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 $\psi[\phi_2]_p$.

Proof. The proof proceeds by induction over the formula structure of ψ.

Each of the formulas $\bot$, $\top$, and P for $P \in \Sigma$ contains only one subformula, namely itself. Hence, if $\psi = \psi[\phi_1]_\epsilon$ equals $\bot$, $\top$, or P, then $\psi[\phi_1]_\epsilon = \phi_1$, $\psi[\phi_2]_\epsilon = \phi_2$, and we are done by assumption.

If $\psi = \psi_1 \wedge \psi_2$, then either $p = \epsilon$ (this case is treated as above), or ϕ_1 is a subformula of ψ_1 or ψ_2 at position $1p'$ or $2p'$, respectively. Without loss of generality, assume that ϕ_1 is a subformula of ψ_1, so $\psi = \psi_1[\phi_1]_{p'} \wedge \psi_2$. By the induction hypothesis, $\psi_1[\phi_1]_{p'}$ and $\psi_1[\phi_2]_{p'}$ are equivalent. Hence, for any valuation $\mathcal{A}$, $\mathcal{A}(\psi[\phi_1]_{1p'}) = \mathcal{A}(\psi_1[\phi_1]_{p'} \wedge \psi_2) = \min(\{\mathcal{A}(\psi_1[\phi_1]_{p'}), \mathcal{A}(\psi_2)\}) = \min(\{\mathcal{A}(\psi_1[\phi_2]_{p'}), \mathcal{A}(\psi_2)\}) = \mathcal{A}(\psi_1[\phi_2]_{p'} \wedge \psi_2) = \mathcal{A}(\psi[\phi_2]_{1p'})$. The other boolean connectives are handled analogously. □

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* $\vee\wedge$

$(\phi \wedge \phi) \leftrightarrow \phi$	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 $\top$

$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$	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 $\neg\bot$

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

$(\phi \to \psi) \leftrightarrow (\neg\phi \vee \psi)$	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?

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:

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

Step 2: Eliminate implications:

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

Step 3: Push negations downward:

$$\neg(\phi \lor \psi) \Rightarrow_{\text{ECNF}} (\neg\phi \land \neg\psi)$$
$$\neg(\phi \land \psi) \Rightarrow_{\text{ECNF}} (\neg\phi \lor \neg\psi)$$

Step 4: Eliminate multiple negations:

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

Step 5: Push disjunctions downward:

$$(\phi \land \psi) \lor \chi \Rightarrow_{\text{ECNF}} (\phi \lor \chi) \land (\psi \lor \chi)$$

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

$$(\phi \land \top) \Rightarrow_{\text{ECNF}} \phi$$
$$(\phi \land \bot) \Rightarrow_{\text{ECNF}} \bot$$
$$(\phi \lor \top) \Rightarrow_{\text{ECNF}} \top$$
$$(\phi \lor \bot) \Rightarrow_{\text{ECNF}} \phi$$
$$\neg\bot \Rightarrow_{\text{ECNF}} \top$$
$$\neg\top \Rightarrow_{\text{ECNF}} \bot$$

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

For step 1, we can prove termination in the following way: We define a function μ from formulas to positive integers such that $\mu(\bot) = \mu(\top) = \mu(P) = 1$, $\mu(\neg\phi) = \mu(\phi)$, $\mu(\phi \land \psi) = \mu(\phi \lor \psi) = \mu(\phi \rightarrow \psi) = \mu(\phi) + \mu(\psi)$, and $\mu(\phi \leftrightarrow \psi) = 2\mu(\phi) + 2\mu(\psi) + 1$. Observe that μ is constructed in such a way that $\mu(\phi_1) > \mu(\phi_2)$ implies $\mu(\psi[\phi_1]_p) > \mu(\psi[\phi_2]_p)$ for all formulas ϕ_1, ϕ_2, and ψ and positions p. Using this property, we can show that whenever a formula χ' is the result of applying the rule of step 1 to a formula χ, then $\mu(\chi) > \mu(\chi')$. Since μ takes only positive integer values, step 1 must terminate.

Termination of steps 3 and 5 is proved similarly. For step 3, we use a function μ from formulas to positive integers such that $\mu(\bot) = \mu(\top) = \mu(P) = 1$, $\mu(\neg\phi) = 2\mu(\phi)$, $\mu(\phi \land \psi) = \mu(\phi \lor \psi) = \mu(\phi \rightarrow \psi) = \mu(\phi \leftrightarrow \psi) = \mu(\phi) + \mu(\psi) + 1$. Whenever a formula χ' is the result of applying a rule of step 3 to a formula χ, then $\mu(\chi) > \mu(\chi')$. Since μ takes only positive integer values, step 3 must terminate.

For step 5, we use a function μ from formulas to positive integers such that $\mu(\bot) = \mu(\top) = \mu(P) = 1$, $\mu(\neg\phi) = \mu(\phi) + 1$, $\mu(\phi \land \psi) = \mu(\phi \rightarrow \psi) = \mu(\phi \leftrightarrow \psi) = \mu(\phi) + \mu(\psi) + 1$, and $\mu(\phi \lor \psi) = 2\mu(\phi)\mu(\psi)$. Again, if a formula χ' is the result of applying a rule of step 5 to a formula χ, then $\mu(\chi) > \mu(\chi')$. Since μ takes only positive integer values, step 5 terminates, too.

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. $\square$

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.

Idea: A formula $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (P \leftrightarrow \phi)$ 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 $\psi[(\phi_1 \wedge \phi_2) \vee (\neg\phi_1 \wedge \neg\phi_2)]_p$.

Proposition 2.10 *Let P be a propositional variable not occurring in $\psi[\phi]_p$.*

If $\mathrm{pol}(\psi, p) = 1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \wedge (P \rightarrow \phi)$ is satisfiable.

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

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.

Proof. Exercise. □

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 $\wedge$, $\vee$:

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

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

$$\begin{aligned}(\phi \to \bot) &\Rightarrow_{\text{OCNF}} \neg\phi\\(\phi \to \top) &\Rightarrow_{\text{OCNF}} \top\\(\bot \to \phi) &\Rightarrow_{\text{OCNF}} \top\\(\top \to \phi) &\Rightarrow_{\text{OCNF}} \phi\end{aligned}$$

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

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

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

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

Step 3: Eliminate equivalences polarity dependent:

$$\psi[\phi \leftrightarrow \psi]_p \Rightarrow_{\text{OCNF}} \psi[(\phi \to \psi) \wedge (\psi \to \phi)]_p$$

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

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

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

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

Remark: The $\Rightarrow_{\text{OCNF}}$ 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.