3.9. Further Formal Languages

Here we explore the expressive power of various formal languages beyond the Chapters Two and Three languages – some very lean, some absurdly over-stuffed. We do this not for practical purposes of translating English – we wouldn't, for example, want to stop using "$\rightarrow$" to translate "if... then" in favor of some thicket of the symbols discussed below. Instead we're exploring what's even theoretically possible for a formal language – rather like the engineering challenge of seeing how much could be thrown off a car and still keep it running. (The seat cushions? the shock absorbers? the fuel pump?)

Recall that saying a formal language is expressively adequate means: for every possible truth table, the language has some sentence taking that truth table. And if a language has a sentence matching a certain truth table we say that language "**covers**" that truth table. So **an expressively adequate language is a language which covers every possible truth table**.

We know the Chapter Two language $\{\sim, \wedge, \vee\}$ is expressively adequate; because we showed that DNF covers every truth table, and DNF is a subset of the full Chapter Two language. Likewise CNF is expressively adequate, and it too is a subset of the Chapter Two language.[1]

But we also know from previous exercises that $\{\sim, \wedge, \vee\}$ has 'connective overkill' – that there's at least one connective we could throw out of $\{\sim, \wedge, \vee\}$ while still leaving an expressively adequate language.

If a language loses no truth table coverage by throwing out a certain connective, then **that connective is redundant**. And a **formal language is redundant** if even one of its connectives is redundant. So since $\{\sim, \wedge\}$, and $\{\sim, \vee\}$ are both expressively adequate, **$\{\sim, \wedge, \vee\}$ is redundant**.[2] We could throw out either wedge or vel, and the resulting language would still cover all the same truth tables as $\{\sim, \wedge, \vee\}$.

We can put the point more technically in terms of **expressive equivalence**. Two languages are expressively equivalent if they cover the same set of truth tables.

[1] DNF is discussed in 2.27; CNF is discussed in In 2.29.
[2] Discussed in 2.27 §1.

And a language – a set of connectives – is redundant if it has a **proper subset** that's **expressively equivalent** to that language. For instance, {~, ∧, ∨} and {~, ∧} are expressively equivalent – in symbols, **{~, ∧, ∨} ≈ {~, ∧}** – since they cover the same set of truth tables (namely, the set of all possible truth tables). And since {~, ∧} is a proper subset of {~, ∧, ∨}, that makes {~, ∧, ∨} redundant.

Because the Chapter Two language is already redundant, the (expressively equivalent) Chapter Three language {~, ∧, ∨, →, ↔} is all the more so redundant. That makes clear that the connectives new to Chapter Three were added only for ease of translation, not out of any need to cover otherwise neglected truth tables. There are Chapter Two sentences logically equivalent to a conditional or biconditional.

$$(P \rightarrow Q) \equiv \sim(P \wedge \sim Q) \equiv (\sim P \vee Q)$$
$$(P \leftrightarrow Q) \equiv (\sim(P \wedge \sim Q) \wedge \sim(Q \wedge \sim P)) \equiv \sim(\sim(\sim P \vee Q) \vee \sim(P \vee Q))$$

It's just simpler and more natural-looking to translate "If Trixie won the prize then she's buying a hot tub" as "(P → Q)" than as "~(P ∧ ~Q)".

The Chapter Three language {~, ∧, ∨, →, ↔} is redundant, since all of the following subsets are expressively adequate.

{~, ∧}   {~, ∨}   {~, →}

By contrast, none of these smaller languages are redundant. Removing any connective would result in an expressively inadequate language.

In particular: removing the tilde from any of these languages leads to expressive inadequacy. For even the full Chapter Three language yields an inadequate language if we throw out the tilde: {∧, ∨, →, ↔}.

The tilde might, then, seem essential for any expressively adequate set of connectives. But we'll see that's not so – once we move beyond the connectives of the Chapter Three language.

Hence we turn to further connectives, asking what results if we build formal languages with some or all of these. Besides our experiments in how lean a language can get, we thus also explore adding further connectives – in the end, going on a kind of connective shopping spree.

For instance, early in Chapter Two we considered an **exclusive disjunction** such as the following.

We're having either absinthe or aspic, but not both.

The Chapter Two language already had the resources to translate this sentence.

$((P \vee Q) \wedge \sim(P \wedge Q))$

But let's now introduce a connective devoted just to expressing such an **exclusive "or"** – call it **exor**.

