3.32. Sentence Analysis: Truth Trees Without Truth (And Without Trees)

1. Sentence Analysis. Recall that moving from truth trees to analytic trees reduced the number of rules needed; for the True and False Negation rules of old were replaced by the single Double Negation rule of analytic trees. Here we modify analytic trees in turn, in a way that again reduces the number of rules required. We will find that, quite apart from this added economy, the resulting system sheds surprising new light on the original truth tree test. We will call this latest approach the method of **sentence analysis**.

Key to sentence analysis is an understanding of tree paths and branches. In analytic trees we follow each path from bottom to top, bubble-like, looking only at the *basics* on that path. A path is deemed 'semantically impossible' if it contains some sentence letter and also its negation. That's because we consider a tree path as a **set of sentences**: the set of just the sentences on that path. Naturally, a set of sentences containing a sentence letter and its negation will be unsatisfiable, hence inconsistent.

Central to the method of sentence analysis will be **translating** this **set of sentences into a single sentence**. The natural sentence counterpart to a set of sentences is the **conjunction** of those sentences. For just as a set of sentences is consistent only if some valuation satisfies all its members, so the conjunction of those sentences is consistent only if some valuation makes all it parts true.

So the **counterexample set** for an argument has, as its sentence counterpart, the conjunction of the premises and negation of the conclusion – call it the "**counterexample sentence**" for that argument. The following argument thus begins in sentence analysis like so.

1. ~(P ∧ Q) ――――― ∴ (~P ∨ ~Q)	**1. ~(P ∧ Q) ∧ ~(~P ∨ ~Q)**

And instead of tree rules we now use rules for rewriting some part of this initial conjunction. As will soon become clear, these rules are just the rewrite rules used in the earlier **3D Method**.[1]

For example: the analytic tree rule for a **negated disjunction** traded in a negated disjunction (on a tree path) for the negation of both its parts (on that same tree path). That rule is now recast as **conjoining** both those negated parts (in the same spot held previously by the original negated disjunction).

Negated Disjunction	**Negated Disjunction**
✓ $\sim(\bullet \vee \blacktriangle)$	(1) $\blacklozenge \wedge \underline{\sim(\bullet \vee \blacktriangle)} \wedge \heartsuit$
|	(2) $\blacklozenge \wedge \underline{(\sim\bullet \wedge \sim\blacktriangle)} \wedge \heartsuit$
$\sim\bullet$	
$\sim\blacktriangle$	

(The ◆ and ♥ on either side allow other sentences to be conjoined to the negated disjunction. Those further sentences are left unchanged when applying the rule. But note that the rule doesn't *require* there to be sentences on the left and right sides of the one we're looking at. We may have no sentence on the left of what we're analyzing, or no sentence on the right – or no sentence on either side.)

So in our argument the negated disjunction "~(~P ∨ ~Q)" is analyzed like so. (We freely dispense with inessential parentheses – yielding, on Line 2, a triple-barreled conjunction.)

$\sim(P \wedge Q)$	**1.** $\mathbf{\sim(P \wedge Q) \wedge \underline{\sim(\sim P \vee \sim Q)}}$
✓ $\sim(\sim P \vee \sim Q)$	**2.** $\mathbf{\sim(P \wedge Q) \wedge \underline{\sim\sim P} \wedge \underline{\sim\sim Q}}$
|	
✓ $\sim\sim P$	
✓ $\sim\sim Q$	

Obviously the Negated Disjunction rule is just an application of **De Morgan's Law**: "~(P ∨ Q)" is equivalent to " (~P ∧ ~Q)".

---

[1] Introduced in "*3.28. From Formal Language to DNF: The 3D Method*".

The Double Negation Rule in analytic trees left its output *on the same tree path*. In the same way, sentence analysis of a double negation leaves its output sentence as part of the same larger conjunction of sentences.

Double Negation

✓ ~~●
|
●

Double Negation

(1) $\blacklozenge \wedge \underline{\sim\sim\bullet} \wedge \heartsuit$
(2) $\blacklozenge \wedge \underline{\bullet} \wedge \heartsuit$

We thus replace each double negation in our argument with its counterpart.

