Propositional Logic, continued: Soundness, Completeness, & Special Forms

- ★ Soundness
- ★ Completeness
- ★ Conjuctive Normal Form
- ★ Horn Clauses

Soundness (1)

- Suppose $\Phi_1, \ldots, \Phi_n \vdash \Psi$ holds.
- Hence, there is a proof of Ψ having $\Phi_1, \ldots, \Phi_n$ as premises.
- We proceed by induction on the length of these proofs. We need to reformulate the soundness statement such that it is amenable to induction.

  $M(k)$: ***For all sequents*** $\Phi_1, \ldots, \Phi_n \vdash \Psi$ ***that have a proof of length*** k***, it is the case that*** $\Phi_1, \ldots, \Phi_n \models \Psi$

  We intend to use course-of-values induction on k.
- Technical problem:
  - Chopping a proof may nod lead to correct sub-proofs, since some boxes may still be open.
  - However, a chopped proof (a prefix of the sequence of formulas representing a proof) may form a correct proof if ***the assumptions of the open boxes are added to the premises***.

Soundness (2)

To solve our technical problem, we change the structure of the proof as in the following example. Consider the following sequent:

$$p \land q \to r \vdash p \to (q \to r)$$

1	$p \land q \to r$	premise
2	p	assumption
3	q	assumption
4	$p \land q$	$\land$i 2,3
5	r	$\to$e 1,4
6	$q \to r$	$\to$i 3–5
7	$p \to (q \to r)$	$\to$i 2–6

1	$\emptyset$	$p \land q \to r$	premise
2	$\{2\}$	p	assumption
3	$\{2,3\}$	q	assumption
4	$\{2,3\}$	$p \land q$	$\land$i 2,3
5	$\{2,3\}$	r	$\to$e 1,4
6	$\{2\}$	$q \to r$	$\to$i 3–5
7	$\emptyset$	$p \to (q \to r)$	$\to$i 2–6

Note: The set at the right of a formula in a proof line grows and shrinks as a stack, reflecting the way boxes are opened and closed.

Soundness (3)

Just for the purpose of proving soundness, we formally change the definition of the proof as follows.

Definition: A proof of the sequent $\Gamma \vdash \Psi$ is a sequence of pairs $[(d_1, \chi_1), \ldots, (d_k, \chi_k)]$ where:

(1) $d_1 = \emptyset$;

(2) each d_i is a subset of $\{1, \ldots, i\}$;

(3) for each i, χ_i is either

- a premise (i.e., $\chi_i \in \Gamma$), or
- an assumption (i.e. $\chi_i \in d_i$), or
- χ_i follows from previous lines by applying deduction rules;

(4) for each i, d_i is equal to

- d_{i-1} if no box was closed/opened at line i;
- $d_{i-1} \cup \{i\}$ if a box is opened on line i;
- $d_{i-1} \backslash \{\rho\}$ if a box with assumption at line *rho* was closed

The length of such a proof is k.

Soundness (4)

Our inductive statement now becomes:

> For any proof of length k $[(d_1,\chi_1),\ldots,(d_k,\chi_k)]$, and any assignment of truth values that makes the premises in Γ and the assumptions in d_k true, it is the case that χ_k evaluates to T.

When there is a chopping with no open boxes, this hypothesis precisely covers the semantic entailment.

Soundness (5)

We now proceed with the proof.

Base case $k = 1$: the proof has length 1, hence it is of the form

$$1\ \emptyset\ \chi\ \text{premise}$$

The statement is obviously true, any assignment of truth values that makes all the premises true, shall make this premise true as well.

Induction case $k > 1$: Suppose we have a proof

$$\begin{array}{llll} 1 & \emptyset & \chi_1 & premise \\ & & \vdots & \\ k & d_k & \chi_k & justification-\text{k} \end{array}$$

Soundness (6)

justification-k is a natural deduction rule, hence we proceed by case analysis.

$$\frac{\Phi \quad \Psi}{\Phi \wedge \Psi}\ \wedge\mathrm{i} \qquad \frac{\Phi \quad \Psi}{\Phi}\ \wedge\mathrm{e1} \qquad \frac{\Phi \quad \Psi}{\Psi}\ \wedge\mathrm{e2} \qquad \frac{\perp}{\Phi}\ \perp\mathrm{e}$$

$$\frac{\neg\neg\Phi}{\Phi}\ \neg\neg\mathrm{e}$$

$$\frac{\Phi}{\Phi \vee \Psi}\ \vee\mathrm{i1} \qquad \frac{\Psi}{\Phi \vee \Psi}\ \vee\mathrm{i2} \qquad \frac{\Phi \wedge \Psi \quad \boxed{\begin{array}{c}\Phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\Psi\\ \vdots\\ \chi\end{array}}}{\chi}\ \vee\mathrm{e}$$

$$\frac{\boxed{\begin{array}{c}\Phi\\ \vdots\\ \Psi\end{array}}}{\Phi \to \Psi}\ \to\mathrm{i} \qquad \frac{\Phi \quad \Phi \to \Psi}{\Psi}\ \to\mathrm{e}$$

$$\frac{\boxed{\begin{array}{c}\Phi\\ \vdots\\ \perp\end{array}}}{\neg\Phi}\ \to\mathrm{i} \qquad \frac{\neg\neg\Phi}{\Phi}\ \neg\neg\mathrm{e}$$

Soundness (7)

$\wedge$i: It must be the case that $\chi = \chi_1 \wedge \chi_2$, with χ_i appearing at line $k_i < k$, with $i \in \{1,2\}$. The formulas χ_1, χ_2 have shorter proofs, and therefore, using the induction hypothesis, they have the truth value T. Using the truth table for $\wedge$, we conclude that the truth value of χ is T.

$\vee$e: It must be the case that some formula $\chi_1 \vee \chi_2$ appears in in the proof, and that we have two boxes with assumptions χ_1 and χ_2 and conclusion χ. The proof of $\chi_1 \vee \chi_2$ is shorter hence, according to the induction hypothesis, it has a truth value of T. According to the truth table of $\vee$, either χ_1 or χ_2 have the truth value T. Assume it is χ_2 (the case when χ_1 has a truth value of T is similar). Then, the assumption χ_2 of the second box is true, and using the induction hypothesis, its conclusion χ has the truth value T.

The other cases are similar.

Completeness (1)

Theorem: Whenever $\Phi_1, \ldots, \Phi_n \models \Psi$ holds, there exists a natural deduction proof for the sequent $\Phi_1, \ldots, \Phi_n \vdash \Psi$.

The proof consists of three steps:

Step 1: $\models \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots))$

Step 2: $\vdash \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots))$

Step 3: $\Phi_1, \ldots, \Phi_n \vdash \Psi$

Definition: A formula of propositional logic is a ***tautology*** if it is true for all assignments of truth values to its propositional atoms, i.e. if $\models \Phi$.

Completeness (2)

Step 1: $\models \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots))$

This step is easy. Suppose $\Phi_1, \ldots, \Phi_n \models \Psi$ holds. The implication truth table shows that the only possiblity for $\models \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots))$ to fail is to have an assignment of truth values to its atoms that results in all $\Phi_1, \ldots, \Phi_n$ having the truth value T and Ψ having the truth value F; —but this is impossible, as it contradicts the hypothesis.

Step 3: $\Phi_1, \ldots, \Phi_n \vdash \Psi$

This step is also easy. Suppose $\vdash \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots))$ holds, i.e. has a natural deduction proof Π. Then we augment this proof by adding the premises $\Phi_1, \ldots, \Phi_n$ to the front, and then using the rule $\to$e at the end to produce Ψ. In other words, we produce the proof:

$$
\begin{array}{ll}
 & \Phi_1 \\
 & \vdots \\
 & \Phi_n \\
\Pi \left\{ \begin{array}{c} \vdots \\ \Phi_1 \to (\Phi_2 \to (\cdots(\Phi_n \to \Psi)\cdots)) \end{array} \right. & \\
 & \Psi \quad \to \mathrm{e}
\end{array}
$$

Completeness (3)

The core of the completeness proof is **Step 2**, which requires to show the following:

> If $\models \Phi$ holds, then $\vdash \Phi$ holds. In other words, if Phi is a tautology, then Φ is a theorem.

The idea of the proof is the following:

- Suppose $\models \Phi$ holds.
- If formula Ψ has n atoms $p_1, \ldots, p_n$, then Φ has truth value T for all the 2^n lines in its truth table.
- Then, we ***"encode"*** each line in the truth table of Φ as a sequent and assemble them into a proof of Φ using the disjunction rules.

