PROPOSITIONAL LOGIC

• Proposition - A declarative statement that is either true or false (but not both).

• Symbols in propositional logic:

  Proposition symbols
  TRUE, FALSE, p, q, r, ...
  Connectives
  ¬, ∧, ∨, →, ↔.

• Atom - a proposition symbol
  Literal - an atom p or its negation ¬p. An atom p is a positive literal and ¬p is a negative literal.

• Definition. (Well-Formed Formulas (WFF) = sentences).

  The well-formed formulas (or formulas for short), are defined inductively as follows:

  (1) An atom is a formula.
  (2) If G is a formula, then ¬G is a formula.
  (3) If G and H are formulas, then (G ∧ H), (G ∨ H), (G → H) and (G ↔ H) are formulas.
  (4) All formulas are generated by applying the above rules.

• A propositional theory Δ - a finite set of propositional formulas.

• Herbrand Base of Δ - the (finite) set of propositions (atoms) occurring in Δ, denoted as HB(Δ).

• Truth value of a formula φ in terms of the truth values of atoms occurring in φ.

  Let p and q be two propositions. The truth values of the formulas ¬p, p ∧ q, p ∨ q, p → q and p ↔ q in terms of the truth values of p and q are given by the following table:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬p</th>
<th>p ∧ q</th>
<th>p ∨ q</th>
<th>p → q</th>
<th>p ↔ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

• Interpretation - an assignment which assigns either T or F to each atom in HB(Δ). Equivalently, an interpretation I for a propositional theory Δ is a subset of HB(Δ) such that atoms in I are assigned T and those not in I are assigned F.

• Model of Δ - an interpretation M is a model of Δ if for each formula φ ∈ Δ, the truth value of φ under M is T. If the truth value of φ under I is T, then we say φ is satisfied by I. Otherwise, we say φ is falsified by I.

• Example 1. (propositional theory, interpretation and model).

  Consider the set of formulas Δ = {p ∧ q, r ∨ s, ¬a ∨ b}. Clearly Δ is a propositional theory. Consider the following interpretations I₁ = {p, r, b}, I₂ = {p, q} and I₃ = {p, q, s}. We can verify that I₁,
$I_2$ are not models of $\Delta$ and $I_3$ is a model from the following truth table:

<table>
<thead>
<tr>
<th>Inter.</th>
<th>a</th>
<th>b</th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
<th>$p \land q$</th>
<th>$r \lor s$</th>
<th>$\neg a \lor b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>$I_2$</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>$I_3$</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

- **Valid formula** - A formula $\phi$ is valid if it is true under all interpretations.
- **Unsatisfiable formula** - A formula $\phi$ is unsatisfiable if it is false under all interpretations, i.e., it has no models.
- **Satisfiable formula** - A formula $\phi$ is satisfiable if and only if $\phi$ has a model, i.e., if and only if is NOT unsatisfiable.

- **Equivalent formulas** - Two formulas $\phi$ and $\psi$ are equivalent if they have the same models. In other words, $\phi$ and $\psi$ are equivalent if they have the same truth value under every interpretation for $\phi$ and $\psi$.

For example, the formulas $p \rightarrow q$ and $\neg p \lor q$ are equivalent. The formulas $p \rightarrow q$ and $\neg q \rightarrow \neg p$ are also equivalent.

**Laws (Equivalent formulas) which can be used to perform formula transformation.**

1. $\phi \iff \psi = (\phi \rightarrow \psi) \land (\psi \rightarrow \phi)$
2. $\phi \rightarrow \psi = \neg \phi \lor \psi$
3a. $\phi \lor \psi = \psi \lor \phi$  
3b. $\phi \land \psi = \psi \land \phi$
4a. $\phi \lor (\psi \lor \gamma) = (\phi \lor \psi) \lor \gamma$
4b. $\phi \land (\psi \land \gamma) = (\phi \land \psi) \land \gamma$
5a. $\phi \lor (\psi \land \gamma) = (\phi \lor \psi) \land (\phi \lor \gamma)$
5b. $\phi \land (\psi \lor \gamma) = (\phi \land \psi) \lor (\phi \land \gamma)$
6a. $\phi \lor \text{false} = \phi$
6b. $\phi \land \text{true} = \phi$
7a. $\phi \lor \text{true} = \text{true}$  
7b. $\phi \land \text{false} = \text{false}$
8a. $\phi \lor \neg \phi = \text{true}$  
8b. $\phi \land \neg \phi = \text{false}$
9. $\neg (\neg \phi) = \phi$
10a. $\neg (\phi \lor \psi) = \neg \phi \land \neg \psi$
10b. $\neg (\phi \land \psi) = \neg \phi \lor \neg \psi$

- **Clause** - A disjunction of literals of the form $L_1 \lor L_2 \lor \ldots \lor L_m$.

