Substitution Theorem

Proposition 2.7:
Let ϕ_1 and ϕ_2 be equivalent formulas, and $\psi[\phi_1]_p$ be a formula in which ϕ_1 occurs as a subformula at position p.

Then $\psi[\phi_1]_p$ is equivalent to $\psi[\phi_2]_p$.

Equivalences

Proposition 2.8:

The following equivalences are valid for all formulas ϕ, ψ, χ:

$(\phi \wedge \phi) \leftrightarrow \phi$	Idempotency $\wedge$
$(\phi \vee \phi) \leftrightarrow \phi$	Idempotency $\vee$
$(\phi \wedge \psi) \leftrightarrow (\psi \wedge \phi)$	Commutativity $\wedge$
$(\phi \vee \psi) \leftrightarrow (\psi \vee \phi)$	Commutativity $\vee$
$(\phi \wedge (\psi \wedge \chi)) \leftrightarrow ((\phi \wedge \psi) \wedge \chi)$	Associativity $\wedge$
$(\phi \vee (\psi \vee \chi)) \leftrightarrow ((\phi \vee \psi) \vee \chi)$	Associativity $\vee$
$(\phi \wedge (\psi \vee \chi)) \leftrightarrow (\phi \wedge \psi) \vee (\phi \wedge \chi)$	Distributivity $\wedge\vee$
$(\phi \vee (\psi \wedge \chi)) \leftrightarrow (\phi \vee \psi) \wedge (\phi \vee \chi)$	Distributivity $\vee\wedge$

Equivalences

$(\phi \wedge \phi) \leftrightarrow \phi$	Absorption $\wedge$
$(\phi \vee \phi) \leftrightarrow \phi$	Absorption $\vee$
$(\phi \wedge (\phi \vee \psi)) \leftrightarrow \phi$	Absorption $\wedge\vee$
$(\phi \vee (\phi \wedge \psi)) \leftrightarrow \phi$	Absorption $\vee\wedge$
$(\phi \wedge \neg\phi) \leftrightarrow \bot$	Introduction $\bot$
$(\phi \vee \neg\phi) \leftrightarrow \top$	Introduction $\top$

Equivalences

$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$	De Morgan $\neg\vee$
$\neg(\phi \wedge \psi) \leftrightarrow (\neg\phi \vee \neg\psi)$	De Morgan $\neg\wedge$
$\neg\top \leftrightarrow \bot$	Propagate $\neg\top$
$\neg\bot \leftrightarrow \top$	Propagate $\neg\bot$

Equivalences

$(\phi \wedge \top) \leftrightarrow \phi$	Absorption $\top\wedge$
$(\phi \vee \bot) \leftrightarrow \phi$	Absorption $\bot\vee$
$(\phi \rightarrow \bot) \leftrightarrow \neg\phi$	Eliminate $\bot \rightarrow$
$(\phi \leftrightarrow \bot) \leftrightarrow \neg\phi$	Eliminate $\bot \leftrightarrow$
$(\phi \leftrightarrow \top) \leftrightarrow \phi$	Eliminate $\top \leftrightarrow$
$(\phi \vee \top) \leftrightarrow \top$	Propagate $\top$
$(\phi \wedge \bot) \leftrightarrow \bot$	Propagate $\bot$

Equivalences

$(\phi \to \psi) \leftrightarrow (\neg\phi \vee \psi)$	Eliminate $\to$
$(\phi \leftrightarrow \psi) \leftrightarrow (\phi \to \psi) \wedge (\psi \to \phi)$	Eliminate1 $\leftrightarrow$
$(\phi \leftrightarrow \psi) \leftrightarrow (\phi \wedge \psi) \vee (\neg\phi \wedge \neg\psi)$	Eliminate2 $\leftrightarrow$

For simplification purposes the equivalences are typically applied as left to right rules.

2.4 Normal Forms

We define conjunctions of formulas as follows:

$$\bigwedge_{i=1}^{0} \phi_i = \top.$$

$$\bigwedge_{i=1}^{1} \phi_i = \phi_1.$$

$$\bigwedge_{i=1}^{n+1} \phi_i = \bigwedge_{i=1}^{n} \phi_i \wedge \phi_{n+1}.$$

and analogously disjunctions:

$$\bigvee_{i=1}^{0} \phi_i = \bot.$$

$$\bigvee_{i=1}^{1} \phi_i = \phi_1.$$

$$\bigvee_{i=1}^{n+1} \phi_i = \bigvee_{i=1}^{n} \phi_i \vee \phi_{n+1}.$$

Literals and Clauses

A literal is either a propositional variable P or a negated propositional variable $\neg P$.

