2.33. Logical Duality

1. Duals and Duality. Here we return to striking parallels noted 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 formal language and its semantics. These parallels yield not only a deeper understanding of the formal language, but also a network of ‘shortcuts’ useful for proving things about the language and the claims made in that 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* in the alien’s interpretation 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: 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 the semantic rule for conjunctions as the disjunction rule**.

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 the semantic rule for disjunctions as the conjunctions rule**.

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), the conjunction and disjunction rules are likewise revealed as semantic 'mirror images' of one another.

Such 'mirror images' are called **duals**. So the semantic rules for conjunctions and disjunction are duals of one another.

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 truth table in the conjunction rule and taking its dual yields the truth table in the disjunction rule. But as we've seen, the dual of that disjunction truth table is just the conjunction table again. In general: **the dual of the dual is just the original**. (In technical jargon: duality is **involutary**.)

Duality was the basis for a striking parallel between the semantic rules for conjunction and disjunction, noted earlier.[2]

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

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 replacing items with their duals. What duality reveals is an underlying **symmetry** to the conjunction and disjunction rules.

---

[2] At the end of 2.15.

Consider next the semantic rule for negation.

●	~●
1	0
0	1

To construct its dual, we again 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 semantic rule for negation again. So: **the negation rule is its own dual**. (As a mirror image analogy, imagine a shape which is perfectly left-right symmetrical. Taking the mirror image of that shape just yields that shape again.)

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**

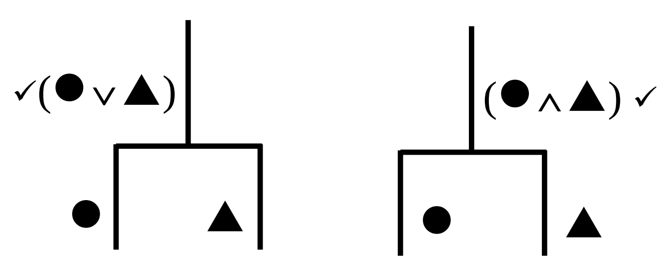
And while our examples of duality were presented in truth table form – systematically swapping True and False in 1/0 format – the semantic rules expressed in **truth tree** notation yield a visually striking illustration of duality in terms of mirror image symmetry.

Since left of the line means True and right means False, we systematically switch True and False in truth trees 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 rule yields its dual.

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

The dual of the True Conjunction rule is the False Disjunction rule.

The same True-False interchange shows that **the dual of the True Disjunction rule is the False Conjunction rule**.



And the mirror image – the **dual** – of the **True Negation** rule is the **False Negation** rule (and – thanks to involution – vice versa).

2. Duals of Truth Tables: The True/False Swap. By focusing on semantic rules, we've met the concept of duality at the intersection of two distinct parts of formal logic: the **formal semantics** (in both truth table and truth tree notation) and the **formal language** (the family of sentences generated by the construction rules).

For as noted earlier[3], the semantic rules (in 1/0, truth table notation) knit together truth tables and formal sentences. Since the three 'molecular' construction rules (for negations, conjunctions, and disjunctions) have matching semantic rules, each move made in the construction of a sentence is matched by a parallel move by the semantic rules (in the form of a truth table for that sentence). This parallel guarantees that every sentence of the formal language has a corresponding truth table – the 'semantic shadow' cast by that sentence.

So while we have discussed only the duals of semantic rules – the point where formal sentences and truth tables meet – we can pursue the topic of duality further in either of these two directions: developing an account of **duals of individual truth tables** (rather than of general semantic rules) or **duals of (construction rules and) individual sentences**.

---

[3] In 2.16.

The earlier discussion of duality extends naturally to duality for individual truth tables. Given some truth table, we construct its dual by systematically replacing true with false (and vice versa). Call this the "True/False Swap" method of duality.

> **True/False Swap**: For a given truth table, the True/False Swap of that truth table is the result of replacing each True in that truth table with False, and each False in that original truth table with True.

For instance, the following truth table is only false in the first valuation – where "P" and "Q" are both true.

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

P	**Q**	**Dual of Truth Table 1**
0	**0**	**1**
0	1	0
1	0	0
1	1	0

So its dual is a truth table true only true when both "P" and "Q" are false.

A Shortcut for Dual Truth Tables: The Flip

Our practice in truth tables has been to list first the valuation where "P" and "Q" are both true, then the valuation where "P" is true and "Q" is false, and so on. The above Dual of Truth Table 1 reverses that order; but **flipping the truth table upside down** restores the traditional order.

P	**Q**	**Dual of Truth Table 1, Flipped**
1	1	0
1	0	0
0	1	0
0	**0**	**1**

Since this flip restores the sentence letter values to the (traditional) order they had prior to the True/False Swap, swapping and then flipping the sentence letter values is equivalent to doing nothing to those sentence letter values. That suggests a shortcut in the True/False swap: **swap 1s and 0s just in the last column** (the truth table for the whole sentence), and then **flip that column**.[4]

Applying this abbreviated True/False Swap to that last example yields just the right results: a sentence only true when both “P” and “Q” are false.

