Normal Forms

Definition (CNF, DNF)

A formula is in *conjunctive normal form (CNF)* or *clause normal form* if it is a conjunction of disjunctions of literals, or in other words, a conjunction of clauses.

A formula is in *disjunctive normal form (DNF)*, if it is a disjunction of conjunctions of literals.

Checking the validity of CNF formulas or the unsatisfiability of DNF formulas is easy:

(i) a formula in CNF is valid, if and only if each of its disjunctions contains a pair of complementary literals P and $\neg P$,

(ii) conversely, a formula in DNF is unsatisfiable, if and only if each of its conjunctions contains a pair of complementary literals P and $\neg P$

Basic CNF Transformation

ElimEquiv	$\chi[(\phi \leftrightarrow \psi)]_p \Rightarrow_{\text{BCNF}} \chi[(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)]_p$
ElimImp	$\chi[(\phi \rightarrow \psi)]_p \Rightarrow_{\text{BCNF}} \chi[(\neg\phi \vee \psi)]_p$
PushNeg1	$\chi[\neg(\phi \vee \psi)]_p \Rightarrow_{\text{BCNF}} \chi[(\neg\phi \wedge \neg\psi)]_p$
PushNeg2	$\chi[\neg(\phi \wedge \psi)]_p \Rightarrow_{\text{BCNF}} \chi[(\neg\phi \vee \neg\psi)]_p$
PushNeg3	$\chi[\neg\neg\phi]_p \Rightarrow_{\text{BCNF}} \chi[\phi]_p$
PushDisj	$\chi[(\phi_1 \wedge \phi_2) \vee \psi]_p \Rightarrow_{\text{BCNF}} \chi[(\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)]_p$
ElimTB1	$\chi[(\phi \wedge \top)]_p \Rightarrow_{\text{BCNF}} \chi[\phi]_p$
ElimTB2	$\chi[(\phi \wedge \bot)]_p \Rightarrow_{\text{BCNF}} \chi[\bot]_p$
ElimTB3	$\chi[(\phi \vee \top)]_p \Rightarrow_{\text{BCNF}} \chi[\top]_p$
ElimTB4	$\chi[(\phi \vee \bot)]_p \Rightarrow_{\text{BCNF}} \chi[\phi]_p$
ElimTB5	$\chi[\neg\bot]_p \Rightarrow_{\text{BCNF}} \chi[\top]_p$
ElimTB6	$\chi[\neg\top]_p \Rightarrow_{\text{BCNF}} \chi[\bot]_p$

Basic CNF Algorithm

---

1 **Algorithm: 2** bcnf(ϕ)

Input : A propositional formula ϕ.
Output: A propositional formula ψ equivalent to ϕ in CNF.

2 **whilerule** *(***ElimEquiv**(ϕ)*)* **do** ;
3 **whilerule** *(***ElimImp**(ϕ)*)* **do** ;
4 **whilerule** *(***ElimTB1**(ϕ),. . .,**ElimTB6**(ϕ)*)* **do** ;
5 **whilerule** *(***PushNeg1**(ϕ),. . .,**PushNeg3**(ϕ)*)* **do** ;
6 **whilerule** *(***PushDisj**(ϕ)*)* **do** ;
7 **return** ϕ;

Advanced CNF Algorithm

For the formula

$$P_1 \leftrightarrow (P_2 \leftrightarrow (P_3 \leftrightarrow (\ldots (P_{n-1} \leftrightarrow P_n) \ldots)))$$

the basic CNF algorithm generates a CNF with 2^{n-1} clauses.

2.5.4 Proposition (Renaming Variables)

Let P be a propositional variable not occurring in $\psi[\phi]_p$.

1. If $\mathrm{pol}(\psi, p) = 1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (P \to \phi)$ is satisfiable.
2. If $\mathrm{pol}(\psi, p) = -1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (\phi \to P)$ is satisfiable.
3. If $\mathrm{pol}(\psi, p) = 0$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (P \leftrightarrow \phi)$ is satisfiable.

Renaming

SimpleRenaming $\phi \Rightarrow_{\text{SimpRen}} \phi[P_1]_{p_1}[P_2]_{p_2} \ldots [P_n]_{p_n} \wedge \text{def}(\phi, p_1, P_1) \wedge \ldots \wedge \text{def}(\phi[P_1]_{p_1}[P_2]_{p_2} \ldots [P_{n-1}]_{p_{n-1}}, p_n, P_n)$
provided $\{p_1, \ldots, p_n\} \subset \text{pos}(\phi)$ and for all $i, i+j$ either $p_i \parallel p_{i+j}$ or $p_i > p_{i+j}$ and the P_i are different and new to ϕ

Simple choice: choose $\{p_1, \ldots, p_n\}$ to be all non-literal and non-negation positions of ϕ.