Completeness (4)

The first part of the proof is based on the following lemma.

Lemma: Let Φ be a formula containing the propositional atoms $p_1, \ldots, p_n$, and l a line of Φ's truth table. Let $\hat{p}_i$ be p_i if the entry in line l of p_i is T, otherwise $\hat{p}_i$ is $\neg p_i$. Then,

$\hat{p}_1, \ldots, \hat{p}_n \vdash \Phi$ is provable if the entry for Φ in line l is T;

$\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\Phi$ is provable if the entry for Φ in line l is F.

Completeness (5)

The proof of the lemma is by course of values induction on the height of the syntax tree of Ψ.

Base case: If Φ is an atom (i.e. a formula of height 1), then we have to show that $p \vdash p$ and $\neg p \vdash \neg p$ hold. This is immediate.

Completeness (6)

Induction case: The height of Φ is greater than 1. Then, we have the following cases.

- Φ is of the form $\neg\Phi_1$.
  - If Φ evaluates to T, then Φ_1 evaluates to F; Φ_1 has the same atoms as Φ, but a lower height, hence by induction hypothesis $\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\Phi_1$; finally $\neg\Phi_1$ is Φ, hence we are done.
  - If Φ evaluates to F, then Φ_1 evaluates to T; by induction hypothesis we get $\hat{p}_1, \ldots, \hat{p}_n \vdash \Phi_1$, which can be extended to $\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\neg\Phi_1$ using the $\neg\neg$i rule; but $\neg\neg\Phi_1$ is $\neg\Phi_1$, hence we are done.
- Φ is of the form $\Phi_1 \circ \Phi_2$, where $\circ \in \{\wedge, \vee \rightarrow\}$. Let $q_1, \ldots, q_l$ be the atoms of Φ_1 and $r_1, \ldots, r_k$ the atoms of Φ_2, where $\{q_1, \ldots, q_l\} \cup \{r_1, \ldots, r_k\} = \{p_1, \ldots, p_n\}$. We are left with proving that

$$\hat{q}_1, \ldots, \hat{q}_l \vdash \Psi_1 \text{ and } \hat{r}_1, \ldots, \hat{r}_k \vdash \Psi_2$$
$$\text{implies } \hat{p}_1, \ldots, \hat{p}_n \vdash \Psi_1 \wedge \Psi_2$$

  for appropriate formulas Ψ_1 and Ψ_2.

Completeness (7)

We show the proof for $\circ = \wedge$, that is, we consider the case when $\Phi = \Phi_1 \wedge \Phi_2$.

- If both Φ_1 and Φ_2 evaluate to T, then by the induction hypothesis $\hat{q}_1, \ldots, \hat{q}_l \vdash \Phi_1$ and $\hat{r}_1, \ldots, \hat{r}_k \vdash \Phi_2$, hence $\hat{p}_1, \ldots, \hat{p}_n \vdash \Phi_1 \wedge \Phi_2$, and we are done.

- If Φ_1 evaluates to F and Φ_2 evaluates to T, then we have $\hat{q}_1, \ldots, \hat{q}_l \vdash \neg\Phi_1$ and $\hat{r}_1, \ldots, \hat{r}_k \vdash \Phi_2$, hence $\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\Phi_1 \wedge \Phi_2$ We are left with proving

$$\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\Phi_1 \wedge \Phi_2 \text{ implies } \hat{p}_1, \ldots, \hat{p}_n \vdash \neg(\Phi_1 \wedge \Phi_2)$$

  (left as an exercise)

- The other two cases are similar, requiring the following proofs:

$$\begin{aligned} \Phi_1 \wedge \neg\Phi_2 \quad &\vdash \quad \neg(\Phi_1 \wedge \Phi_2) \\ \neg\Phi_1 \wedge \Phi_2 \quad &\vdash \quad \neg(\Phi_1 \wedge \Phi_2) \end{aligned}$$

Completeness (8)

If Φ is of the form $\Phi_1 \vee \Phi_2$, we can reduce the proof to the search for the following proofs.

