Expressive Completeness and Disjunctive Normal Form

Expressive Completeness Theorem: Every truth-function can be expressed by a sentence of SL that contains no sentential connectives other than {~, ∨, · }.

In other words, any possible truth table for any possible connective has an equivalent truth table where the only connectives are {~, ∨, · }. As an example of this theorem, take our connective '⊃'. In order to show that {~, ∨, · } can be used, in conjunction with our sentence letters (P, Q, R, etc.), to eliminate ⊃ we need to prove that there is an argument with exactly the same truth table as ⊃. In other words, we need a truth table having only P and Q as sentences letters and the connectives {~, ∨, · } where the second line is F, and the first, third, and fourth lines are T. Thankfully, the argument is quite simple:

$$P \supset Q \dashv \vDash {\sim}P \vee Q$$

Hence, we can do all of our logic without the horseshoe. The important thing isn't, though, that we can eliminate connectives, but that with our sentence letters and {~, ∨, · } we can create *any* truth table. 2 rows, 4 rows, 8 rows, or trillion row tables can all be created with those 3 connectives. Pretty cool, huh?

Let's create a truth table for a connective that we haven't defined yet.[1]

p	q	p⊕q
T	T	F
T	*F*	*T*
F	*T*	*T*
F	F	F

For this two-place connective, symbolized by the ⊕, when both inputs are T or F, the output is F, and T when exactly one input is F and the other T (the opposite of the triple bar). According to our theorem, we *must* be able to construct a truth table using just {~, ∨, · } to replace ⊕. What does such an argument look like?[2]

---

[1] Remember: all connectives are defined in terms of their truth tables. Just because a connective doesn't match an English word or phrase doesn't mean it doesn't exist. Every possible truth table has an associated connective - even if we don't care about this in terms of English conversation. Since there are an infinite number of truth tables, there must also be an infinite number of connectives.

[2] Please note that the ^ symbol (the caret) here is just another way some logic textbooks symbolize conjunction. The ^ is the same as our · .

$(p \wedge \neg q) \vee (\neg p \wedge q)$
F
T
T
F

Unlike ⊃, creating an equivalent argument required that we create two arguments and connect them via the ∨. So, we have created a *disjunction* of *conjunctions*. This is an example of *disjunctive normal form*.

> **Disjunctive Normal Form (DNF) Theorem:** every compound / complex proposition which assigns T , Φ, can be put in disjunctive normal form.[3,4]

Hence, since the Expressive Adequacy Theorem is true of our logic, we *must* be able to produce (almost, footnote 4) any connective's truth table in DNF. Creating the disjunctive normal form of an argument is quite simple.

1. Construct the truth table for Φ (above, we were using P⊕Q).
2. Isolate each row where Φ receives a T (for ⊕, it was the second and third lines)
3. For each row which receives a T, create one argument using ONLY the sentences of that row, and {~, ・} which also outputs T for that row *alone*. (if you are creative enough to come up with one argument that outputs T for only the lines in the original argument which get a T, then that is fine - it's just rare!)
4. Take each individual argument that you have constructed, and join them via {∨}.

Step 3 is usually the most difficult. Again, look at the P⊕Q table. We can create an argument using {~, ・} to make line 2 true. It is simply (P ・ ~Q) because when P is true and Q is false the argument comes out as true. Then, looking at row 3, we can make an equivalent argument for that by constructing the argument (~P ・ Q) because when P is false and Q is true, the conjunction will come out true. Then, since those are the only two rows where the output is T, we connect the two arguments with a ∨. That is how to create the DNF equivalent of (almost) any connective.

---

[3] Since we have been using capital English letters to represent sentences I didn't want to use P or Q as the variable to represent any argument involving a connective. Hence, I am using the Greek letter Φ (pronounced: fī).

[4] It is worth noting that we are glossing over whether or not a sentence letter can appear more than once within any given disjunct. Depending on how you view this requirement actually determines whether *every* truth function can be symbolized - but we won't be looking at this issue in our discussion.