~(P ∧ Q)
✓ ~(~P ∨ ~Q)
|
✓ ~~P
✓ ~~Q
P
Q

1. $\sim(P \wedge Q) \wedge \sim(\sim P \vee \sim Q)$
2. $\sim(P \wedge Q) \wedge \underline{\sim\sim P} \wedge \underline{\sim\sim Q}$
3. $\sim(P \wedge Q) \wedge \underline{P} \wedge \sim\sim Q$
4. $\sim(P \wedge Q) \wedge P \wedge \underline{Q}$

(Recall that in analytic trees we had no need for a rule breaking down singly-negated sentence letters such as “~P”; for as **basics**, they marked the end of analysis just as much as sentence letters did. Since sentence analysis likewise takes basics as fundament parts, here too we have no use for any rule treating single-negations of sentence letters.

Our rule for Negated Conjunctions enacts another form of DeMorgan’s Law: “~(P ∧ Q)” is equivalent to “ (~P ∨ ~Q)”.

Negated Conjunction

✓ ~(● ∧ ▲)

~●   ~▲

Negated Conjunction

(1) $\blacklozenge \wedge \underline{\sim(\bullet \wedge \blacktriangle)} \wedge \heartsuit$
(2) $\blacklozenge \wedge \underline{(\sim\bullet \vee \sim\blacktriangle)} \wedge \heartsuit$

(Note that in sentence analysis **the sentence counterpart to a branch is a disjunction**. For already in truth trees, when introducing a branch for a true disjunction, we said: if we're in a situation where the whole disjunction is true, we're *either* in a situation where the left part is true, *or* in a situation where the right part is true – *possibly both*. A branch was, all along, just a more graphic depiction of an inclusive disjunction.)

~(P ∧ Q)
✓ ~(~P ∨ ~Q)
✓ ~~P
✓ ~~Q
P
Q
~P ~Q

1. $\mathbf{{\sim}(P \wedge Q) \wedge \underline{{\sim}({\sim}P \vee {\sim}Q)}}$
2. $\mathbf{{\sim}(P \wedge Q) \wedge \underline{{\sim}{\sim}P} \wedge \underline{{\sim}{\sim}Q}}$
3. $\mathbf{{\sim}(P \wedge Q) \wedge \underline{P} \wedge {\sim}{\sim}Q}$
4. $\mathbf{{\sim}(P \wedge Q) \wedge P \wedge \underline{Q}}$
5. $\mathbf{({\sim}P \vee {\sim}Q) \wedge \underline{{\sim}{\sim}P} \wedge \underline{{\sim}{\sim}Q}}$

Recall that in analytic trees we had no need for a rule breaking down (singly-) negated sentence letters such as "~P"; for as **basics**, they marked the end of analysis just as much as sentence letters did. Since sentence analysis likewise takes basics as fundament parts, here too we have no use for any rule treating (single-) negations of sentence letters.

Another rule from analytic trees left parts of a sentence on the same line as the whole: the Conjunction Rule.

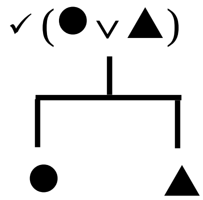
Conjunction

✓ (● ∧ ▲)
●
▲

But sentence analysis allows us to discard this rule as well. Since conjunctions now serve as tree paths, we have no need to break them down.

The one remaining rule from analytic trees – the Disjunction Rule – introduced a branch into the tree.

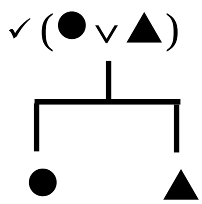
Disjunction



Here again, the sentence counterpart to a branch will be a disjunction.

The Disjunction Rule conjoins each part of the disjunction to the sentences (on the left and right) that the original disjunction was conjoined with.

Disjunction



Disjunction

$$(1)\ \blacklozenge \wedge \underline{(\bullet \vee \blacktriangle)} \wedge \heartsuit$$
$$(2)\ (\blacklozenge \wedge \underline{\bullet} \wedge \heartsuit) \vee (\blacklozenge \wedge \underline{\blacktriangle} \wedge \heartsuit)$$

Should that rule seem mysterious, matters are clearer when we recognize that this ‘branching’ rule is really just two applications of **distribution**.