$$\begin{array}{rcl} \Phi_1 \wedge \Phi_2 & \vdash & \Phi_1 \vee \Phi_2 \\ \Phi_1 \wedge \neg\Phi_2 & \vdash & \Phi_1 \vee \Phi_2 \\ \neg\Phi_1 \wedge \Phi_2 & \vdash & \Phi_1 \vee \Phi_2 \\ \neg\Phi_1 \wedge \neg\Phi_2 & \vdash & \neg(\Phi_1 \vee \Phi_2) \end{array}$$

If Φ is of the form $\Phi_1 \rightarrow \Phi_2$, we can reduce the proof to the search for the following proofs.

$$\begin{array}{rcl} \Phi_1 \wedge \Phi_2 & \vdash & \Phi_1 \rightarrow \Phi_2 \\ \Phi_1 \wedge \neg\Phi_2 & \vdash & \neg(\Phi_1 \rightarrow \Phi_2) \\ \neg\Phi_1 \wedge \Phi_2 & \vdash & \Phi_1 \rightarrow \Phi_2 \\ \neg\Phi_1 \wedge \neg\Phi_2 & \vdash & \Phi_1 \rightarrow \Phi_2 \end{array}$$

Completeness (9)

The last piece of the puzzle is to assemble these proofs of the form

$$\hat{p}_1, \ldots, \hat{p}_n \vdash \Phi$$

each representing a line in the truth table, into a proof of $\vdash \Phi$, without premises.

We use the disjunction rules to generate the lines of the truth table, then we appropriately insert the above proofs.

We exemplify this procedure for the case of two atoms, for the tautology $\vdash p \wedge q \to p$.

Completeness (10)

Assembling the proof for the tautology $\vdash p \wedge q \rightarrow p$.

1	$p \vee \neg p$								LEM
2	p			ass	$\neg p$				ass
3	$q \vee \neg q$			LEM	$q \vee \neg$ q				LEM
4	q	ass	$\neg q$	ass	q	ass	$\neg q$	ass	
5	$\vdots$		$\vdots$		$\vdots$		$\vdots$		
6									
7	$p \wedge q \rightarrow p$		$p \wedge q \rightarrow p$		$p \wedge q \rightarrow p$		$p \wedge q \rightarrow p$		
8	$p \wedge q \rightarrow p$			$\vee$e	$p \wedge q \rightarrow p$				$\vee$e
9	$p \wedge q \rightarrow p$								$\vee$e

Definitions:

- Let Φ and Ψ be propositional logic formulas. They are ***semantically equivalent*** iff $\Phi \models \Psi$ and $\Psi \models \Phi$. We denote this by $\Phi \equiv \Psi$.
- Φ is ***valid*** iff $\models \Phi$.

Remarks:

- Two formulas Φ and Ψ are semantically equivalent iff $\models (\Phi \rightarrow \Psi) \wedge (\Psi \rightarrow \Phi)$.
- Because of soundness and completeness of propositional logic, semantic equivalence is identical with provable equivalence $\vdash (\Phi \rightarrow \Psi) \wedge (\Psi \rightarrow \Phi)$. (This is a fortunate case, most logics are not complete).
- Our aim is to transform formulas into equivalent ones for which checking validity is easier.

Conjunctive Normal Form

Definitions:

- A ***literal*** is either an atom p, or the negation of an atom $\neg p$.
- A formula Φ is in ***conjunctive normal form (CNF)*** if it is of the form $\Psi_1 \wedge \Psi_2 \wedge \cdots \wedge \Psi_n$, for some $n \geq 1$, where each Ψ_i is a disjunction of literals, for all $i \in \{1, \ldots, n\}$.

Note: Sometimes we include the case $n = 0$, in which case, by convention, the term is $\top$.

Examples of CNFs:

$$(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$$
$$(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$$

Not in CNF:

$$(\neg(q \vee p) \vee r) \wedge (\neg p \vee r) \wedge q$$

Validity of a Disjunction of Literals

Lemma: A disjunction of literals $L_1 \vee L_2 \vee \cdots \vee L_n$ is valid iff there exist i, j, with $1 \leq i, j \leq n$, such that L_i is $\neg L_j$.

