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 $M(\alpha) \subseteq \mathbb{I}(\beta)$, 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:

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

(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:

		m needs to assign
$\neg P \vee F$	$\leftarrow$	$\underline{\underline{P \to 0}}$ or $F \to 1$
$P \vee B$	$\leftarrow$	$P \to 1$ or $\underline{\underline{B \to 1}}$
$\neg B \vee \neg F \vee P$	$\leftarrow$	$B \to 0$ or $\underline{\underline{F \to 0}}$ or $P \to 1$

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

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

Propositional Logic: Inference

(9) What we'd like are some <u>sound</u> rewrite rules so that if α can be rewritten as β according to those rules (written $\alpha \vdash \beta$), then $\alpha \models \beta$. Then we can show sentence α is unsatisfiable by showing:

$$\alpha \vdash \alpha_1 \vdash \ldots \vdash \alpha_k \vdash \text{False}$$

since that means:

$$\alpha \models \alpha_1 \models \ldots \models \alpha_k \models \text{False}$$

and thus:

$$\mathbb{I}(\alpha) \subseteq \mathbb{I}(\alpha_1) \subseteq \cdots \subseteq \mathbb{I}(\alpha_k) \subseteq \mathbb{I}(\text{False}) = \emptyset$$

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

---

(10) Consider the example CNF sentence α:

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

As before, we can view each clause as a constraint on any model $m \in \mathbb{I}(\alpha)$:

	m needs to assign
$\neg P \vee \neg F$	$P \to 0$ or $F \to 0$
$P \vee B$	$P \to 1$ or $B \to 1$
F	$F \to 1$
$\neg B$	$B \to 0$

Propositional Logic: Inference

(11) If we look at the first two clauses:

$\neg P \vee \neg F$	m needs to assign $P \to 0$ or $F \to 0$	P is either going to be 0 or 1 ...
$P \vee B$	$P \to 1$ or $B \to 1$	...if it's 0, then we need $B \to 1$ ...if it's 1, then we need $F \to 0$

We can infer that:

$$\alpha \models \neg F \vee B$$

Note that this also means:

$$\alpha \models (\neg F \vee B) \wedge \alpha$$

because for any sentence γ

$\alpha \models \gamma$ implies $\mathcal{I}(\alpha) \subseteq \mathcal{I}(\gamma)$

implies $\mathcal{I}(\alpha) \subseteq \mathcal{I}(\gamma) \cap \mathcal{I}(\alpha)$

implies $\alpha \models \gamma \wedge \alpha$

---

(12) We can continue this process:

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

$$\models (\neg P \vee \neg F) \wedge (P \vee B) \wedge F \wedge \neg B \wedge (\neg F \vee B)$$ [by "resolving" $\neg P \vee \neg F$ and $P \vee B$]

$$\models (\neg P \vee \neg F) \wedge (P \vee B) \wedge F \wedge \neg B \wedge (\neg F \vee B) \wedge B$$ [by "resolving" F and $\neg F \vee B$]

$$\models (\neg P \vee \neg F) \wedge (P \vee B) \wedge F \wedge \neg B \wedge (\neg F \vee B) \wedge B \wedge \text{False}$$ [by "resolving" $\neg B$ and B]

$$\models \text{False}$$

And prove that α is unsatisfiable.

(13) The general version of this rewrite (called the Resolution Rule) can be defined in the base case as:

$$l_1 \wedge \bar{l}_1 \wedge \beta \vdash \text{False}$$

and in the general case as

$$(l_1 \vee \cdots \vee l_i \vee \cdots \vee l_m) \wedge (l_1' \vee \cdots \vee l_j' \vee \cdots \vee l_n) \wedge \beta$$
$$\vdash (l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_m \vee l_1' \vee \cdots \vee l_{j-1}' \vee l_{j+1}' \vee \cdots \vee l_n) \wedge \beta$$

for literals $l_1, \ldots, l_m, l_1', \ldots, l_n' \in \text{LITERALS}(\Sigma)$ s.t. $l_i = \overline{l_j'}$ and arbitrary sentence $\beta \in \mathcal{L}(\Sigma)$.