●	▲	(● ⊕ ▲)
1	1	0
1	0	1
0	1	1
0	0	0

The language {⊕} is **expressively inadequate**. Indeed, even adding a tilde won't yield an expressively adequate language; for other than truth tables for basics, no {~, ⊕} truth table has an odd number of 1s or an odd number of 0s.[3] The truth tables for, e.g., "(P ∧ Q)" and "(P ∨ Q)" are thus not covered by {~, ⊕}.

Still, it's interesting to note what {~, ⊕} can do. For one thing: **the negation of an exclusive disjunction is equivalent to a biconditional**.

P	**Q**	**(P ⊕ Q)**	**~(P ⊕ Q)**	**(P ↔ Q)**
1	1	0	**1**	**1**
1	0	1	**0**	**0**
0	1	1	**0**	**0**
0	0	0	**1**	**1**

So {~, ⊕, ↔} ≈ {~, ⊕}. Since adding the bicon to {~, ⊕} doesn't cover any new truth tables, {~, ⊕, ↔} is an (inadequate and) **redundant** language. And since the

[3] Recall (from 2.26) that a **basic** is any sentence letter or negation of a sentence letter

negation of a biconditional is likewise equivalent to an exclusive disjunction, $\{\sim, \oplus\} \approx \{\sim, \leftrightarrow\}$.

In fact we can show the following three-way equivalence.

$$\{\sim, \oplus\} \approx \{\sim, \leftrightarrow\} \approx \{\leftrightarrow, \oplus\}$$

Notice in particular: the language $\{\leftrightarrow, \oplus\}$ can **cover the truth table for the negation** "~P". For the first time we encounter a language which can **cover negation truth tables without using a tilde**.

P	(P ⊕ P)	**(P ↔ (P ⊕ P))**	**~P**
1	0	**0**	**0**
0	0	**1**	**1**

Another example comes in the language $\{\rightarrow, \oplus\}$, which also covers the truth table for "~P".

P	(P ⊕ P)	**(P → (P ⊕ P))**	**~P**
1	0	**0**	**0**
0	0	**1**	**1**

Since$\{\rightarrow, \sim\}$ is expressively adequate, so is $\{\rightarrow, \oplus\}$.

A single connective covering the negation truth table without a tilde is one translating a "neither... nor" claim: the "↓" symbol, called the **dagger**.[4]

●	▲	(● ↓ ▲)
1	1	0
1	0	0
0	1	0
0	**0**	**1**

[4] As a memory aid: **d**ownward **d**agger **d**enies a **d**isjunction. In electrical engineering it's called a "**NOR**". The dagger was explored already by C.S Peirce in (Peirce 1880: 13-18).

Among the striking features of {↓} is that it too covers the truth table for "~P".

●	▲	(● ↓ ▲)
1	1	0
1	0	0
0	1	0
0	0	1

P	(P ↓ P)
1	0
0	1

"~" and "~(P ∨ Q)" together yield the semantic equivalent of "(P ∨ Q)," so {↓} can cover that truth table as well. Since {~, ∨} is expressively adequate, **{↓} is expressively adequate**.

The other **expressively adequate** single connective is the **stroke**, "|,"equivalent to "not both".[5]

●	▲	(● | ▲)
1	**1**	**0**
1	0	1
0	1	1
0	0	1

Because {|} covers both negation and conjunction truth tables, it's equivalent to the expressively adequate language {∧, ~}.

Next we add a connective to express a sentence such as "(P ∧ ~Q)".

●	▲	(● % ▲)
1	1	0
1	**0**	**1**
0	1	0
0	0	0

[5] In electrical engineering this is called a "**NAND**," on analogy with NOR.

We use the symbol "**%**" because "(P ∧ ~Q)" translates "P **without** Q," and "**%**" looks like "**w/o**". For that reason we call this symbol the **wo.**[6]

{**%**} by itself is expressively **inadequate**. But combined with other connectives it yields an expressively adequate language. Most obviously: {~, **%**} is expressively adequate. Note that "~(P **%** Q)" is logically equivalent to "(P → Q)".

P	**Q**	**(P % Q)**	**~(P % Q)**	**(P → Q)**
1	1	0	**1**	**1**
1	0	1	**0**	**0**
0	1	0	**1**	**1**
0	0	0	**1**	**1**

Since {~, →} is **expressively adequate**, **{~, %}** is as well.