Proof:

- If there exist i, j such that L_i is $\neg L_j$, then clearly $L_1 \vee L_2 \vee \cdots \vee L_n$ evaluates to T for all assignments.
- For the converse, if no literal has a matching negation, then:
  - For each positive literal we assign F to the corresponding atom.
  - For each negative literal we assign T to the corresponding atom.
  - This assignment falsifies the disjunction, which is impossiible. (Example: for $\neg q \vee p \vee r$, take p and r to be true, and q to be false and q to be true.)
  - Hence, there exist i, j such that L_i is $\neg L_j$.

Definition: A formula Ψ is satisfiable if there exists an assignment of truth values to its propositional atoms such that Φ is true.

Proposition: A propositional logic formula Φ is satisfiable iff $\neg\Phi$ is not valid.

Proof:

- If Φ is satisfiable, then there exists a valuation (assignment of truth values to its atoms) which makes Φ true. For this valuation $\neg\Phi$ has the truth value F, hence $\neg\Phi$ cannot be valid.
- Conversely, if $\neg\Phi$ is not valid, then there exists a valuation for which $\neg\Phi$ has the truth value F. This valuation makes Φ have the truth value T, hence Φ is satisfiable.

This is a simple, but very useful result.

Useful Identities (Boolean Algebra)

$\wedge$ and $\vee$ are ***idempotent***

$$\Phi \wedge \Phi \equiv \Phi$$
$$\Phi \vee \Phi \equiv \Phi$$

$\wedge$ and $\vee$ are ***commutative***

$$\Phi \wedge \Psi \equiv \Psi \wedge \Phi$$
$$\Phi \vee \Psi \equiv \Psi \vee \Phi$$

$\wedge$ and $\vee$ are ***associative***

$$\Phi \wedge (\Psi \wedge \eta) \equiv (\Phi \wedge \Psi) \wedge \eta$$
$$\Phi \vee (\Psi \vee \eta) \equiv (\Phi \vee \Psi) \vee \eta$$

$\wedge$ and $\vee$ are ***absorbative***

$$\Phi \wedge (\Phi \vee \eta) \equiv \Phi$$
$$\Phi \vee (\Phi \wedge \eta) \equiv \Phi$$

$\wedge$ and $\vee$ are ***distributive***

$$\Phi \wedge (\Psi \vee \eta) \equiv (\Phi \wedge \Psi) \vee (\Phi \wedge \eta)$$
$$\Phi \vee (\Psi \wedge \eta) \equiv (\Phi \vee \Psi) \wedge (\Phi \vee \eta)$$

Rules for T and F

$$F \wedge \Phi \equiv F \qquad \Phi \wedge \neg\Phi \equiv F$$
$$T \vee \Phi \equiv T \qquad \Phi \vee \neg\Phi \equiv T$$

The de Morgan rules

$$\neg(\Phi \wedge \Psi) \equiv \neg\Phi \vee \neg\Psi$$
$$\neg(\Phi \vee \Psi) \equiv \neg\Phi \wedge \neg\Psi$$

Double negation rules

$$\neg\neg\Phi \equiv \Phi$$

A Procedure to Compute CNFs

We present an algorithm to compute a CNF formula equivalent to a given arbitrary formula Φ. The algorithm is deterministic and computes a unique CNF for any formula.
The algorithm is described as:

$$\texttt{CNF}(\texttt{NNF}(\texttt{IMPL\_FREE}(\Phi)))$$

for a given formula Φ. The **`CNF`**, **`NNF`**, and **`IMPL_FREE`** functions shall be discussed shortly.

function `IMPL_FREE`(Φ) :
/* precondition: Φ is an arbitrary formula */
/* postcondition: returns an implication free formula equivalent to Φ */
begin function
  case
    Φ is a literal: **return** Φ
    Φ is $\neg\Phi_1$: **return** $\neg(\mathtt{IMPL\_FREE}(\Phi_1))$
    Φ is $\Phi_1 \wedge \Phi_2$: **return** $\mathtt{IMPL\_FREE}(\Phi_1) \wedge \mathtt{IMPL\_FREE}(\Phi_2)$
    Φ is $\Phi_1 \vee \Phi_2$: **return** $\mathtt{IMPL\_FREE}(\Phi_1) \vee \mathtt{IMPL\_FREE}(\Phi_2)$
    Φ is $\Phi_1 \to \Phi_2$: **return** $\neg\mathtt{IMPL\_FREE}(\Phi_1) \vee \mathtt{IMPL\_FREE}(\Phi_2)$
  end case
end function

