Mathematical Logics
PL - Reasoning via Truth Tables*

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli

Lecture index

1. Truth tables – summary
2. Decision problems
3. Deciding satisfiability (SAT)
    a. CNF
    b. The DPLL SAT decision procedure
4. Satisfiability in practice: MiniSat

Conjunctive Normal form

Definition

- A literal is either a propositional variable or the negation of a propositional variable.

$$p, \quad \neg q$$

- A clause is a disjunction of literals.

$$(a \vee \neg b \vee c)$$

- A formula is in conjunctive normal form, if it is a conjunction of clauses.

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

Conjunctive Normal form

Conjunctive Normal form

A formula in conjunctive normal form has the following shape:

$$(l_{11} \vee \ldots \vee l_{1n_1}) \wedge \ldots \wedge (l_{m1} \vee \ldots \vee l_{mn_m})$$

equivalently written as

$$\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{n_j} l_{ij} \right)$$

where l_{ij} is the j-th literal of the i-th clause composing φ

Example

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

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

Properties of ∧ and ∨

Commutativity of	$\wedge$**:**	$\varphi \wedge \psi$	$\equiv$	$\psi \wedge \varphi$
Commutativity of	$\vee$**:**	$\varphi \vee \psi$	$\equiv$	$\psi \vee \varphi$
Absorption of	$\wedge$**:**	$\varphi \wedge \varphi$	$\equiv$	φ
Absorption of	$\vee$**:**	$\varphi \vee \varphi$	$\equiv$	φ

Properties of clauses

Order of literals does not matter

If a clause C is obtained by reordering the literals of a clause C' then the two clauses are equivalent.

$$(p \lor q \lor r \lor \neg r) \equiv (\neg r \lor q \lor p \lor r)$$

Multiple literals can be merged

If a clause contains more than one occurrence of the same literal then it is equivalent to the close obtained by deleting all but one of these occurrences:

$$(p \lor q \lor r \lor q \lor \neg r) \equiv (p \lor q \lor r \lor \neg r)$$

Clauses as set of literals

From these properties we can represent a clause as a set of literals, by leaving disjunction implicit and by ignoring replication and order of literals

$(p \lor q \lor r \lor \neg r)$ is represented by the set $\{p, q, r, \neg r\}$

Properties of formulas in CNF

Order of claused does not matter

If a CNF formula φ is obtained by reordering the clauses of a CNF formula φ' then φ and φ' are equivalent

$$(a \vee b) \wedge (c \vee \neg b) \wedge (\neg b) \equiv (c \vee \neg b) \wedge (\neg b) \wedge (a \vee b)$$

Multiple clauses can be merged

If a CNF formula contains more than one occurrence of the same clause then it is equivalent to the formula obtained by deleting all but one of the duplicated occurrences:

$$(a \vee b) \wedge (c \vee \neg b) \wedge (a \vee b) \equiv (a \vee b) \wedge (c \vee \neg b)$$

A CNF formula can be seen as a set of clauses

From the props. of clauses and of CNF formulas, we can represent a CNF formula as a set of sets of literals.

$(a \vee b) \wedge (c \vee \neg b) \wedge (\neg b)$ is represented by: $\{\{a, b\}, \{c, \neg b\}, \{\neg b\}\}$

Properties of formulas in CNF (cont'd)

Proposition

Existence *Every formula can be reduced into CNF*

Equivalence $\models CNF(\varphi) \equiv \varphi$

Reduction in CNF

Definition (the CNF function)

The function *CNF*, which transforms a propositional formula in its CNF is recursively defined as follows:

$$
\begin{aligned}
CNF(p) &= p \quad \text{if } p \in PROP \\
CNF(\neg p) &= \neg p \text{ if } p \in PROP \\
CNF(\phi \to \psi) &= CNF(\neg\phi) \otimes CNF(\psi) \\
CNF(\phi \wedge \psi) &= CNF(\phi) \wedge CNF(\psi) \\
CNF(\phi \vee \psi) &= CNF(\phi) \otimes CNF(\psi) \\
CNF(\phi \equiv \psi) &= CNF(\phi \to \psi) \wedge CNF(\psi \to \phi) \\
CNF(\neg\neg\phi) &= CNF(\phi) \\
CNF(\neg(\phi \to \psi)) &= CNF(\phi) \wedge CNF(\neg\psi) \\
CNF(\neg(\phi \wedge \psi)) &= CNF(\neg\phi) \otimes CNF(\neg\psi) \\
CNF(\neg(\phi \vee \psi)) &= CNF(\neg\phi) \wedge CNF(\neg\psi) \\
CNF(\neg(\phi \equiv \psi)) &= CNF(\phi \wedge \neg\psi) \otimes CNF(\psi \wedge \neg\phi)
\end{aligned}
$$

where $(C_1 \wedge \cdots \wedge C_n) \otimes (D_1 \wedge \cdots \wedge D_m)$ is defined as

$$(C_1 \vee D_1) \wedge \cdots \wedge (C_1 \vee D_m) \wedge \cdots \wedge (C_n \vee D_1) \wedge \cdots \wedge (C_n \vee D_m)$$

CNF transformation example 1

$$CNF((a \wedge b) \vee \neg (c \rightarrow d)) \quad =$$

$$CNF(a \wedge b) \otimes CNF(\neg (c \rightarrow d)) \quad =$$

$$(CNF(a) \wedge CNF(b)) \otimes (CNF(c) \wedge CNF(\neg d)) \quad =$$

$$(a \wedge b) \otimes (c \wedge \neg d) \quad =$$

$$(a \vee c) \wedge (a \vee \neg d) \wedge (b \vee c) \wedge (b \vee \neg d)$$

CNF transformation example 2

$CNF((\neg((p \rightarrow q) \wedge (p \vee q \rightarrow r)) \rightarrow (p \rightarrow r)))$ =

$CNF(\neg\neg((p \rightarrow q) \wedge (p \vee q \rightarrow r))) \otimes CNF(p \rightarrow r)$ =

$CNF((p \rightarrow q) \wedge (p \vee q \rightarrow r)) \otimes (CNF(\neg p) \otimes CNF(r))$ =

$(CNF(p \rightarrow q) \wedge CNF(p \vee q \rightarrow r)) \otimes (\neg p \vee r)$ =

$((CNF(\neg p) \otimes CNF(q)) \wedge (CNF(\neg (p \vee q)) \otimes CNF(r))) \otimes (\neg p \vee r)$ =

$((\neg p \otimes q) \wedge ((CNF(\neg p) \wedge CNF(\neg q)) \otimes CNF(r))) \otimes (\neg p \vee r)$ =

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

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

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

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

CNF transformation

Cost of CNF

CNF is a normal form, it is simpler since it uses only 3 connective (e.g., $\wedge$, $\vee$ and $\neg$) in a very specific form. Checking satisfiability/validity of a formula in CNF is easier. But there is a price: . . .

Example (Exponential explosion)

Compute the CNF of

$$p1 \equiv (p2 \equiv (p3 \equiv (p4 \equiv (p5 \equiv p6)))).$$

The first step yields:

$$CNF(p1 \rightarrow (p2 \equiv (p3 \equiv (p4 \equiv (p5 \equiv p6)))))\wedge$$
$$CNF((p2 \equiv (p3 \equiv (p4 \equiv (p5 \equiv p6)))) \rightarrow p1)$$

NOTE: If we continue, the formula will grow exponentially. There are techniques for partially limiting this problem

Mathematical Logics
PL - Reasoning via Truth Tables*

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli