacing a tilde with its dual leaves it unchanged.) This is the **Connective Swap Method**.

> **Connective Swap Method**: to find a dual of a given sentence, replace each wedge in the sentence with a vel, and each vel with a wedge.

Whereas the True/False Swap method had us find a dual sentence via a detour through truth tables, with the Connective Swap Method that is reversed, allowing duals of truth tables to ride on the coattails of sentence duals. To find the dual of a given truth table through Connective Swap, we (i) find a sentence matching that truth table (any matching sentence will do);

then (ii) build the Connective Swap dual of that sentence; then (iii) build the truth table for that dual sentence.

For example, to find a dual of the following truth table through Connective Swap, we first find a corresponding formal sentence.

P	**Q**	**Truth Table**
1	1	1
1	0	0
0	1	0
0	0	1

Here familiarity with DNF comes in handy, yielding a corresponding sentence “((P ∧ Q) ∨ (~P ∧ ~Q))”.

P	**Q**	**Truth Table**	**(P ∧ Q)**	**~P**	**~Q**	**(~P ∧ ~Q)**	**((P ∧ Q) ∨ (~P ∧ ~Q))**
1	1	**1**	1	0	0	0	**1**
1	0	**0**	0	0	1	0	**0**
0	1	**0**	0	1	0	0	**0**
0	0	**1**	0	1	1	1	**1**

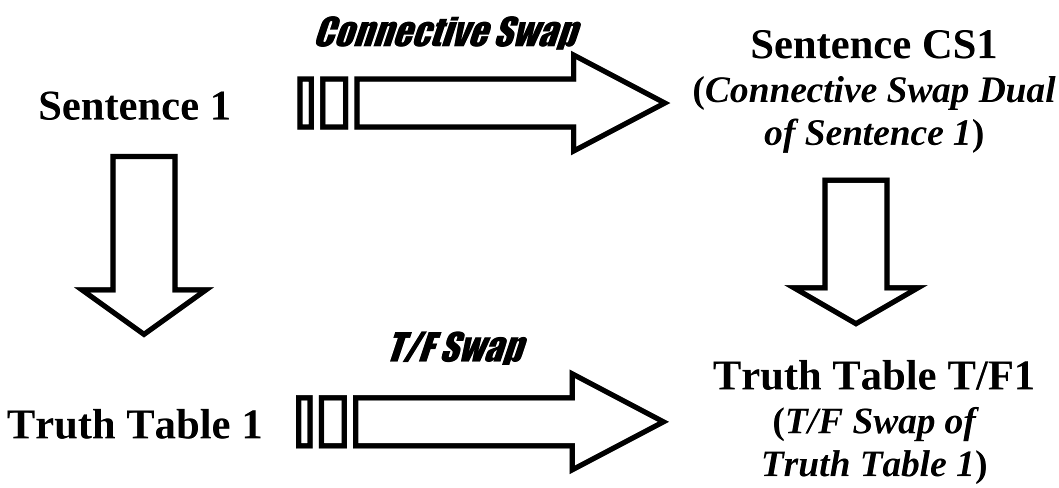
The dual of “((P ∧ Q) ∨ (~P ∧ ~Q))” by connective swap is the sentence “((P ∨ Q) ∧ (~P ∨ ~Q))” – stated here with its corresponding truth table.

P	**Q**	**Truth Table**	**(P ∨ Q)**	**~P**	**~Q**	**(~P ∨ ~Q)**	**((P ∨ Q) ∧ (~P ∨ ~Q))**
1	1	**1**	1	0	0	0	**0**
1	0	**0**	1	0	1	1	**1**
0	1	**0**	1	1	0	1	**1**
0	0	**1**	0	1	1	1	**0**

And note: the truth table for "((P ∨ Q) ∧ (~P ∨ ~Q))" is indeed the True/False Swap of the original truth table: true when just "Q" is true, and when just "P" is true.

P	Q	T/F Swap
False	False	False
False	**True**	**True**
True	False	**True**
True	True	False

A nice parallel holds in general between the methods of Connective Swap and True/False Swap: beginning with a formal sentence, the semantic rules assign it a corresponding truth table. And the Connective Swap dual of that sentence is takes a truth table that's the True/False dual of the original truth table.



Since the semantic rules assign each formal sentence exactly one truth table, Connective Swap will always bring True/False Swap in its wake. And for that reason the Connective Swap Method will be our preferred method for finding duals. In particular: in what follows, when we speak in of **the** (unique) dual of a sentence, we will mean the Connective Swap dual of that sentence.

3. Duality: Further Features. It will hold in general that **the dual of a tautology is a contradiction**. That is clear already from the True/False Swap Method: the truth table for a tautology (true in every valuation) becomes the truth table for a contradiction (false in every valuation).

The Connective Swap Method yields the same results: the tautology “(P ∨ ~P),” for instance, has as its dual the contradiction “(P ∧ ~P)”.

True	**False**
Conjunction	**Disjunction**
Tautology	**Contradiction**

Negation

Next, note that by the Connective Swap Method **a sentence in Disjunctive Normal Form has a sentence in Conjunctive Normal Form as its dual**. So whereas, e.g., “(~P ∧ Q) ∨ (P ∧ ~Q)” is in DNF, its dual “(~P ∨ Q) ∧ (P ∨ ~Q)” is in CNF.

True	**False**
Conjunction	**Disjunction**
Tautology	**Contradiction**
CNF Sentence	**DNF Sentence**

Negation

This point confirms our previous observation about duality. For we noted earlier that a DNF sentence, each of whose “cells” (basic conjunctions) contains a sentence letter and also its negation, is a contradiction.[3] For instance, “(~P ∧ P) ∨ (Q ∧ ~Q)” is a DNF contradiction. And note that its Connective Swap dual is the CNF sentence “(~P ∨ P) ∧ (Q ∨ ~Q)” – which, having a sentence letter and its negation in each of its “cells” (basic disjunctions), is a tautology. Once again: the dual of a contradiction is a tautology.

[3] In “*3.29. Conjunctive and Disjunctive Normal Form*”.

Finally, note that **if two sentences are logically equivalent, their duals sentences are also logically equivalent**. This follows directly from the parallel , noted above, between duals sentences from Connective Swap and dual truth tables from True/False Swap. For Sentence 1 and Sentence 2 to be logically equivalent just means they take the same truth table – say, Truth Table 1. And the parallel between Connective Swap and True/False Swap guarantees that the Connective Swap of both sentences will be a pair of sentences both taking the dual of Truth Table 1. Since the dual of Sentence 1 and the dual of Sentence 2 both take the same truth table, they are logically equivalent sentences.

Summary: Duals of Truth Tables and Sentences

Method 1: True/False Swap Method

- For a given truth table (an array of 2^N 1s and/or 0s), its dual is obtained by replacing each 1 with a 0, and each 0 with a 1.

Method 2: Tilde Insertion Method

- For a given sentence, its dual is obtained by (i) inserting a tilde **before each sentence letter**, and **before the entire sentence**.

Method 3: Connective Swap Method

- For a given sentence, its dual is obtained by replacing each wedge with a vel, and each vel with a wedge.

The dual of a **contradiction** is a **tautology**.

The dual of a sentence in **Disjunctive Normal Form** is a sentence in **Conjunctive Normal Form**.

If two sentences are **logically equivalent**, their duals sentences are also logically equivalent.