P	**Q**	**Truth Table 1**	**T/F Swap of Truth Table 1 Column**	**Flip of T/F Swap of Truth Table 1 Column**
1	1	**0**	1	**0**
1	0	**1**	0	**0**
0	1	**1**	0	**0**
0	0	**1**	0	**1**

3. Duals of Sentences: The Connective Swap. Since the parallel between construction rules and semantic rules guarantees that each formal sentence has a truth table, extending the True/False Swap to sentences looks easy: **(a)** given a formal sentence, build its truth table; **(b)** apply the True-False Swap to that truth table to get its dual truth table; and then **(c)** find the sentence matching that dual truth table. This matching sentence should count as the dual of the original sentence.

The hiccup comes in the (c) step. Certainly given any truth table we can find a formal sentence taking that truth table. (The DNF Method, for instance, will always supply a sentence to match a given truth table.[5]) The

[4] As noted in (Quine 1982: 80).
[5] As set out in 2.27.

prior example of the True/False Swap is an illustration: since "~(P ∧ Q)" takes Truth Table 1, we could say the dual of "~(P ∧ Q)" is 'the' DNF sentence matching the Dual of Truth Table 1 – namely, "(~P ∧ ~Q)".

~(P ∧ Q) (~P ∧ ~Q)

(a) **(c?)**

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

P	**Q**	**Dual of Truth Table 1**
1	1	0
1	0	0
0	1	0
0	0	1

(b)

But the last step here yields only **a** sentence matching the dual truth table, whereas Step (c) of the instructions directed us to find **the** (one) matching sentence. Certainly "(~P ∧ ~Q)" matches the dual truth table here – but so do "~(P ∨ Q)" and "(~P ∧ (~Q ∨ (P ∧ ~P))" and infinitely many other sentences.[6] **Which sentence** (if any) **counts as the** (one, genuine) **dual** of our original sentence, "~(P ∧ Q)"? This approach offers no answer.

What's needed is a method that picks, among the infinity of sentences taking that dual truth table, one sentence especially qualifying as **the** dual of the original sentence. Guidance here comes in returning to our first example of duality – the semantic rules – and noting that their sentence counterparts are the **construction rules**. Each of the (molecular) construction rules involves adding a single **connective** (with parentheses, as required). Pairing construction rules (in a way that parallels dual semantic rules) therefore involves pairing **connectives**. Just as semantic rules for conjunction and disjunction were paired as duals (and particular truth tables inherited that duality), so **wedge and vel are dual connectives** (and particular sentences inherit that duality, based on their connectives). Likewise, just as the semantic Negation Rule is its own dual, the **tilde is its own dual connective**.

---

[6] To see why there are infinitely many sentences matching the dual truth table, note that since "(~P ∧ ~Q)" matches that truth table, so does the double negation of "(~P ∧ ~Q)", its quadruple negation, and so on.

Swapping dual connectives in this way yields the **Connective Swap dual** of a sentence.

> **Connective Swap**: for a given sentence, its Connective Swap dual is the result of replacing each wedge with a vel, and each vel with a wedge.[7]

The two sorts of duality run in parallel, since the steps in a truth table always mirror the steps of a construction tree, and a construction tree adds only one connective in each step. For example, here's the truth table for the sentence "(~P ∨ (P ∧ Q))".

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

True/False Swap of the truth values yields the following. (I put "T/F" next to each sentence, to note that its truth table has undergone True/False Swap.)

True-False Swap of "(~P ∨ (P ∧ Q))"

$P_{T/F}$	$Q_{T/F}$	$\sim P_{T/F}$	$(P \wedge Q)_{T/F}$	$(\sim P \vee (P \wedge Q))_{T/F}$
0	0	1	0	0
0	1	1	1	1
1	0	0	1	0
1	1	0	1	0

[7] The Connective Swap involves replacing each connective with its dual connective. But since the tilde is its own dual, swapping a tilde with its dual amounts to leaving the tilde unchanged.

Flipping these truth tables puts them in the traditional order.

True-False Swap of “(~P ∨ (P ∧ Q))” (Flipped)

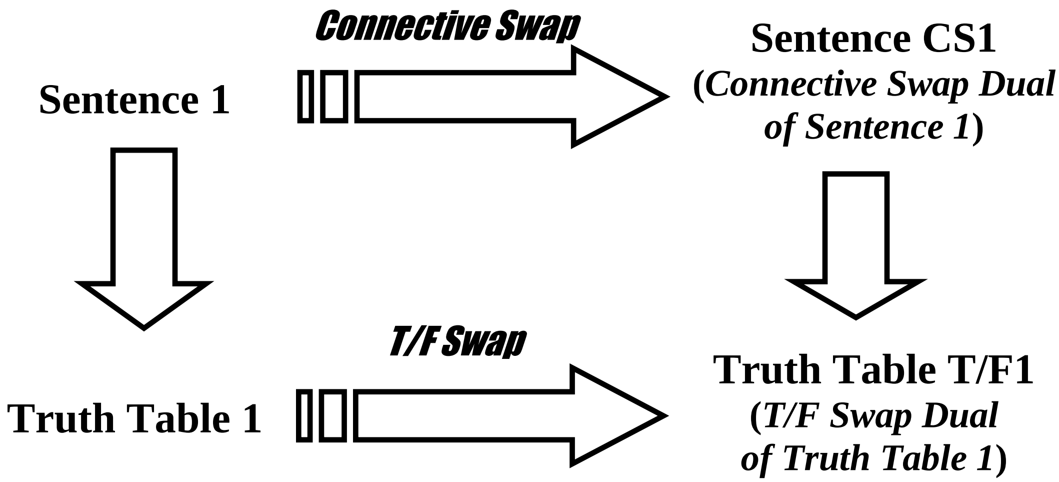
$P_{T/F}$	$Q_{T/F}$	$\sim P_{T/F}$	$(P \wedge Q)_{T/F}$	$(\sim P \vee (P \wedge Q))_{T/F}$
1	1	0	1	0
1	0	0	1	0
0	1	1	1	1
0	0	1	0	0