A clause is a (possibly empty) disjunction of literals.

CNF and DNF

A formula is in conjunctive normal form (CNF, 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.

Warning: definitions in the literature differ:

are complementary literals permitted?
are duplicated literals permitted?
are empty disjunctions/conjunctions permitted?

CNF and DNF

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

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

Conversely, a formula in DNF is unsatisfiable, if and only if each of its conjunctions contains a pair of complementary literals P and $\neg P$.

On the other hand, checking the unsatisfiability of CNF formulas or the validity of DNF formulas is known to be coNP-complete.

Conversion to CNF/DNF

Proposition 2.9:
For every formula there is an equivalent formula in CNF (and also an equivalent formula in DNF).

Proof:
We consider the case of CNF and propose a naive algorithm.

Apply the following rules as long as possible (modulo associativity and commutativity of $\wedge$ and $\vee$):

Step 1: Eliminate equivalences:

$$(\phi \leftrightarrow \psi) \ \Rightarrow_{\mathsf{ECNF}} \ (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$$

Conversion to CNF/DNF

Step 2: Eliminate implications:

$$(\phi \rightarrow \psi) \ \Rightarrow_{\mathsf{ECNF}} \ (\neg\phi \vee \psi)$$

Step 3: Push negations downward:

$$\neg(\phi \vee \psi) \ \Rightarrow_{\mathsf{ECNF}} \ (\neg\phi \wedge \neg\psi)$$

$$\neg(\phi \wedge \psi) \ \Rightarrow_{\mathsf{ECNF}} \ (\neg\phi \vee \neg\psi)$$

Step 4: Eliminate multiple negations:

$$\neg\neg\phi \ \Rightarrow_{\mathsf{ECNF}} \ \phi$$

Conversion to CNF/DNF

Step 5: Push disjunctions downward:

$$(\phi \wedge \psi) \vee \chi \;\Rightarrow_{\mathsf{ECNF}}\; (\phi \vee \chi) \wedge (\psi \vee \chi)$$

Step 6: Eliminate $\top$ and $\bot$:

$$\begin{aligned}
(\phi \wedge \top) &\;\Rightarrow_{\mathsf{ECNF}}\; \phi \\
(\phi \wedge \bot) &\;\Rightarrow_{\mathsf{ECNF}}\; \bot \\
(\phi \vee \top) &\;\Rightarrow_{\mathsf{ECNF}}\; \top \\
(\phi \vee \bot) &\;\Rightarrow_{\mathsf{ECNF}}\; \phi \\
\neg\bot &\;\Rightarrow_{\mathsf{ECNF}}\; \top \\
\neg\top &\;\Rightarrow_{\mathsf{ECNF}}\; \bot
\end{aligned}$$

Conversion to CNF/DNF

Proving termination is easy for steps 2, 4, and 6; steps 1, 3, and 5 are a bit more complicated.

The resulting formula is equivalent to the original one and in CNF.

The conversion of a formula to DNF works in the same way, except that conjunctions have to be pushed downward in step 5.

□

Complexity

Conversion to CNF (or DNF) may produce a formula whose size is exponential in the size of the original one.

Satisfiability-preserving Transformations

The goal

“find a formula ψ in CNF such that $\phi \models\mid \psi$”

is unpractical.

But if we relax the requirement to

“find a formula ψ in CNF such that $\phi \models \bot \Leftrightarrow \psi \models \bot$”

we can get an efficient transformation.

Satisfiability-preserving Transformations

Idea: A formula $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \land (P \leftrightarrow \phi)$ is satisfiable where P is a new propositional variable that does not occur in ψ and works as an abbreviation for ϕ.

We can use this rule recursively for all subformulas in the original formula (this introduces a linear number of new propositional variables).

Conversion of the resulting formula to CNF increases the size only by an additional factor (each formula $P \leftrightarrow \phi$ gives rise to at most one application of the distributivity law).

Optimized Transformations

A further improvement is possible by taking the polarity of the subformula into account.

For example if $\psi[\phi_1 \leftrightarrow \phi_2]_p$ and $\mathrm{pol}(\psi, p) = -1$ then for CNF transformation do $\psi[(\phi_1 \wedge \phi_2) \vee (\neg\phi_1 \wedge \neg\phi_2)]_p$.

Optimized Transformations

Proposition 2.10:

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

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.

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.

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.

