

PROPOSITIONAL LOGIC

① Try to answer the following:

"The longest day of the year (in terms of number of daylight hours) occurs during which month?"

It's likely you might have a reflex answer to this question:

June?



But then we might follow up with the question:

"In Australia, the longest day of the year occurs during which month?"

Oh, I see.



② At this point, some reasoning is required.

Australia is in the Southern hemisphere.

In the Southern hemisphere, the summer solstice occurs in December.

The summer solstice is the longest day of the year.

Thus, in Australia, the longest day of the year is in December.

December!



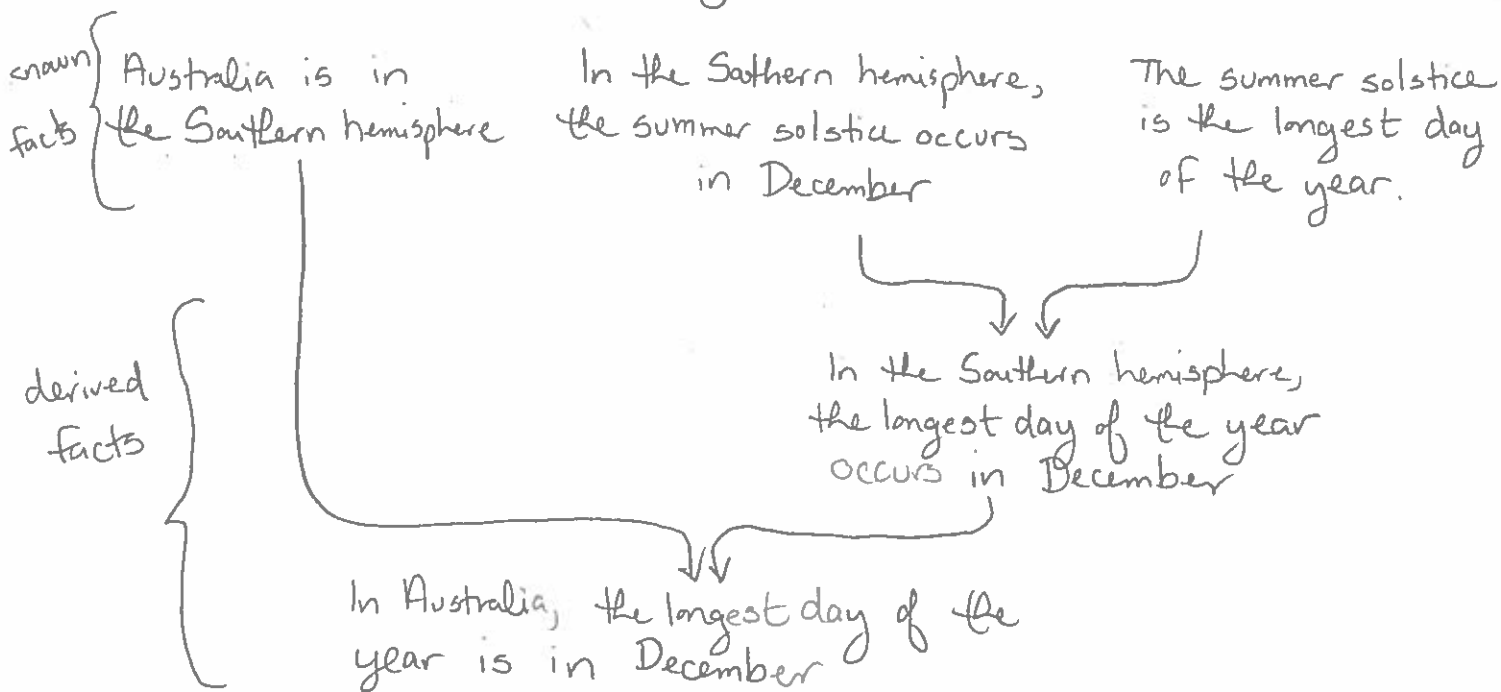
PROPOSITIONAL LOGIC

- ③ What just happened? Well, one (unlikely) theory of cognition is that we just have a big table that stores answers to all questions we might be asked:

<u>Place</u>	<u>Longest Day Occurs</u>
Canada	June
Egypt	June
China	June
Australia	December
Fiji	December

Of course, then for any city, town, geographic feature, etc. we'd need a separate table entry.

