# Digital Logic Design: a rigorous approach ©

## Chapter 6: Propositional Logic

(part 4)

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

April 5, 2020

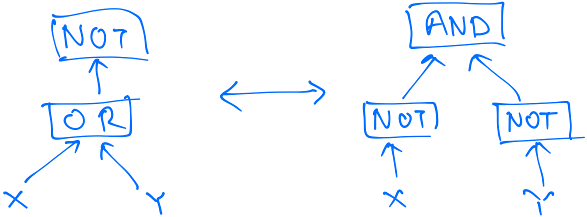
Book Homepage:
http://www.eng.tau.ac.il/~guy/Even-Medina

# De Morgan's Laws

**Theorem (De Morgan's Laws)**

*The following two Boolean formulas are tautologies:*

1. $(\neg(X + Y)) \leftrightarrow (\bar{X} \cdot \bar{Y})$.
2. $(\neg(X \cdot Y)) \leftrightarrow (\bar{X} + \bar{Y})$.



# De Morgan Dual

Given a Boolean Formula $\varphi \in \mathcal{BF}(U, \{\vee, \wedge, \neg\})$, apply the following "replacements":

- $X_i \mapsto \neg X_i$
- $\neg X_i \mapsto X_i$
- $\vee \mapsto \wedge$
- $\wedge \mapsto \vee$

What do you get?

## Example

$$\varphi = (X_1 + \neg X_2) \cdot (\neg X_2 + X_3)$$

is replaced by

$$\mathrm{dual}(\varphi) = (\neg X_1 \cdot X_2) + (X_2 \cdot \neg X_3).$$

What is the semantic relation between $\varphi$ and $\mathrm{dual}(\varphi)$?

We define the De Morgan Dual using a recursive algorithm.

**Algorithm 3** DM($\varphi$) - An algorithm for computing the De Morgan dual of a Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$.

1. Base Cases:
   1. If $\varphi = 0$, then return 1. If $\varphi = 1$, then return 0.
   2. If $\varphi = (\neg 0)$, then return 0. If $\varphi = (\neg 1)$, then return 1.
   3. If $\varphi = X_i$, then return $(\neg X_i)$.
   4. If $\varphi = (\neg X_i)$, then return $X_i$.
2. Reduction Rules:
   1. If $\varphi = (\neg \varphi_1)$, then return $(\neg \text{DM}(\varphi_1))$.
   2. If $\varphi = (\varphi_1 \cdot \varphi_2)$, then return $(\text{DM}(\varphi_1) + \text{DM}(\varphi_2))$.
   3. If $\varphi = (\varphi_1 + \varphi_2)$, then return $(\text{DM}(\varphi_1) \cdot \text{DM}(\varphi_2))$.

**Example**

$\text{DM}(X \cdot (\neg Y))$.

$$DM(X \cdot \overline{Y})$$ $$DM(\varphi_1 \cdot \varphi_2) = DM(\varphi_1) + DM(\varphi_2)$$

$$= DM(X) + DM(\overline{Y})$$

$$= \overline{X} + Y$$ $$DM(X) = \overline{X}$$ $$DM(\overline{Y}) = Y$$

---

$$DM(\text{not}(X + Y))$$ $$DM(\neg\varphi) = \neg DM(\varphi)$$

$$= \text{not}((DM(X + Y)))$$ $$DM(\varphi_1 + \varphi_1) = DM(\varphi_1) \cdot DM(\varphi_2)$$

$$= \text{not}(DM(X) \cdot DM(Y))$$ $$DM(X) = \overline{X}$$

$$= \text{not}(\overline{X} \cdot \overline{Y})$$

**Exercise**

Prove that $\mathrm{DM}(\varphi) \in \mathcal{BF}$.

The dual can be obtained by applying replacements to the labels in the parse tree of $\varphi$ or directly to the "characters" of the string $\varphi$.

**Theorem**

*For every Boolean formula $\varphi$, $DM(\varphi)$ is logically equivalent to $(\neg\varphi)$.*

**Corollary**

For every Boolean formula $\varphi$, $DM(DM(\varphi))$ is logically equivalent to $\varphi$.

Nice trick, but is it of any use?!