IMPL_FREE Example

Let $\Phi = \neg p \wedge q \to p \wedge (r \to q)$.

$$
\begin{aligned}
&\texttt{IMPL\_FREE}(\Phi) \\
&= \neg\texttt{IMPL\_FREE}(\neg p \wedge q) \vee \texttt{IMPL\_FREE}(p \wedge (r \to q)) \\
&= \neg((\texttt{IMPL\_FREE}(\neg p)) \wedge \texttt{IMPL\_FREE}(q)) \vee \texttt{IMPL\_FREE}(p \wedge (r \to q)) \\
&= \neg((\neg p) \wedge \texttt{IMPL\_FREE}(q)) \vee \texttt{IMPL\_FREE}(p \wedge (r \to q)) \\
&= \neg((\neg p) \wedge q) \vee \texttt{IMPL\_FREE}(p \wedge (r \to q)) \\
&= \neg((\neg p) \wedge q) \vee (p \wedge (\neg\texttt{IMPL\_FREE}(r) \vee \texttt{IMPL\_FREE}(q))) \\
&= \neg((\neg p) \wedge q) \vee (p \wedge (\neg r \vee \texttt{IMPL\_FREE}(q))) \\
&= \neg((\neg p) \wedge q) \vee (p \wedge (\neg r \vee q))
\end{aligned}
$$

function **NNF**(Φ) :
/* precondition: Φ is implication free */
/* postcondition: returns an NNF formula equivalent to Φ */
begin function
  case
    Φ is a literal: **return** Φ
    Φ is $\neg\neg\Phi_1$: **return** **NNF**(Φ_1)
    Φ is $\Phi_1 \wedge \Phi_2$: **return** **NNF**$(\Phi_1) \wedge$ **NNF**(Φ_2)
    Φ is $\Phi_1 \vee \Phi_2$: **return** **NNF**$(\Phi_1) \vee$ **NNF**(Φ_2)
    Φ is $\neg(\Phi_1 \wedge \Phi_2)$: **return** **NNF**$(\neg\Phi_1 \vee \neg\Phi_2)$
    Φ is $\neg(\Phi_1 \vee \Phi_2)$: **return** **NNF**$(\neg\Phi_1 \wedge \neg\Phi_2)$
  end case
end function

NNF Example

Let $\Phi = \neg((\neg p) \wedge q) \vee (p \wedge (\neg r \vee q))$.

$$\begin{aligned}
&\mathsf{NNF}(\Phi)\\
&= \mathsf{NNF}(\neg((\neg p) \wedge q)) \vee \mathsf{NNF}(p \wedge (\neg r \vee q))\\
&= \mathsf{NNF}(\neg(\neg p) \vee \neg q)) \vee \mathsf{NNF}(p \wedge (\neg r \vee q))\\
&= (\mathsf{NNF}(\neg\neg p)) \vee (\mathsf{NNF}(\neg q)) \vee \mathsf{NNF}(p \wedge (\neg r \vee q))\\
&= p \vee (\mathsf{NNF}(\neg q)) \vee \mathsf{NNF}(p \wedge (\neg r \vee q))\\
&= p \vee \neg q \vee \mathsf{NNF}(p \wedge (\neg r \vee q))\\
&= p \vee \neg q \vee (\mathsf{NNF}(p) \wedge \mathsf{NNF}(\neg r \vee q))\\
&= p \vee \neg q \vee (p \wedge \mathsf{NNF}(\neg r \vee q))\\
&= p \vee \neg q \vee (p \wedge (\mathsf{NNF}(\neg r) \vee \mathsf{NNF}(q)))\\
&= p \vee \neg q \vee (p \wedge (\neg r \vee \mathsf{NNF}(q)))\\
&= p \vee \neg q \vee (p \wedge (\neg r \vee q))
\end{aligned}$$

function `CNF`(Φ) :
/* precondition: Φ is implication and in NNF */
/* postcondition: returns an CNF formula equivalent to Φ */
begin function
  case
    Φ is a literal: **return** Φ
    Φ is $\Phi_1 \wedge \Phi_2$: **return** $\texttt{CNF}(\Phi_1) \wedge \texttt{CNF}(\Phi_2)$
    Φ is $\Phi_1 \vee \Phi_2$: **return** $\texttt{DISTR}(\texttt{CNF}(\Phi_1), \texttt{CNF}(\Phi_2))$
  end case
end function