---

(14) Soundness Thm: If $\alpha \vdash \gamma$, then $\alpha \models \gamma$.

Proof: (base) $\mathbb{I}(l_1 \wedge \bar{l}_i \wedge \beta) = \mathbb{I}(l_i) \cap \mathbb{I}(\bar{l}_1) \cap \mathbb{I}(\beta) = \emptyset = \mathbb{I}(\text{False})$

so $l_1 \wedge \bar{l}_1 \wedge \beta \models \text{False}$

(general) $\mathbb{I}\big((l_1 \vee \cdots \vee l_i \vee \cdots \vee l_m) \wedge (l_i' \vee \cdots \vee l_j' \vee \cdots \vee l_n) \wedge \beta\big)$

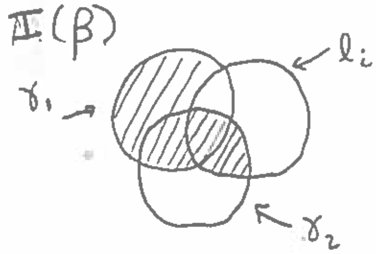
$= \mathbb{I}(l_1 \vee \cdots \vee l_i \vee \cdots \vee l_m) \cap \mathbb{I}(l_1' \vee \cdots \vee l_j' \vee \cdots \vee l_n) \cap \mathbb{I}(\beta)$

$= \big(\mathbb{I}(l_i) \cup \mathbb{I}(l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_m)\big)$
$\cap \big(\mathbb{I}(l_j') \cup \mathbb{I}(l_1' \vee \cdots \vee l_{j-1}' \vee l_{j+1}' \vee \cdots \vee l_n)\big)$
$\cap\, \mathbb{I}(\beta)$

$= \big(\mathbb{I}(l_i) \cup \mathbb{I}(\gamma_1)\big) \cap \big(\mathbb{I}(\bar{l}_i) \cup \mathbb{I}(\gamma_1')\big) \cap \mathbb{I}(\beta)$

$\subseteq \big(\mathbb{I}(\gamma_1) \cup \mathbb{I}(\gamma_1')\big) \cap \mathbb{I}(\beta)$

$= \mathbb{I}\big((\gamma_1 \vee \gamma_1') \wedge \beta\big)$



(15) So that means, if we can find a sequence of rewrites

$$\alpha \vdash \alpha_1 \vdash \ldots \vdash \alpha_k \vdash \text{False}$$

then

$$\alpha \vDash \text{False}$$

But is it also true that if we can't find a sequence of rewrites s.t. $\alpha \vdash^{*} \text{False}$, then we can conclude $\alpha \nvDash \text{False}$ (hence α is satisfiable)?

---

(16) Surprisingly, yes. To show this, first define the resolution closure of a set of clauses S as the smallest set $RC(S)$ such that:

- $c \in S \Rightarrow c \in RC(S)$
- $c_1, c_2 \in RC(S)$ and $c_1 \wedge c_2 \vdash c \Rightarrow c \in RC(S)$

In other words, $RC(S)$ is the set of clauses you can derive through repeated application of the Resolution Rule.

---

(17) For instance,

if $S = \{A \vee B, \neg B \vee \neg C\}$ then $RC(S) = \{A \vee B, \neg B \vee \neg C, A \vee \neg C\}$

if $S = \{A \vee B, B \vee \neg C\}$ then $RC(S) = S$.

PROPOSITIONAL LOGIC: INFERENCE

(18) Completeness Thm: If CNF sentence $c_1 \wedge \cdots \wedge c_N$ is unsatisfiable, then $\text{False} \in RC(\{c_1, \ldots, c_N\})$

Proof:

(i) We'll show the contrapositive, i.e. if $\text{False} \notin RC(S)$ for $S = \{c_1, \ldots, c_N\}$, then $c_1 \wedge \cdots \wedge c_N$ is satisfiable.