- ④ A more acceptable theory is that we do store some set of facts, but we can derive more facts through the application of rules of inference.



PROPOSITIONAL LOGIC

⑤ This reasoning seems very natural, but what are we actually doing? Can we create a mathematical model that replicates this process? This is what the study of logic attempts to do.

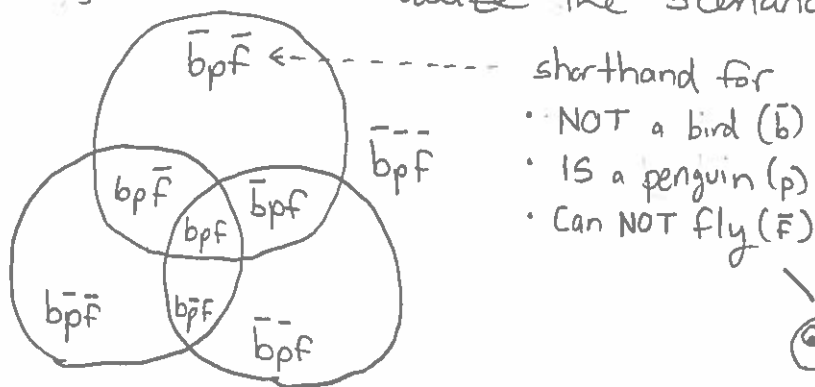
⑥ We'll begin by focusing on if-then statements. Consider the following simple example:

If you are a penguin, then you cannot fly.
Suppose you can fly.
Are you a penguin?

⑦ Fundamentally, we can boil this down into eight "possible worlds" that consider three criteria:

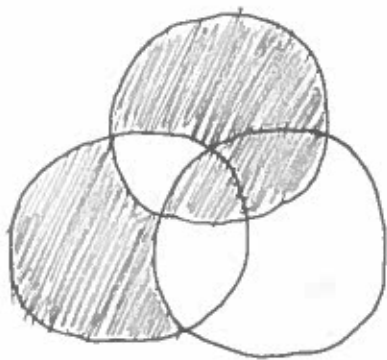
- whether you are a BIRD
- whether you are a PENGUIN
- whether you can FLY

Using a Venn diagram, we can visualize the scenarios:



PROPOSITIONAL LOGIC

- ⑧ We might consider some scenarios to be impossible. For instance, $\bar{b}p\bar{f}$ is impossible because a penguin is a bird by definition (so it can't be a penguin but not a bird). We could color in impossible scenarios:



- ⑨ But it's more standard to use a truth table:

