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

Basic CNF Algorithm

Algorithm: 2 bcnf(ϕ)

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

whilerule *(***ElimEquiv**(ϕ)*)* **do** ;
whilerule *(***ElimImp**(ϕ)*)* **do** ;
whilerule *(***ElimTB1**(ϕ),...,**ElimTB6**(ϕ)*)* **do** ;
whilerule *(***PushNeg1**(ϕ),...,**PushNeg3**(ϕ)*)* **do** ;
whilerule *(***PushDisj**(ϕ)*)* **do** ;
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 $\text{pol}(\psi, p) = 1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (P \rightarrow \phi)$ is satisfiable.
2. If $\text{pol}(\psi, p) = -1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (\phi \rightarrow P)$ is satisfiable.
3. If $\text{pol}(\psi, p) = 0$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (P \leftrightarrow \phi)$ is satisfiable.

Renaming

SimpleRenaming $\phi \Rightarrow_{\text{SimpRen}} \phi[P_1]_{p_1}[P_2]_{p_2} \dots [P_n]_{p_n} \wedge \text{def}(\phi, p_1, P_1) \wedge \dots \wedge \text{def}(\phi[P_1]_{p_1}[P_2]_{p_2} \dots [P_{n-1}]_{p_{n-1}}, p_n, P_n)$
provided $\{p_1, \dots, p_n\} \subset \text{pos}(\phi)$ 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 $\{p_1, \dots, p_n\}$ to be all non-literal and non-negation positions of ϕ.

Renaming Definition

$$\mathrm{def}(\psi, p, P) := \begin{cases} (P \rightarrow \psi|_p) & \text{if } \mathrm{pol}(\psi, p) = 1 \\ (\psi|_p \rightarrow P) & \text{if } \mathrm{pol}(\psi, p) = -1 \\ (P \leftrightarrow \psi|_p) & \text{if } \mathrm{pol}(\psi, p) = 0 \end{cases}$$

Obvious Positions

A smaller set of positions from ϕ, called *obvious positions*, is still preventing the explosion and given by the rules:

(i) p is an obvious position if $\phi|_p$ is an equivalence and there is a position $q < p$ such that $\phi|_q$ is either an equivalence or disjunctive in ϕ or

(ii) pq is an obvious position if $\phi|_{pq}$ is a conjunctive formula in ϕ, $\phi|_p$ is a disjunctive formula in ϕ and for all positions r with $p < r < pq$ the formula $\phi|_r$ is not a conjunctive formula.

A formula $\phi|_p$ is conjunctive in ϕ if $\phi|_p$ is a conjunction and $\operatorname{pol}(\phi, p) \in \{0, 1\}$ or $\phi|_p$ is a disjunction or implication and $\operatorname{pol}(\phi, p) \in \{0, -1\}$.

Analogously, a formula $\phi|_p$ is disjunctive in ϕ if $\phi|_p$ is a disjunction or implication and $\operatorname{pol}(\phi, p) \in \{0, 1\}$ or $\phi|_p$ is a conjunction and $\operatorname{pol}(\phi, p) \in \{0, -1\}$.

Polarity Dependent Equivalence Elimination

ElimEquiv1 $\chi[(\phi \leftrightarrow \psi)]_p \Rightarrow_{\mathrm{ACNF}} \chi[(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)]_p$
provided $\mathrm{pol}(\chi, p) \in \{0, 1\}$

ElimEquiv2 $\chi[(\phi \leftrightarrow \psi)]_p \Rightarrow_{\mathrm{ACNF}} \chi[(\phi \wedge \psi) \vee (\neg\phi \wedge \neg\psi)]_p$
provided $\mathrm{pol}(\chi, p) = -1$

Extra $\top, \bot$ Elimination Rules

ElimTB7	$\chi[\phi \rightarrow \bot]_p$	$\Rightarrow_{\mathrm{ACNF}}$	$\chi[\neg\phi]_p$
ElimTB8	$\chi[\bot \rightarrow \phi]_p$	$\Rightarrow_{\mathrm{ACNF}}$	$\chi[\top]_p$
ElimTB9	$\chi[\phi \rightarrow \top]_p$	$\Rightarrow_{\mathrm{ACNF}}$	$\chi[\top]_p$
ElimTB10	$\chi[\top \rightarrow \phi]_p$	$\Rightarrow_{\mathrm{ACNF}}$	$\chi[\phi]_p$
ElimTB11	$\chi[\phi \leftrightarrow \bot]_p$	$\Rightarrow_{\mathrm{ACNF}}$	$\chi[\neg\phi]_p$
ElimTB12	$\chi[\phi \leftrightarrow \top]_p$	$\Rightarrow_{\mathrm{ACNF}}$	$\chi[\phi]_p$

where the two rules ElimTB11, ElimTB12 for equivalences are applied with respect to commutativity of $\leftrightarrow$.

Advanced CNF Algorithm

Algorithm: 3 acnf(ϕ)

Input : A formula ϕ.

Output: A formula ψ in CNF satisfiability preserving to ϕ.

```
whilerule (ElimTB1(φ),...,ElimTB12(φ)) do ;
SimpleRenaming(φ) on obvious positions;
whilerule (ElimEquiv1(φ),ElimEquiv2(φ)) do ;
whilerule (ElimImp(φ)) do ;
whilerule (PushNeg1(φ),...,PushNeg3(φ)) do ;
whilerule (PushDisj(φ)) do ;
return φ;
```