Let $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$

$\boxed{\text{Assume False} \notin RC(S).}$

(ii) $\boxed{\text{Let's construct a conjunction } l_1 \wedge \cdots \wedge l_M \text{ of literals s.t. } l_1 \wedge \cdots \wedge l_M \models c \text{ for every clause } c \in RC(S).}$

If we can do that, then

$$l_1 \wedge \cdots \wedge l_M \models c_1 \wedge \cdots \wedge c_N$$

so $I(l_1 \wedge \cdots \wedge l_M) \subseteq I(c_1 \wedge \cdots \wedge c_N)$, and since $I(l_1 \wedge \cdots \wedge l_M)$ is nonempty, therefore $I(c_1 \wedge \cdots \wedge c_N)$ is nonempty. $\therefore$ $c_1 \wedge \cdots \wedge c_N$ is satisfiable.

Examples:

(i) Let: $\Sigma = \{A, B, C\}$

$S = \{c_1, c_2\}$

where: c_1 is $\neg B \vee \neg C$

c_2 is $\neg A \vee C$

Thus:

$RC(S) = \{\neg B \vee \neg C, \neg A \vee C, \neg A \vee \neg B\}$

(ii) if we construct

$A \wedge \neg B \wedge C$

then:

$A \wedge \neg B \wedge C \models \neg B \vee \neg C$

$A \wedge \neg B \wedge C \models \neg A \vee C$

$A \wedge \neg B \wedge C \models \neg A \vee \neg B$

PROPOSITIONAL LOGIC: INFERENCE

(18) (cont.)

(iii) We'll initialize $l_0 = \text{True}$.

For $m = 1$ to M, set $l_m \in \{\sigma_m, \neg\sigma_m\}$
(if possible) such that:

$$l_0 \wedge \cdots \wedge l_m \not\models \neg c \text{ for each clause } c \in RC(S)$$

otherwise set $l_m = \sigma_m$.

(iv) For the sake of contradiction:

Assume m is the first iteration at which $l_0 \wedge \cdots \wedge l_m \models \neg c$ for some clause $c \in RC(S)$.

We know $m > 0$, because $\text{True} \not\models \neg c$ for all c except False, which is not in $RC(S)$ by the premise in (i)

(v) At iteration m, there exist clauses $c_1, c_2 \in RC(S)$ s.t.

$$l_0 \wedge \cdots \wedge l_{m-1} \wedge \sigma_m \models \neg c_1$$
$$l_0 \wedge \cdots \wedge l_{m-1} \wedge \neg\sigma_m \models \neg c_2$$

These clauses must have the forms:

$$c_1 : \; c_1' \vee \neg\sigma_m$$
$$c_2 : \; c_2' \vee \sigma_m$$

Otherwise:

$$l_0 \wedge \cdots \wedge l_{m-1} \models \neg c_1$$
$$l_0 \wedge \cdots \wedge l_{m-1} \models \neg c_2$$

which violates the assumption that m is the earliest iteration s.t. $l_0 \wedge \cdots \wedge l_m \models \neg c$ for $c \in RC(S)$

(iii) $l_0 = \text{True}$

$l_1 = A$

$l_2 = \neg B$ (otherwise $A \wedge B \models \neg(\neg A \vee \neg B)$)

$l_3 = C$ (otherwise $A \wedge \neg B \wedge \neg C \models \neg C_2$)

(v) say at iteration 3, we have $l_0 = \text{True}$

$l_1 = A$

$l_2 = B$

then:

$$\text{True} \wedge A \wedge B \wedge C \models \neg \underbrace{C_1}_{\neg B \vee \neg C}$$
$$\text{True} \wedge A \wedge B \wedge \neg C \models \neg \underbrace{C_2}_{\neg A \vee C}$$

18 (cont.)

(vi) Since $c_1, c_2 \in RC(S)$, thus $c_1' \vee c_2' \in RC(S)$, since c_1 and c_2 resolve to $c_1' \vee c_2'$.

(vi) $\neg B \vee \neg C$ resolves with $\neg A \vee C$ to obtain $\neg A \vee \neg B$, which is in $RC(S)$.