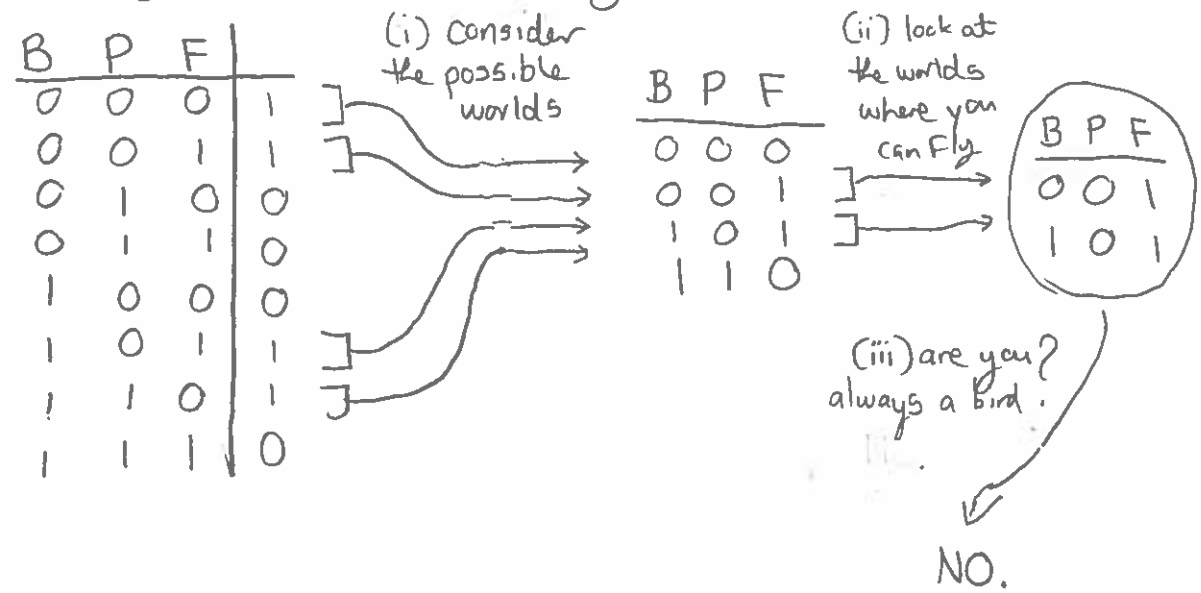
BIRD	PENGUIN	FLY	
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

↑ how many possible truth tables?

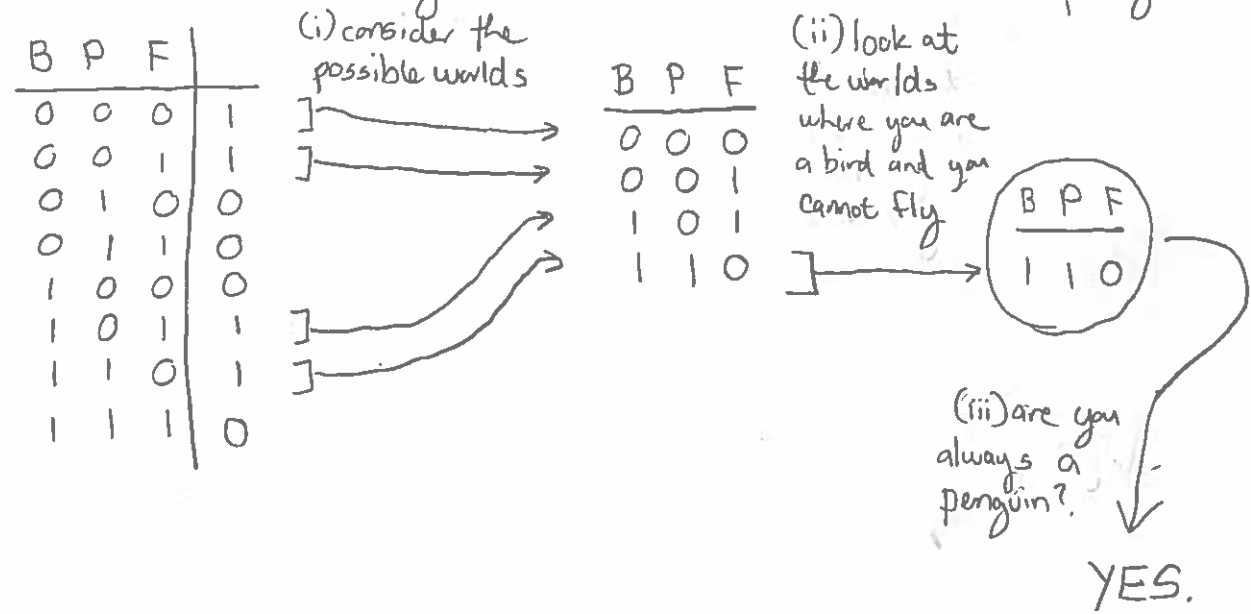
PROPOSITIONAL LOGIC

⑩ A truth table can answer many questions:

- if you can fly, does that mean you are a bird?