We can also show that **{%, ↔}, {→, %}, and {→, ⊕} are expressively adequate**. For example: "(P ↔ (P **%** Q))" is equivalent to "(P | Q)," and we already know { | } to be expressively adequate. So {**%**, ↔} is expressively adequate.

P	**Q**	**(P % Q)**	**(P ↔ (P % Q))**	**(P | Q)**
1	1	0	**0**	**0**
1	0	1	**1**	**1**
0	1	0	**1**	**1**
0	0	0	**1**	**1**

We saw earlier that {↔, ⊕} can cover the truth table for "~P". So any connective which yields an expressively adequate language combined with a tilde will also be part of an adequate language when partnered with ↔ and ⊕.

[6] A wo sentence is sometimes called "asymmetric difference".

Since {~, ∧}, {~, ∨}, {~, →}, and {~, **%**} are all expressively adequate, the following four languages are all **expressively adequate** as well.

$\{\underline{\leftrightarrow, \oplus}, \wedge\}$	$\{\underline{\leftrightarrow, \oplus}, \rightarrow\}$
$\{\underline{\leftrightarrow, \oplus}, \vee\}$	$\{\underline{\leftrightarrow, \oplus}, \textbf{\%}\}$

The right two are **redundant**, since {⊕, →} and {↔, **%**} are already expressively adequate. But {↔, ⊕, ∧} and {↔, ⊕, ∧} **aren't redundant**; throwing any connective out of one of these languages results in expressive inadequacy.

We round out our connective roundup with two unusual additions. The first is the **tee**, a one-connective equivalent of a tautology such as "(P ∨ ~P)".

●	T
1	1
0	1

While the tilde is a one-place connective, and our other connectives so far are all two-place connectives, the **tee is a zero-place connective**: its value is 1 no matter what "●" is. Indeed, we could write the semantic rule for tee like this.[7]

T
1

Any language that can build a tautology can cover the truth table for "T". For example, since "(P → P)" is a tautology, {→} covers the truth table for "T". Hence it would be **redundant** to add "T" to any language with an arrow.

[7] Note that this follows the '2 rule' for number of valuations needed (from 2.12): for **N** many sentence letters we need $2^{\mathbf{N}}$ valuations to go through all the possibilities. Since a truth table for "T" involves **0** sentence letters, we need $2^{\mathbf{0}} = 1$ valuation to exhaust the possibilities.

But **{% ⊤}** yields a **(non-redundant) expressively adequate** language. For {**%**, ⊤} covers the truth table for "~P", and {%, ~} is expressively adequate.

P	⊤	(⊤ **%** P)
1	1	**0**
0	1	**1**

{⊕, ⊤} can likewise cover the truth table for "~P".

P	⊤	(⊤ ⊕ P)
1	1	**0**
0	1	**1**

So {⊕, ⊤, ~} ≈ {⊕, ⊤}. Now, both languages are **expressively inadequate**. But since the languages {~, ∧}, {~, ∨}, {~, →}, and {~, **%**} are all expressively adequate, so are all the following languages. (The right two are **redundant**, since {⊕, →} and {⊤, **%**} are already expressively adequate.)

{<u>⊕, ⊤</u>, ∧}	{<u>⊕, ⊤</u>, →}
{<u>⊕, ⊤</u>, ∨}	{<u>⊕, ⊤</u>, **%**}

Finally we introduce the symbol "⊥". Like "⊤" it's a zero-placed connective – the one-connective equivalent of a contradiction such as "(P ∧ ~P)".

●	⊥
1	0
0	0

Because "⊥" is an inverted tee, we call it **eet**.[8]

[8] Following (Smullyan 1992: 132). Some logicians use Latin names for these two connectives, calling "⊤" "**verum**" and "⊥" "**falsum**" – e.g. (Gamut 1982/1991 Vol I: 137).

The language {$\rightarrow$, $\perp$} is **expressively adequate**. And while {$\leftrightarrow$, $\perp$} is inadequate, it can cover the truth table for "~P".

P	$\perp$	$(\perp \leftrightarrow P)$
1	0	**0**
0	0	**1**

So since the languages {~, $\wedge$}, {~, $\vee$}, {~, $\rightarrow$}, and {~, **%**} are **expressively adequate**, the following languages are as well.