**Theorem.** Each formula $\phi$ can be equivalently transformed to a formula $\phi'$ such that $\phi'$ is of the form $C_1 \land C_2 \land \ldots \land C_n$ where each $C_j$ is a clause.

Such a form $\phi'$ is called a conjunctive normal form of $\phi$. 

Conjunctive-Normal-Form Algorithm (outline).

Input:
A formula $\phi$.

Output:
A formula $\phi' = \phi$ such that $\phi'$ is in conjunctive normal form.

(1) Use laws $\phi \leftrightarrow \psi = (\phi \rightarrow \psi) \land (\psi \rightarrow \phi)$ and $\phi \rightarrow \psi = \neg \phi \lor \psi$ to eliminate connectives "$\leftrightarrow$" and "$\rightarrow$".

(2) Repeatedly apply the law $\neg(\neg \phi) = \phi$ to bring the negation sign "$\neg$" immediately before atom.

(3) Repeatedly apply distributive law $\phi \lor (\psi \land \gamma) = (\phi \lor \psi) \land (\phi \lor \gamma)$ and other laws to obtain a conjunctive normal form.

For example, the formula $\phi = (p \leftrightarrow q) \lor \neg(r \lor s)$ can be transformed to the formula $\phi' = (\neg p \lor q \lor \neg r) \land (\neg p \lor q \lor \neg s) \land (p \lor \neg q \lor \neg r) \land (p \lor \neg q \lor \neg s)$.

LOGICAL ENTAILMENT (also called LOGICAL CONSEQUENCE)

Definition (Logical Entailment). Let $\Delta = \{\phi_1, \phi_2, ..., \phi_n\}$ be a set of formulas and $\phi$ be a formula. We say $\phi_1 \land \phi_2 \land ... \land \phi_n$ logically entails $\phi$, if and only if any model of $\phi_1 \land \phi_2 \land ... \land \phi_n$ is a model of $\phi$. When $\phi_1 \land \phi_2 \land ... \land \phi_n$ logically entails $\phi$, we also say $\phi$ is a logical consequence of $\phi_1, \phi_2, ..., \phi_n$ (or $\phi$ logically follows from $\phi_1, \phi_2, ..., \phi_n$).

Example. Consider formulas $\{p \lor q \lor r, p \lor \neg r\}$. The formula $p \lor q$ is a logical consequence of $p \lor q \lor r$ and $p \lor \neg r$.

Theorem. A formula $\phi$ is a logical consequence of formulas $\phi_1, \phi_2, ..., \phi_n$ if and only if the formula $(\phi_1 \land \phi_2 \land ... \land \phi_n \rightarrow \phi)$ is valid.

Theorem. A formula $\phi$ is a logical consequence of formulas $\phi_1, \phi_2, ..., \phi_n$ if and only if the formula $\phi_1 \land \phi_2 \land ... \land \phi_n \land \neg \phi$ is unsatisfiable.

The above two theorems are very important because they tell us that the problem of showing $\phi$ being a logical consequence of a set of formulas can be reduced to the problem of showing a related formula to be unsatisfiable. The latter problem can be solved efficiently using resolution which we will describe shortly.
THE RESOLUTION PRINCIPLE

We assume from now on that each propositional formula \( \phi \) is represented in conjunctive normal form and thus we can equivalently represent \( \phi \) as \( \{ C_1, C_2, ..., C_n \} \) where each \( C_j \) is a clause and \( \phi = C_1 \land C_2 \land ... \land C_n \).

- Complementary literals - an atom \( p \) and its negation \( \neg p \) are called complementary literals.

**Definition. (resolvent).** Let \( C_1 \) and \( C_2 \) be two clauses such that \( C_1 = C'_1 \lor p \) and \( C_2 = C'_2 \lor \neg p \). The clause \( C = C'_1 \lor C'_2 \) is called the resolvent of \( C_1 \) and \( C_2 \), denoted as \( C = \text{res}(C_1, C_2) \). Here the atom \( p \) is called the resolving literal.

For example, let \( C_1 = a \lor \neg b \lor d \) and \( C_2 = q \lor \neg r \lor \neg d \). Then we have \( C = \text{res}(C_1, C_2) = a \lor \neg b \lor q \lor \neg r \).