- if a bird cannot fly, does that mean it's a penguin?



PROPOSITIONAL LOGIC*

- ⑪ Unfortunately, it's impractical to represent knowledge with truth tables:

A_1	A_2	...	A_n	
0	0		0	} 2^n rows = too many to fit in memory for moderately large n
0	0		1	
0	0		0	
0	0		1	
⋮	⋮		⋮	
⋮	⋮		⋮	
1	1		0	
1	1		1	

- ⑫ Propositional logic allows us to represent knowledge more compactly and answer questions more efficiently. Essentially, it is a language for which every sentence encodes a truth table. We can view it as an encoding of the space of truth tables.

e.g. $B \wedge P$ encodes

B	P	F	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$(B \wedge \neg P) \Rightarrow F$ encodes

B	P	F	
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Language is a choice of what we prioritize as important (and atomic). There is a reason why the word "in" is short and the word "disestablishmentarianism" is long.

PROPOSITIONAL LOGIC

- ⑬ Let's define the language. First, we assume an alphabet Σ of symbols. For our animal scenario, this would be $\Sigma = \{B, P, F\}$. Often this alphabet Σ is referred to as a signature.

For a signature Σ , the propositional language $\mathcal{L}(\Sigma)$ is the minimal set of strings s.t.:

- for all $\sigma \in \Sigma$, $\sigma \in \mathcal{L}(\Sigma)$
- True $\in \mathcal{L}(\Sigma)$
- False $\in \mathcal{L}(\Sigma)$
- if $\alpha \in \mathcal{L}(\Sigma)$, then $\neg \alpha \in \mathcal{L}(\Sigma)$ and $(\neg \alpha) \in \mathcal{L}(\Sigma)$
- if $\alpha \in \mathcal{L}(\Sigma)$ and $\beta \in \mathcal{L}(\Sigma)$, then:
 - $(\alpha \wedge \beta) \in \mathcal{L}(\Sigma)$
 - $(\alpha \vee \beta) \in \mathcal{L}(\Sigma)$
 - $(\alpha \Rightarrow \beta) \in \mathcal{L}(\Sigma)$
 - $(\alpha \Leftrightarrow \beta) \in \mathcal{L}(\Sigma)$

- ⑭ Each string in $\mathcal{L}(\Sigma)$ is called a sentence of $\mathcal{L}(\Sigma)$ (or sometimes a formula). Examples include:

$$\begin{aligned} & B \\ & \neg P \\ & (\neg P \vee B) \\ & (\neg P \Rightarrow (B \vee F) \wedge (B \vee \neg P) \Rightarrow \neg F) \end{aligned}$$

which are
sentences

PROPOSITIONAL LOGIC

15) So far we have our language (untethered to meaning) and the meanings we'd like to encode:

B
¬P
(¬P ∨ B)
(¬P ⇒ ((B ∨ F) ∧ ((B ∨ ¬P) ⇒ ¬F)))

language $\mathcal{L}(\Sigma)$
"syntax"

B	P	F	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

B	P	F	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

truth tables $\tau(\Sigma)$
"semantics"

16) To make this into a logic, we need to define an interpretation function $I: \mathcal{L}(\Sigma) \mapsto \tau(\Sigma)$ that assigns a truth table to each sentence in our language.

First, define a model as a function from signature Σ to $\{0, 1\}$. Essentially a model is a row of a truth table (without the truth value). The space of models is written $M(\Sigma)$.