{$\underline{\leftrightarrow, \perp}$, $\wedge$} {$\underline{\leftrightarrow, \perp}$, $\rightarrow$}
{$\underline{\leftrightarrow, \perp}$, $\vee$} {$\underline{\leftrightarrow, \perp}$, **%**}

(The right two are **redundant**, since {$\perp$, $\rightarrow$} and {$\leftrightarrow$, **%**} are already expressively adequate.)

We can collect all of these connectives into one super-language we'll call **A** (for "all our connectives").

A: { | , $\rightarrow$, $\leftrightarrow$, $\vee$, $\top$, ~, $\perp$, $\wedge$, $\oplus$, **%**, $\downarrow$}

Together these cover all the four-valuation truth tables – either with just a sentence letter (as in tables 8 and 9) or with a single connective.[9] Here is a group portrait, the culmination of our connective spending spree.

$\top$	(P | Q)	$(P \rightarrow Q)$	$(Q \rightarrow P)$	$(P \vee Q)$	$(P \leftrightarrow Q)$	~P	P	Q	~Q	$(P \oplus Q)$	$(P \wedge Q)$	(P % Q)	(Q % P)	$(P \downarrow Q)$	$\perp$
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	0	1	1	1	1	0	1	1	0	0	1	0	0	0	0
1	1	0	1	1	0	0	1	0	1	1	0	1	0	0	0
1	1	1	0	1	0	1	0	1	0	1	0	0	1	0	0
1	1	1	1	0	1	1	0	0	1	0	0	0	0	1	0

[9] We could, if we liked, have a single connective even for tables (8) and (9), by adding the identity operator "**i**" – where "iP" has the same truth table as "P".

And here are all the **expressively adequate, non-redundant languages** built from these connectives.

$\{\downarrow\}$ $\{\mid\}$

$\{\rightarrow, \%\}$

$\{\rightarrow, \sim\}$ $\{\sim, \%\}$

$\{\rightarrow, \oplus\}$ $\{\leftrightarrow, \%\}$

$\{\rightarrow, \bot\}$ $\{\top, \%\}$

$\{\vee, \sim\}$ $\{\sim, \wedge\}$

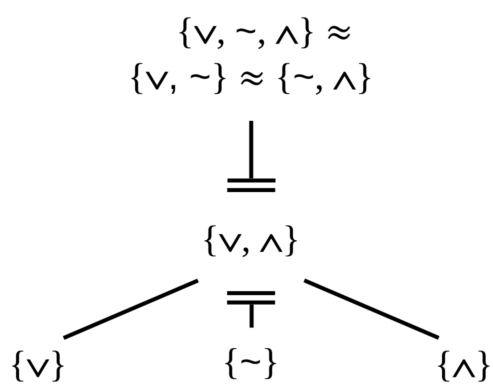
$\{\vee, \leftrightarrow, \oplus\}$ $\{\leftrightarrow, \oplus, \wedge\}$

$\{\vee, \top, \oplus\}$ $\{\leftrightarrow, \bot, \wedge\}$

$\{\vee, \leftrightarrow, \bot\}$ $\{\oplus, \top, \wedge\}$

If we include redundant and expressively inadequate languages, however, the map is considerably bigger.