Distribution

$$“(\underline{\bullet \wedge} (\blacktriangle \vee \blacklozenge))” \equiv “((\underline{\bullet \wedge} \blacktriangle) \vee (\underline{\bullet \wedge} \blacklozenge))”$$

Suppose, for example, we analyze this disjunction.

$$1.\ P \wedge \underline{(\mathbf{Q} \vee \mathbf{R})} \wedge S$$

The first round of distribution attaches “P ∧” onto each part of “(Q ∨ R)”.

$$1.\ \underline{P \wedge}(\mathbf{Q} \vee \mathbf{R}) \wedge S$$
$$2.\ ((\underline{P \wedge}\mathbf{Q}) \vee (\underline{P \wedge}\mathbf{R})) \wedge S$$

A second round of distribution – attaching “∧ S” to each part of the disjunction on Line 2 – yields the same result as our disjunction rule.

$$2.\ ((P \wedge \mathbf{Q}) \vee (P \wedge \mathbf{R}))\ \underline{\wedge S}$$
$$3.\ (P \wedge \mathbf{Q}\ \underline{\wedge S}) \vee (P \wedge \mathbf{R}\ \underline{\wedge S})$$

(If there had been only a left part or a right conjoined to the sentence being analyzed, just one round of distribution would be needed.)

In the next example only a left part, “P,” is conjoined to “(Q ∨ R)” (because there is no further part of the sentence to the right of “(Q ∨ R)”).

$$1.\ \underline{P \wedge}(\mathbf{Q} \vee \mathbf{R})$$
$$2.\ (\underline{P \wedge}\mathbf{Q}) \vee (\underline{P \wedge}\mathbf{R})$$

***Stopped here 10.12.14

Returning to our argument: we analyze the negated conjunction “~(P ∧ Q)” by distributing the right part of the conjunction, “(P ∧ Q),” onto “~P” and “~Q”.

✓ ~(P ∧ Q)
✓ ~(~P ∨ ~Q)

✓ ~~P
✓ ~~Q
P
Q

~P ~Q

$$\mathbf{1.\ {\sim}(P \wedge Q) \wedge {\sim}({\sim}P \vee {\sim}Q)}$$
$$\mathbf{2.\ {\sim}(P \wedge Q) \wedge {\sim}{\sim}P \wedge {\sim}{\sim}Q}$$
$$\mathbf{3.\ {\sim}(P \wedge Q) \wedge \underline{P} \wedge {\sim}{\sim}Q}$$
$$\mathbf{4.\ {\sim}(P \wedge Q)\ \underline{\wedge P \wedge Q}}$$
$$\mathbf{5.\ ({\sim}P\ \underline{\wedge P \wedge Q}) \vee ({\sim}Q\ \underline{\wedge P \wedge Q})}$$

This yields the right result: two ‘branches’ (the two parts of the disjunction) – one containing {~P, P, Q}, the other containing {~Q, P, Q}.

And just as the analytic tree at this point closes on each path – the left for containing {~P, P}, the right for containing {~Q, Q} – we see now that sentence analysis ‘closes’ as well. For each half of this disjunction is a contradiction – the left because it’s a conjunction with both “P” and “~P” as parts, the right for being a conjunction with “Q” and “~Q” as parts

$$
\begin{gathered}
\mathbf{1.}\ {\sim}(P \wedge Q) \wedge {\sim}({\sim}P \vee {\sim}Q) \\
\mathbf{2.}\ {\sim}(P \wedge Q) \wedge {\sim\sim}P \wedge {\sim\sim}Q \\
\mathbf{3.}\ {\sim}(P \wedge Q) \wedge P \wedge {\sim\sim}Q \\
\mathbf{4.}\ {\sim}(P \wedge Q) \wedge P \wedge Q \\
\mathbf{5.}\ \cancel{({\sim}P \wedge P \wedge Q)} \vee \cancel{({\sim}Q \wedge P \wedge Q)}
\end{gathered}
$$

Since each part of the disjunction is inconsistent, the whole disjunction is as well.[2] And since the **counterexample sentence** (the sentence counterpart to the argument’s counterexample set) **is inconsistent**, the argument is **valid**.

A familiar valid argument of old provides another illustration of sentence analysis.

1. (P ∨ Q)
2. ~P

____

∴ Q

$$
\begin{gathered}
\mathbf{1.}\ (P \vee Q)\,\underline{\wedge {\sim}P \wedge {\sim}Q} \\
\mathbf{2.}\ (P\,\underline{\wedge {\sim}P \wedge {\sim}Q}) \vee (Q\,\underline{\wedge {\sim}P \wedge {\sim}Q})
\end{gathered}
$$

(In Sentence (1) there is only a right part, “(~P ∧ ~Q),” attached to the disjunction being analyzed; so only one round of distribution is needed to apply the Disjunction Rule.)