$x \Leftrightarrow \neg(\neg(x))$

THM: $DM(\varphi) \iff \neg\varphi$

proof complete ind. on size $n$ of parse tree.

basis $n = 1,2$ : $\varphi \in \{0, 1, x_i, \text{not}(x_i)\}$

check ! not(0), not(1)

hyp: size of parse tree of $\varphi \leq n$

$\Rightarrow$ $DM(\varphi) \iff \neg\varphi$

step : $\varphi \in \{\neg\varphi_1 , \varphi_1 + \varphi_2 , \varphi_1 \cdot \varphi_2\}$

$\varphi = \varphi_1 + \varphi_2$ : $DM(\varphi) = DM(\varphi_1) \cdot DM(\varphi_2)$

(ind. hyp + substitution) $\iff$ $\overline{\varphi_1} \cdot \overline{\varphi_2}$

(de-Morgan TAUT + substitution) $\iff$ $\text{not}(\varphi_1 + \varphi_2)$

$= \neg\varphi$

$\varphi = \varphi_1 \circ \varphi_2$ exercise!

$\varphi = \neg \varphi_1$

$$DM(\varphi) = \neg DM(\varphi_1)$$

ind. hyp
+ substi.

$$\Longleftrightarrow \neg(\neg \varphi_1)$$

$$= \neg \varphi$$

$\boxtimes$

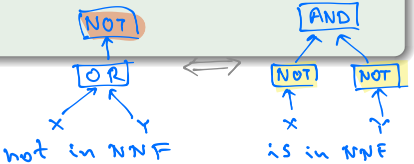
A formula is in negation normal form if negation is applied only directly to variables or constants. ($\neg 0 = 1$, $\neg 1 = 0$, so we can easily eliminate negations of constants)

## Definition

A Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$ is in negation normal form if the parse tree $(G, \pi)$ of $\varphi$ satisfies the following condition. If a vertex in $G$ is labeled by negation (i.e., $\pi(v) = \neg$), then $v$ is a parent of a leaf.

## Example

- $\neg(X_1 + X_2)$ and $(\neg X_1 \cdot \neg X_2)$.
- $\neg(X_1 \cdot \neg X_2)$ and $(\neg X_1 + X_2)$.



# Negation Normal Form

**Definition**

A Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$ is in negation normal form if the parse tree $(G, \pi)$ of $\varphi$ satisfies the following condition. If a vertex in $G$ is labeled by negation (i.e., $\pi(v) = \neg$), then $v$ is a parent of a leaf.

**Lemma**

*If $\varphi$ is in negation normal form, then so is $DM(\varphi)$.* exercise!

We present an algorithm $NNF(\varphi)$ that transforms a Boolean formula $\varphi$ into a logically equivalent formula in negation normal form.

**Algorithm 4** NNF($\varphi$) - An algorithm for computing the negation normal form of a Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$.

1. Base Cases: If $\varphi \in \{0, 1, X_i, (\neg X_i), \neg 0, \neg 1\}$, then return $\varphi$.
2. Reduction Rules:
    1. If $\varphi = (\neg \varphi_1)$, then return DM(NNF($\varphi_1$)).
    2. If $\varphi = (\varphi_1 \cdot \varphi_2)$, then return (NNF($\varphi_1$) $\cdot$ NNF($\varphi_2$)).
    3. If $\varphi = (\varphi_1 + \varphi_2)$, then return (NNF($\varphi_1$) $+$ NNF($\varphi_2$)).

**Theorem**

*Let $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$. Then, NNF($\varphi$) is logically equivalent to $\varphi$ and in negation normal form.*

NNF($\varphi$)
- it is Boolean Formula. (exercise)
- equiv. to $\varphi$.
- it is in NNF.

* for simplicity we allow not(0), not(1). Clearly, one could eliminate such negations.

$\Rightarrow$ do not get confused by this point!

THM: $NNF(\varphi) \Leftrightarrow \varphi$ and $NNF(\varphi)$ in NNF.

proof by comp. ind. on $n$ size of parse tree of $\varphi$.

Fill details by yourself:

basis: $n \in \{1, 2\}$ ....

hyp: ....

step: $\varphi \in \{ not(\varphi_1), \varphi_1 + \varphi_2, \varphi_1 \cdot \varphi_2 \}$

cases: $\varphi = \varphi_1 \cdot \varphi_2$ ...., $\varphi = \varphi_1 + \varphi_2$ ....

$\varphi = \neg \varphi_1$: $NNF(\varphi) = DM(NNF(\varphi_1))$

ind. hyp. + subst. $\Leftrightarrow DM(\varphi_1)$

$\Leftrightarrow \neg \varphi_1$

$= \varphi$

$$\left.\begin{array}{l} \\ \\ \end{array}\right\} \begin{array}{c} NNF(\varphi) \\ \Leftrightarrow \\ \varphi \end{array}$$

prove that $NNF(\varphi)$ is NNF:

$\varphi = \neg \varphi_1$

$NNF(\varphi) = DM(NNF(\varphi_1))$

$NNF(\varphi_1)$ is NNF (ind. hyp.)

$DM(NNF(\varphi_1))$ is NNF (DM preserves NNF)

$\boxtimes$