Proof:

Exercise. $\square$

Optimized Transformations

The number of eventually generated clauses is a good indicator for useful CNF transformations:

ψ	$\nu(\psi)$	$\bar{\nu}(\psi)$
$\phi_1 \wedge \phi_2$	$\nu(\phi_1) + \nu(\phi_2)$	$\bar{\nu}(\phi_1)\bar{\nu}(\phi_2)$
$\phi_1 \vee \phi_2$	$\nu(\phi_1)\nu(\phi_2)$	$\bar{\nu}(\phi_1) + \bar{\nu}(\phi_2)$
$\phi_1 \rightarrow \phi_2$	$\bar{\nu}(\phi_1)\nu(\phi_2)$	$\nu(\phi_1) + \bar{\nu}(\phi_2)$
$\phi_1 \leftrightarrow \phi_2$	$\nu(\phi_1)\bar{\nu}(\phi_2) + \bar{\nu}(\phi_1)\nu(\phi_2)$	$\nu(\phi_1)\nu(\phi_2) + \bar{\nu}(\phi_1)\bar{\nu}(\phi_2)$
$\neg\phi_1$	$\bar{\nu}(\phi_1)$	$\nu(\phi_1)$
$P, \top, \bot$	1	1

Optimized CNF

Step 1: Exhaustively apply modulo C of $\leftrightarrow$, AC of $\wedge$, $\vee$:

$$
\begin{aligned}
(\phi \wedge \top) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \vee \bot) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \leftrightarrow \bot) &\Rightarrow_{\mathsf{OCNF}} \neg\phi \\
(\phi \leftrightarrow \top) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \vee \top) &\Rightarrow_{\mathsf{OCNF}} \top \\
(\phi \wedge \bot) &\Rightarrow_{\mathsf{OCNF}} \bot
\end{aligned}
$$

Optimized CNF

$$
\begin{aligned}
(\phi \wedge \phi) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \vee \phi) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \wedge (\phi \vee \psi)) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \vee (\phi \wedge \psi)) &\Rightarrow_{\mathsf{OCNF}} \phi \\
(\phi \wedge \neg\phi) &\Rightarrow_{\mathsf{OCNF}} \bot \\
(\phi \vee \neg\phi) &\Rightarrow_{\mathsf{OCNF}} \top \\
\neg\top &\Rightarrow_{\mathsf{OCNF}} \bot \\
\neg\bot &\Rightarrow_{\mathsf{OCNF}} \top
\end{aligned}
$$

Optimized CNF

$$(\phi \to \bot) \Rightarrow_{\mathsf{OCNF}} \neg\phi$$

$$(\phi \to \top) \Rightarrow_{\mathsf{OCNF}} \top$$

$$(\bot \to \phi) \Rightarrow_{\mathsf{OCNF}} \top$$

$$(\top \to \phi) \Rightarrow_{\mathsf{OCNF}} \phi$$

Optimized CNF

Step 2: Introduce top-down fresh variables for beneficial subformulas:

$$\psi[\phi]_p \Rightarrow_{\text{OCNF}} \psi[P]_p \land \text{def}(\psi, p)$$

where P is new to $\psi[\phi]_p$, $\text{def}(\psi, p)$ is defined polarity dependent according to Proposition 2.10 and $\nu(\psi[\phi]_p) > \nu(\psi[P]_p \land \text{def}(\psi, p))$.

Remark: Although computing ν is not practical in general, the test $\nu(\psi[\phi]_p) > \nu(\psi[P]_p \land \text{def}(\psi, p))$ can be computed in constant time.

Optimized CNF

Step 3: Eliminate equivalences polarity dependent:

$$\psi[\phi \leftrightarrow \psi]_p \;\Rightarrow_{\mathsf{OCNF}}\; \psi[(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)]_p$$

if $\mathrm{pol}(\psi, p) = 1$ or $\mathrm{pol}(\psi, p) = 0$

$$\psi[\phi \leftrightarrow \psi]_p \;\Rightarrow_{\mathsf{OCNF}}\; \psi[(\phi \wedge \psi) \vee (\neg\psi \wedge \neg\phi)]_p$$

if $\mathrm{pol}(\psi, p) = -1$

Optimized CNF

Step 4: Apply steps 2, 3, 4, 5 of $\Rightarrow_{\mathrm{ECNF}}$

Remark: The $\Rightarrow_{\mathrm{OCNF}}$ algorithm is already close to a state of the art algorithm. Missing are further redundancy tests and simplification mechanisms we will discuss later on in this section.