Since each part of the disjunction is inconsistent, sentence analysis concludes that **the argument is valid**.

$$
\begin{gathered}
\mathbf{1.}\ (P \vee Q) \wedge {\sim}P \wedge {\sim}Q \\
\mathbf{2.}\ (\cancel{P \wedge {\sim}P \wedge {\sim}Q}) \vee (\cancel{Q \wedge {\sim}P \wedge {\sim}Q})
\end{gathered}
$$

An equally familiar cousin of that last argument illustrates how sentence analysis treats an invalid argument.

---

[2] A point about disjunctions noted in “*3.28. Conjunctive and Disjunctive Normal Forms*”.

1. $(P \vee Q)$
2. P

$\therefore Q$

1. $(P \vee Q) \wedge P \wedge \sim Q$
2. $(P \wedge P \wedge \sim Q) \vee \cancel{(Q \wedge P \wedge \sim Q)}$

While the right conjunction is inconsistent (thanks to its parts "Q" and "~Q"), the left conjunction is perfectly consistent. Since at least one of its parts is consistent, the whole disjunction is consistent.[3] And since Sentence (2) is just (an analyzed version of) the argument's counterexample sentence, that counterexample sentence is consistent, and the argument is **invalid**. Here, as in trees methods, if even one path (here: conjunction-of-basics) stays open all the way to the end, the argument is invalid.

2. Semantic Morals. As has been stressed throughout, the method of sentence analysis is an obvious variation on analytic trees; and those were in turn a non-semantic counterpart to truth trees. The same morals that held of truth trees thus apply to these later methods as well. Specifically: an argument is valid if (and only if) any (alleged) validity counterexample for that argument is inconsistent.

Perhaps less obvious is what sentence analysis shows us about the truth tree method. Consider: in sentence analysis an initial conjunction of sentences is transformed into a different sentence – either a conjunction of basics, or a disjunction each of whose parts is such a conjunction of basics. But that describes a sentence in **Disjunctive Normal Form (DNF)**. Indeed, as noted earlier, the rules of sentence analysis simply repeat the earlier 3D Method, which serves to translate any Chapter Three sentence into DNF: the Double Negation rule applies **Double Negation**; the Negated Disjunction and Negated Conjunction rules apply **De Morgan's Law**; and the Disjunction rule applies **Distribution**.

Now we see why, in sentence analysis, the final analyzed sentence is inconsistent if (and only if) all of its parts are: because any DNF sentence is inconsistent if (and only if) all of its 'cells' are[4], and sentence analysis is just the translation of the original counterexample sentence into DNF.

[3] Because a disjunction is satisfiable – consistent – as long as least one of its parts is.
[4] As noted in "*3.29. Conjunctive and Disjunctive Normal Forms*".

But in light of our earlier observation – that sentence analysis is ultimately just a variation on the truth tree method – this highlights an important point about truth trees as well: **truth trees were all along just** a more graphic way of **translating sentences into Disjunctive Normal Form**. Though we already understood the link between validity and inconsistency in truth trees[5], we now see that truth trees reflect the link between inconsistency and DNF: a DNF sentence is inconsistent if (and only if) each of its cells is inconsistent.

We understand now why an argument is valid if (and only if) it closes every truth tree path: a tree path closes just when it's inconsistent, and a tree path is in effect just a cell of a DNF sentence. (That's why we only looked at sentence letters or basics when diagnosing the consistency of a tree branch: a DNF cell is inconsistent if and only if it contains a sentence letter and its negation.) The truth tree test of validity is just the truth tree test of consistency (applied to a counterexample set); and now we add that the truth tree test of consistency is in turn just the DNF test of consistency. Connecting the dots: **the truth tree test of validity is just the DNF test of consistency**. A deeper understanding of trees is achieved, thanks to sentence analysis and Disjunctive Normal Form.

[5] As noted at the end of "*3.25. Truth Trees: Tautology, Contradiction, and Logical Equivalence*," following on the main point of the earlier "*3.20. Validity and Inconsistency*".