PROPOSITIONAL LOGIC: INFERENCE

① We want a ~~computationally efficient~~ way to determine whether $\alpha \models \beta$ for sentences $\alpha, \beta \in \mathcal{L}(\Sigma)$.

we'll settle for just a way

Revisiting our animal example, we could ask (for example):

$$\underbrace{(P \Rightarrow \neg F) \wedge (B \Rightarrow F)}_{\alpha} \overset{?}{\models} \underbrace{P \Rightarrow \neg B}_{\beta}$$

Without resorting to enumerating the truth table, how can we determine whether this entailment holds?

---

② Here is a helpful result:

Thm 1: For all $\alpha, \beta \in \mathcal{L}(\Sigma)$, $\alpha \models \beta$ iff $\mathbb{I}(\alpha \wedge \neg\beta) = \emptyset$

Proof: (only if) Suppose $\alpha \models \beta$ and assume there exists a model $m \in \mathbb{I}(\alpha \wedge \neg\beta)$. Thus $m \in \mathbb{I}(\alpha)$ and $m \in \mathbb{I}(\neg\beta)$, which means $m \notin \mathbb{I}(\beta)$.

Now $\alpha \models \beta$ means (by def'n) that $\mathbb{I}(\alpha) \subseteq \mathbb{I}(\beta)$, which means there is no model m' such that $m' \in \mathbb{I}(\alpha)$ and $m' \notin \mathbb{I}(\beta)$. Contradiction! So $\mathbb{I}(\alpha \wedge \neg\beta) = \emptyset$.

(if) Suppose $\mathbb{I}(\alpha \wedge \neg\beta) = \emptyset$. Thus $\mathbb{I}(\alpha) \cap \mathbb{I}(\neg\beta) = \emptyset$. Now consider $m \in \mathbb{M}(\Sigma)$. If $m \in \mathbb{I}(\alpha)$, then we know $m \notin \mathbb{I}(\neg\beta)$, thus $m \in \mathbb{I}(\beta)$ by definition of $\neg$. So $\mathbb{I}(\alpha) \subseteq \mathbb{I}(\beta)$, which means $\alpha \models \beta$.

Propositional Logic: Inference

③ Now we have a simple test for entailment:

does $\alpha \wedge \neg\beta$ have any models?

This means we can focus on finding an algorithm to determine whether an arbitrary sentence $\gamma \in \mathcal{L}(\Sigma)$ is <u>satisfiable</u> (defined as $\mathbb{I}(\gamma) \neq \emptyset$).

---

④ We'll begin by finding an algorithm for testing satisfiability that assumes the sentence γ has a particular form called Conjunctive Normal Form (CNF), which means that it is a conjunction (i.e. ANDing) of disjunctions (ORs), like:

$$(\neg P \vee F) \wedge (P \vee B) \wedge (\neg B \vee \neg F \vee \neg P)$$

---

⑤ More formally, define the <u>literals</u> of signature Σ to be

$$\text{LITERALS}(\Sigma) = \{\sigma \mid \sigma \in \Sigma\} \cup \{\neg\sigma \mid \sigma \in \Sigma\}$$

We'll use the notation $\bar{l}$ (for literal l) to mean:

$$\bar{l} = \begin{cases} \neg\sigma & \text{if } l = \sigma \\ \sigma & \text{if } l = \neg\sigma \end{cases}$$

The <u>clauses</u> of alphabet Σ are defined:

$$\text{CLAUSES}(\Sigma) = \{l_1 \vee \cdots \vee l_k \mid k \geq 1,\ l_i \in \text{LITERALS}\} \cup \{\text{False}\}$$

Propositional Logic: Inference

(6) A sentence is in CNF if it is a conjunction of clauses. CNF sentences have the benefit that they are somewhat intuitive to work with. Consider, for example, the sentence α:

$$(\neg P \vee F) \wedge (P \vee B) \wedge (\neg B \vee \neg F \vee P)$$

To show this is satisfiable, we need to find a satisfying model of α. In other words, we need $\mathbb{I}(\alpha)$ to be nonempty:

$$\mathbb{I}\big((\neg P \vee F) \wedge (P \vee B) \wedge (\neg B \vee \neg F \vee P) \big)$$

$$= \mathbb{I}(\neg P \vee F) \cap \mathbb{I}(P \vee B) \cap \mathbb{I}(\neg B \vee \neg F \vee P)$$

$$= \underbrace{\big(\mathbb{I}(\neg P) \cup \mathbb{I}(F)\big)}_{\substack{\text{models that} \\ \text{assign } P \to 0 \\ \text{or } F \to 1}} \cap \underbrace{\big(\mathbb{I}(P) \cup \mathbb{I}(B)\big)}_{\substack{\text{models that} \\ \text{assign } P \to 1 \\ \text{or } B \to 1}} \cap \underbrace{\big(\mathbb{I}(\neg B) \cup \mathbb{I}(\neg F) \cup \mathbb{I}(P)\big)}_{\substack{\text{models that} \\ \text{assign } B \to 0 \text{ or } F \to 0 \\ \text{or } P \to 1}}$$

So we can view each clause as a constraint on our model m:

$$\begin{array}{rcl} & & m \text{ needs to assign} \\ \neg P \vee F & \leftarrow & \underline{\underline{P \to 0}} \text{ or } F \to 1 \\ P \vee B & \leftarrow & P \to 1 \text{ or } \underline{\underline{B \to 1}} \\ \neg B \vee \neg F \vee P & \leftarrow & B \to 0 \text{ or } \underline{\underline{F \to 0}} \text{ or } P \to 1 \end{array}$$

From inspection we see that model $\{P \to 0, B \to 1, F \to 0\}$ satisfies α, so α is satisfiable.

Propositional Logic: Inference

---

(7) But how do we show a CNF sentence is unsatisfiable? Somehow we need to prove that there's no possible assignment that satisfies the constraints.

One strategy is to show that the sentence entails a sentence we already know is unsatisfiable. If we know β is unsatisfiable, and we show $\alpha \models \beta$, then:

$$\alpha \models \beta$$

implies $\mathbb{I}(\alpha) \subseteq \mathbb{I}(\beta)$ by def'n

implies $\mathbb{I}(\alpha) \subseteq \emptyset$ because β is unsat.

implies $\mathbb{I}(\alpha) = \emptyset$

So α is unsatisfiable.

---

(8) Do we know any sentences that are unsatisfiable? We definitely know one. The sentence

$$\text{False}$$

is unsatisfiable by definition, because $\mathbb{I}(\text{False}) = \emptyset$.

So we can show α is unsatisfiable by showing

$$\alpha \models \text{False}$$