DISTR

function **`DISTR`**(Φ_1,Φ_2) :
/* precondition: Φ_1, Φ_2 are in CNF */
/* postcondition: returns an CNF formula equivalent to $\Phi_1 \vee \Phi_2$ */
begin function
    case
        Φ_1 is $\Phi_{11} \wedge \Phi_{12}$: **return** **`DISTR`**$(\Phi_{11}, \Phi_2) \wedge$ **`DISTR`**(Φ_{12}, Φ_2)
        Φ_2 is $\Phi_{21} \wedge \Phi_{22}$: **return** **`DISTR`**$(\Phi_1, \Phi_{21}) \wedge$ **`DISTR`**(Φ_1, Φ_{22})
        otherwise: **return** $\Phi_1 \vee \Phi_2$
    end case
end function

CNF Example

Let $\Phi = p \vee \neg q \vee (p \wedge (\neg r \vee q))$.

$$
\begin{aligned}
& \texttt{CNF}(\Phi) \\
&= \texttt{CNF}(p \vee \neg q \vee (p \wedge (\neg r \vee q))) \\
&= \texttt{DISTR}(\texttt{CNF}(p \vee \neg q), \texttt{CNF}(p \wedge (\neg r \vee q))) \\
&= \texttt{DISTR}(p \vee \neg q, \texttt{CNF}(p \wedge (\neg r \vee q))) \\
&= \texttt{DISTR}(p \vee \neg q, p \wedge (\neg r \vee q)) \\
&= \texttt{DISTR}(p \vee \neg q, p) \wedge \texttt{DISTR}(p \vee \neg q, \neg r \vee q) \\
&= (p \vee \neg q \vee p) \wedge (p \vee \neg q \vee \neg r \vee q)
\end{aligned}
$$

Definitions:

- A ***Horn clause*** is a formula of the form $p_1 \wedge p_2 \wedge \cdots \wedge p_k \rightarrow q$, where $k \geq 1$, and $p_1, p_2, \ldots, p_k, q$ are atoms, $\perp$, or $\top$.

- A ***Horn formula*** is a conjunction of Horn clauses, i.e. a formula Φ of the form $\Psi_1 \wedge \Psi_2 \wedge \cdots \wedge \Psi_n$, ($n \geq 1$), such that each Ψ_i is a Horn clause, $i \in \{1, \ldots, n\}$.

Horn clauses have an efficient procedure to decide their satisfiability, and are the basis for logic programming.

Horn Clauses Examples

Examples (yes)

$$(p \wedge q \wedge s \to p) \wedge (q \wedge r \to p) \wedge (p \wedge s \to s)$$
$$(p \wedge q \wedge s \to \bot) \wedge (q \wedge r \to p) \wedge (\top \to s)$$
$$(p_1 \wedge p_3 \wedge p_5 \to p_{13}) \wedge (\top \to p_5) \wedge (p_5 \wedge p_{11} \to \bot)$$

Examples (no)

$$(p \wedge q \wedge s \to \neg p) \wedge (q \wedge r \to p) \wedge (p \wedge s \to s)$$
$$(p \wedge q \wedge s \to \bot) \wedge (\neg q \wedge r \to p) \wedge (\top \to s)$$
$$(p_1 \wedge p_3 \wedge p_5 \to p_{13} \wedge p_{27}) \wedge (\top \to p_5) \wedge (p_5 \wedge p_{11} \to \bot)$$

function **HORN**(Φ) :
/* precondition: Φ is a Horn formula*/
/* postcondition: decides the satisfiability of Φ*/
begin function
    if Φ contains a clause $\top \rightarrow \bot$ **then return** unsatisfiable
        else mark all atoms p where $\top \rightarrow p$ is a clause of Φ
    while there is a Horn clause $p_1 \wedge \cdots p_{k_i} \rightarrow q_i$ of Φ
            such that all p_j are marked, but q_i isn't **do**
        if $q_i \equiv \bot$ **then return** 'unsatisfiable'
            else mark q_i for all Horn clauses of Φ
    end while
    return 'satisfiable'
end function

Theorem: The **HORN** algorithm is correct: it always terminates and its answer is 'satisfiable' iff the given Horn formula is satisfiable.