B	P	F	
0	0	0	← model
0	0	1	← model
0	1	0	← model
0	1	1	↑
1	0	0	↑
1	0	1	↑
1	1	0	↑
1	1	1	↑

PROPOSITIONAL LOGIC

⑦ Define the interpretation of a sentence in $\mathcal{L}(\Sigma)$ as:

- $I(\text{True}) = M(\Sigma)$
- $I(\text{False}) = \emptyset$
- for all $\sigma \in \Sigma$, $I(\sigma) = \{m \in M(\Sigma) \mid m(\sigma) = 1\}$
- for all $\alpha \in \mathcal{L}(\Sigma)$, $I(\neg \alpha) = M(\Sigma) - I(\alpha)$
- for all $\alpha, \beta \in \mathcal{L}(\Sigma)$:
 - $I(\alpha \wedge \beta) = I(\alpha) \cap I(\beta)$
 - $I(\alpha \vee \beta) = I(\alpha) \cup I(\beta)$
 - $I(\alpha \Rightarrow \beta) = I(\neg \alpha \vee \beta)$
 - $I(\alpha \Leftrightarrow \beta) = I(\alpha \Rightarrow \beta) \cap I(\beta \Rightarrow \alpha)$

⑧ For instance, let's interpret $((B \wedge \neg P) \Rightarrow F)$, i.e. "non-penguin birds can fly"

$$\begin{aligned}
 & I((B \wedge \neg P) \Rightarrow F) \\
 &= I((\neg(B \wedge \neg P) \vee F)) \\
 &= I(\neg(B \wedge \neg P)) \cup I(F) \\
 &= (M(\Sigma) - I((B \wedge \neg P))) \cup I(F) \\
 &= (M(\Sigma) - (I(B) \cap I(\neg P))) \cup I(F) \\
 &= \left(\begin{Bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{Bmatrix} - \left(\begin{Bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{Bmatrix} \cap \begin{Bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{Bmatrix} \right) \right) \cup \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{Bmatrix} \\
 &\quad \quad \quad \rightarrow = (M(\Sigma) - \begin{Bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{Bmatrix}) \cup \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{Bmatrix} \\
 &\quad \quad \quad = \begin{Bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{Bmatrix} \cup \begin{Bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{Bmatrix} \\
 &\quad \quad \quad = \begin{Bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{Bmatrix}
 \end{aligned}$$

PROPOSITIONAL LOGIC

- ① Now that we've associated an interpretation to each logical sentence, we can work in either the language (syntax) space or the truth table (semantic) space:

English
"every penguin
is a bird"

syntax space

$$P \Rightarrow B$$

semantic space

B	P	F	
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

"penguins cannot
fly"

$$P \Rightarrow \neg F$$

B	P	F	
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

can be much
more compact!

always the same
size

What if the signature is larger?

20) Because it is usually more concise to express knowledge in the language space, that's where we mainly want to live. But that also means we want a way to "reason" in that space, i.e. to answer questions like:

Given that:

- every bird can fly
- penguins cannot fly

Is a penguin not a bird?

Let's see how to represent such a question in both the syntax and the semantic space:

Syntax space

Thus we define entailment in the following way:

$$(B \Rightarrow F) \wedge (P \Rightarrow \neg F) \models P \Rightarrow \neg B$$

$$\text{iff } I((B \Rightarrow F) \wedge (P \Rightarrow \neg F)) \subseteq I(P \Rightarrow \neg B)$$

general form:

$$\alpha \models \beta \text{ iff } I(\alpha) \subseteq I(\beta)$$

Semantic space

B	P	F		B	P	F	
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	1	0	1	1	1	0

every bird can fly and penguins cannot \subseteq a penguin is not a bird

in every model where every bird can fly and penguins cannot, a penguin is not a bird.

② The fundamental computational question of propositional logic is, given two propositional sentences $\alpha, \beta \in \mathcal{L}(\Sigma)$, does it hold that $\alpha \models \beta$? We pursue this in the next lesson.