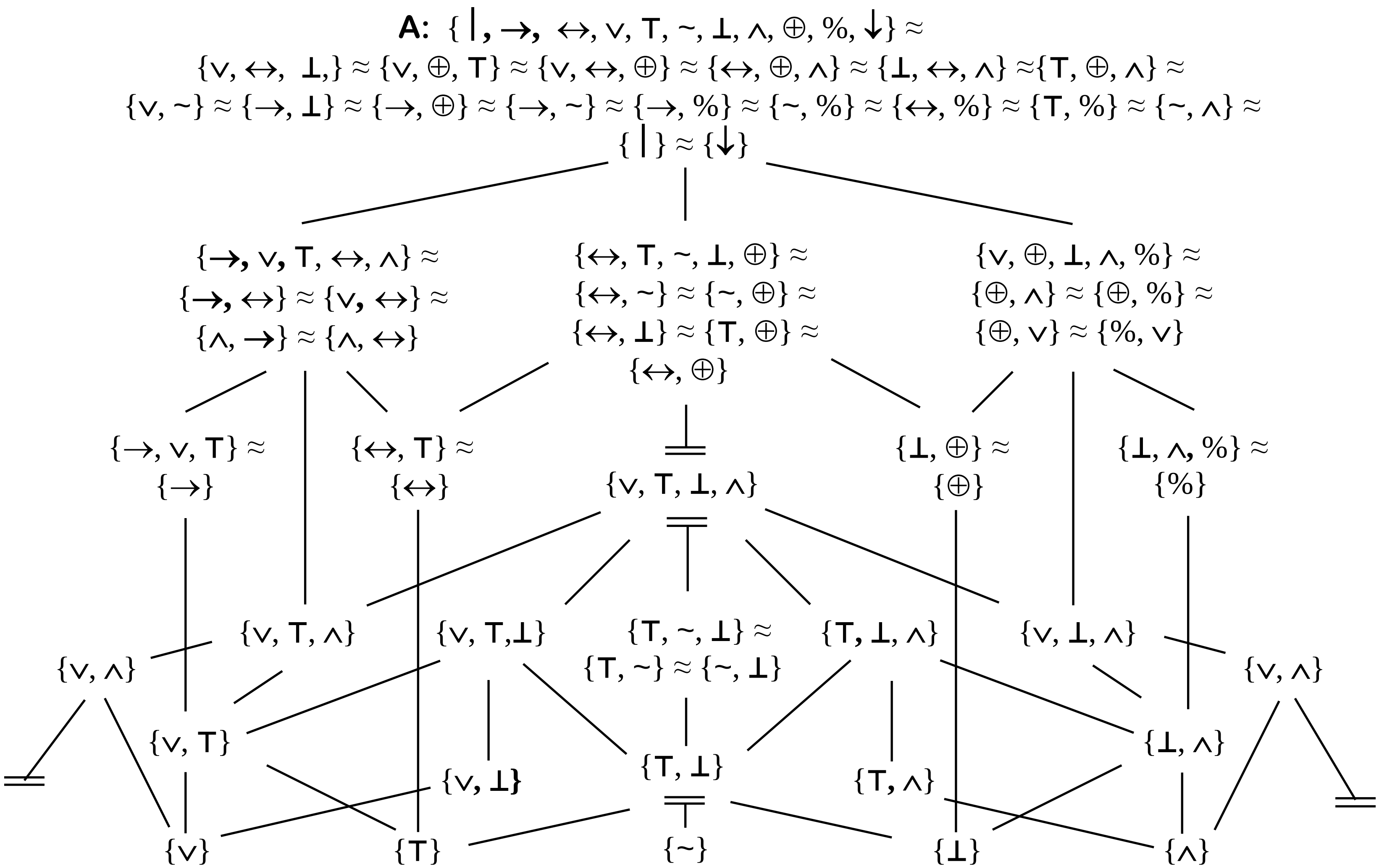
Recall that we mapped the Chapter Two language, and all its sub-languages, like so.[10] (The top three languages are expressively adequate, while all the lower languages are inadequate.)



But the sub-languages of **A** map as follows. Again the top group consists of expressively adequate languages, while all languages below that are inadequate.[11]

[10] In 2.30.

[11] Each group shows only the maximum, redundant language in that group and all its minimal, non-redundant equivalents. The other redundant languages in the group, between these two extremes, aren't listed.



A and Its (Main) Sub-Languages

It’s worth noting that the following are the only **expressively inadequate** 5-connective sub-languages of **A**.

{→, ∨, T, ↔, ∧} {↔, T, ~, ⊥, ⊕} {∨, ⊕, ⊥, ∧, %}

It turns out that **every 5-connective** sub-language of **A** is **redundant**. Indeed, every **4-connective** sub-language of **A** is **redundant** except {∨, T, ⊥, ∧}.

Every sub-language of **A** with **6 or more connectives** is **expressively adequate** and **redundant**.[12]

We note finally that while **A** covers all the four-valuation truth tables, and all the connectives we’re surveyed here, **A** isn’t the set of **all possible connectives**. For we haven’t considered connectives that are 3-place, 4-place, and so on. For instance: while we translated a sentence of the form “If P then Q; otherwise R” as “((P → Q) ∧ (~P→ R)),” we could instead introduce a single connective for such translation.

(P **#** Q R) If P then Q; otherwise R

The larger discussion, of which sorts of languages are expressively adequate and which not, must rely on general principles of expressive adequacy discussed elsewhere.[13]

---

[12] See Appendix X on how to prove these points.

[13] See 3.X. See also Problems X and Y in 3.X.1. on the adequacy of “if… then; otherwise” in particular.