But each of these truth tables is the truth table for the Connective Swap of the listed sentence.

Connective Swap of “(~P ∨ (P ∧ Q))”

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

In general, True/False Swap for truth tables is **bound** to shadow Connective Swap for sentences. A formal sentence is assigned a truth table by the semantic rules; and the True/False Swap dual of that truth table is exactly the truth table taken by the Connective Swap dual of the original sentence.[8]



The Connective Swap therefore exhibits both the features we want for sentence duality: (1) the Connective Swap dual of a sentence takes the same truth table we’d get by performing the True/False Swap on that sentence’s truth table, so **sentence duality always agrees with semantic duality**. And (2) each sentence has one and only one Connective Swap dual.

Moreover, since connectives are paired in Connective Swap (just as truth values are paired in True/False Swap), Connective Swap is **involutary**, just like True/False Swap: applying Connective Swap twice to a sentence just yields that sentence again. For instance, the Connective Swap of “$\sim(P \wedge Q)$” is“$\sim(P \vee Q)$”; but the Connective Swap of “$\sim(P \vee Q)$” is just “$\sim(P \wedge Q)$” again.[9]

[8] It’s no coincidence that the True/False Swap and Connective Swap duals run in parallel. For when figuring out sentence (and connective) duality, the most basic condition laid upon it was that it parallel the True/False Swap. (Any candidate for ‘dual of a sentence’ that didn’t faithfully mirror True/False Swap duality would have been rejected.)

[9] Because both types of duality are involutary, the horizontal arrows in the above diagram in fact go both ways. But since a truth table matches infinitely many sentences, the vertical arrows don’t move upward. (Technically: there is a **homomorphism** from sentences to truth tables. Semantic rules map each sentence onto exactly one truth table, but each truth table maps onto many different sentences.)

Thus we end up with two, parallel, sorts of duality: the True/False Swap duality of truth tables – call it "**semantic duality**" for short; and the Connective Swap duality of formal sentences – call this "**connective duality**" for short. And because the Connective Swap was built to match the True/False Swap, we know that **connective duality always brings semantic duality in its wake**: if Sentence 1 is the connective dual of Sentence 2 (and vice versa), then the truth table for Sentence 1 is the semantic dual of the truth table for Sentence 2 (and vice versa).

But the reverse is not the case: the truth table for Sentence 1 might be the semantic dual of the truth table for Sentence 2, even though Sentence 1 isn't the connective dual of Sentence 2. For instance, the truth table for "$\sim(\sim P \vee \sim Q)$" is the **semantic** dual of the truth table for "$(P \vee Q)$".

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

Yet "$\sim(\sim P \vee \sim Q)$" isn't the **connective** dual of "$(P \vee Q)$".

Still, we can extend semantic duality to sentences in a weaker way, saying that "$\sim(\sim P \vee \sim Q)$" and "$(P \wedge Q)$" are each **a** semantic dual of "$(P \vee Q)$" – since their (shared) truth table is the dual of the truth table for "$(P \vee Q)$". Every sentence will then have exactly one **connective dual** sentence, and an infinite family of **semantic dual** sentences (all logically equivalent to that connective dual sentence).

> A **semantic dual** of Sentence S: a sentence logically equivalent to the connective dual of Sentence S

From this definition alone it follows that **if two sentences are connective duals, they will be semantic duals** (though not necessarily vice versa, as we've seen).[10]

[10] The set of connective duals of Sentence S is thus a **proper subset** of the set of semantic duals of S (to put it mildly).

Summary: Duals of Truth Tables, Duals of Sentences

Semantic Duals of Truth Tables: The True/False Swap

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

Truth Table Shortcut: The Flip

- To quickly get the True/False swap for a truth table, apply the True/False Swap only to the truth table column in question (not, e.g., to the sentence letter columns of the truth table), then flip that column upside down.

Connective Duals of Sentences: The Connective Swap

- For a given sentence, its **Connective Swap dual** (its **connective dual**) is obtained by replacing each vel in that sentence by a wedge, and each wedge by a vel.

Semantic Duals of Sentences

- For a given sentence S, any sentence logically equivalent to the connective dual of S counts as **a semantic dual** of sentence S.