**Theorem.** Let \( C = \text{res}(C_1, C_2) \) be the resolvent of clauses \( C_1 \) and \( C_2 \). Then \( C \) is a logical consequence of \( C_1 \) and \( C_2 \).

**Definition. (resolution derivation).** Let \( S \) be a set of clauses. A resolution derivation of a clause \( C \) from \( S \) is a sequence \( \sigma = (C_1, C_2, ..., C_k) \) of clauses such that

1. Each \( C_l \), either \( C_l \in S \) or \( C_l = \text{res}(C_i, C_j) \) for \( i, j < l \).
2. \( C_k = C \).

A resolution derivation of the empty clause \( \Box \) from \( S \) is called a refutation.

**Theorem.** If a clause \( C \) has a resolution derivation from a set \( S \) of clauses, then \( C \) is a logical consequence of \( S \).

**Theorem. (Soundness of the resolution principle).** Let \( S \) be a set of clauses. If there is a resolution derivation of the empty clause \( \Box \) from \( S \), then \( S \) is unsatisfiable.

**Theorem. (Completeness of the resolution principle).** Let \( S \) be a set of clauses. If \( S \) is unsatisfiable, then there is a resolution derivation of the empty clause \( \Box \) from \( S \).

From the above theorems and the theorems about logical consequence, we can easily see the equivalence of the following statements: (assume \( S = \{ C_1, C_2, ..., C_n \} \) is a set of clauses and \( G \) is a formula)

1. \( G \) is a logical consequence of \( S \);
2. the formula \( (C_1 \land C_2 \land ... \land C_n \land \neg G) \) is unsatisfiable;
3. the set of clauses \( S \cup \{ C_{n+1}, C_{n+2}, ..., C_{n+k} \} \) is unsatisfiable, where \( C_{n+1} \land C_{n+2} \land ... \land C_{n+k} = \neg G \);
4. there is a resolution derivation of the empty clause \( \Box \) from \( S \cup \{ C_{n+1}, C_{n+2}, ..., C_{n+k} \} \).
LOGICAL CONSEQUENCE ALGORITHM

Input:
A set S of clauses and a goal formula G.

Output:
a yes/no answer according to whether G is a logical consequence of S or not.

1. Negate the goal G to get \( \neg G \). Then transform \( \neg G \) to a set of clauses \( S' \).

2. If there is a resolution derivation of the empty clause \( \Box \) from \( S \cup S' \), then answer "yes" and terminate. Otherwise answer "no" and terminate.

Example. Let \( S = \{ p \lor q, \neg p \lor \neg q, \neg p \lor r, \neg q \lor s, p \lor \neg w, q \lor u \} \) and let \( G = (r \lor s) \land (u \lor \neg w) \).

We want to show that G is a logical consequence of S.

We first transform \( \neg G \) into clausal form:
\[
\neg G = (\neg(r \lor s) \lor (u \lor \neg w)) = ((\neg r \land \neg s) \lor (\neg u \land w)) = (\neg r \lor \neg u) \land (\neg r \lor w) \land (\neg s \lor \neg u) \land (\neg s \lor w).
\]

Thus \( S' = \{ \neg r \lor \neg u, \neg r \lor w, \neg s \lor \neg u, \neg s \lor w \} \).

We then search for a resolution derivation of the empty clause \( \Box \) from \( S \cup S' \). One such derivation is given below.

\[
\begin{align*}
\neg r & \lor w & \neg q & \lor s & p & \lor q & \neg p & \lor r & \neg p & \lor u & s & \lor \neg u & \neg p & \lor \neg q & p & \lor \neg w \\
q & \lor r & \neg q & \lor \neg w & q & \lor u & \neg s & \lor w & r & \lor s & \neg w & \lor u & s & \lor w & s & \lor \neg u \\
& \neg w & w & \neg u & \neg w
\end{align*}
\]

Exercises.

1. Let \( \Delta = \{(p \lor \neg r) \rightarrow q, (a \leftrightarrow b) \rightarrow c \} \). Convert \( \Delta \) into an equivalent set of clauses.

2. Let \( S = \{ p \lor q, p \lor \neg q, \neg p \lor q, \neg p \lor \neg q \} \). Indicate whether S is consistent or not. Support your conclusion by 2 ways: (i) Indicate whether S has a model; (ii) Indicate whether there is a resolution
derivation of the empty clause $\square$ from $S$.

3. Let $S = \{ a \lor \neg b \lor c, d \lor b, \neg a \lor d \}$. Show that the clause $c \lor d$ is a logical consequence of $S$ by resolution.

References