(vii) $\ell_0 \wedge \cdots \wedge \ell_{m-1} \wedge \sigma_m \models \neg c_1$

so $\ell_0 \wedge \cdots \wedge \ell_{m-1} \wedge \sigma_m \models \neg(c_1' \vee \neg\sigma_m)$

so $\ell_0 \wedge \cdots \wedge \ell_{m-1} \wedge \sigma_m \models \neg c_1' \wedge \sigma_m$ [De Morgan]

so $\ell_0 \wedge \cdots \wedge \ell_{m-1} \models \neg c_1'$

Analogously, $\ell_0 \wedge \cdots \wedge \ell_{m-1} \models \neg c_2'$

Therefore, $\ell_0 \wedge \cdots \wedge \ell_{m-1} \models \neg c_1' \wedge \neg c_2'$

$\equiv \neg(c_1' \vee c_2')$ [De Morgan]

Since $c_1' \vee c_2' \in RC(S)$, we obtain that $\ell_0 \wedge \cdots \wedge \ell_{m-1} \models \neg c$ for some $c \in RC(S)$, which means m was NOT the first iteration at which $\ell_0 \wedge \cdots \wedge \ell_m \models \neg c$.

— CONTRADICTION!

(viii) Thus, at <u>every</u> iteration m,

$$\ell_1 \wedge \cdots \wedge \ell_m \models c \quad \forall c \in RC(S)$$

$\boxed{\text{So } \ell_1 \wedge \cdots \wedge \ell_M \models c \quad \forall c \in RC(S)}$

QED.

(19) So what have we shown so far?

(i) we want to compute whether

$$\alpha \models \beta$$

for any $\alpha, \beta \in \mathcal{L}(\Sigma)$

this is equivalent to showing whether

$$\alpha \vee \neg \beta$$

is satisfiable

???

(ii) we want to compute whether

γ is satisfiable

for any $\gamma \in \mathcal{L}(\Sigma)$ such that γ is in CNF

this can be done using <u>resolution</u> in a finite number of steps

(20) We're missing one key piece to bring this home. Given a non-CNF sentence α, can we convert this into a CNF sentence β such that α is satisfiable iff β is satisfiable?

PROPOSITIONAL LOGIC: INFERENCE

(21) The answer is yes. Let's go through the recipe, using example (non-CNF) sentence $(Bird \Leftrightarrow (Penguin \vee Fly))$

(i) Replace $(\alpha \Leftrightarrow \beta)$ with $((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$:

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

(ii) Replace $(\alpha \Rightarrow \beta)$ with $(\neg\alpha \vee \beta)$:

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

(iii) Move $\neg$ "inwards" with three replacements: $\neg\neg\alpha$ with α, $\neg(\alpha \wedge \beta)$ with $(\neg\alpha \vee \neg\beta)$, and $\neg(\alpha \vee \beta)$ with $(\neg\alpha \wedge \neg\beta)$:

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

(iv) Distribute ands-over-ors and ors-over-ands with two replacements: $(\alpha \wedge (\beta \vee \gamma))$ with $((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$ and $(\alpha \vee (\beta \wedge \gamma))$ with $((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

$$(\underbrace{(\neg B \vee (P \vee F))}_{\text{Clause 1}} \wedge (\underbrace{(\neg P \vee B)}_{\text{clause 2}} \wedge \underbrace{(F \vee B)}_{\text{clause 3}}))$$

$$\neg B \vee P \vee F \;\wedge\; \neg P \vee B \;\wedge\; F \vee B$$

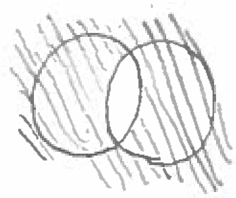
22) It is relatively straightforward to prove the correctness of this conversion by showing the correctness of each step. For instance, we can show that

$$I(\neg(\alpha \wedge \beta)) = I((\neg\alpha \vee \neg\beta))$$

as follows:

$$I(\neg(\alpha \wedge \beta)) = M(\Sigma) - I((\alpha \wedge \beta))$$

$$= M(\Sigma) - (I(\alpha) \cap I(\beta))$$



$$= (M(\Sigma) - I(\alpha)) \cup (M(\Sigma) - I(\beta))$$

$$= I(\neg\alpha) \cup I(\neg\beta)$$

$$= I((\neg\alpha \vee \neg\beta))$$