Digital Logic Design: a rigorous approach ©

Chapter 6: Propositional Logic

(part 3)

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

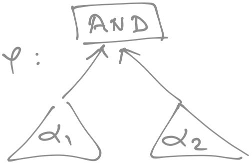
March 29, 2020

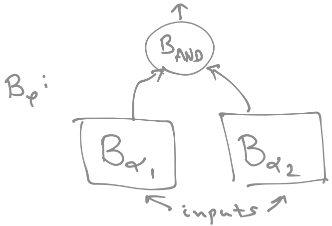
Book Homepage:
http://www.eng.tau.ac.il/~guy/Even-Medina

If $\varphi = (\alpha_1 \text{ AND } \alpha_2)$, then

$$\begin{aligned}
B_\varphi(v) &= \hat{\tau}_v(\varphi) \\
&= \hat{\tau}_v(\alpha_1 \text{ AND } \alpha_2) \\
&= B_{\text{AND}}(\hat{\tau}_v(\alpha_1), \hat{\tau}_v(\alpha_2)) \\
&= B_{\text{AND}}(B_{\alpha_1}(v), B_{\alpha_2}(v)).
\end{aligned}$$

Thus, we can express complicated Boolean functions by composing long Boolean formulas.



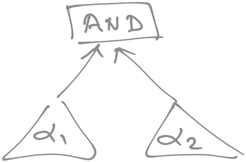


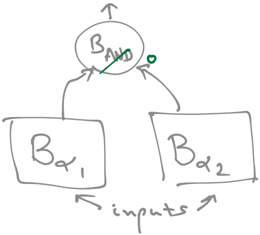
Composition of Boolean formulas

Lemma

If $\varphi = \alpha_1 \circ \alpha_2$ *for a binary connective* $\circ$*, then*

$$\forall v \in \{0,1\}^n : \quad B_\varphi(v) = B_\circ(B_{\alpha_1}(v), B_{\alpha_2}(v)).$$





Claim

Two Boolean formulas p and q are logically equivalent if and only if the formula $(p \leftrightarrow q)$ is a tautology.

p log. equiv q $\iff \forall \tau : \hat{\tau}(p) = \hat{\tau}(q)$

$\iff \forall \tau : B_{\leftrightarrow}(\hat{\tau}(p), \hat{\tau}(q)) = 1$

$\iff \forall v : B_{\leftrightarrow}(B_p(v), B_q(v)) = 1$

$\iff \forall v : B_{p \leftrightarrow q}(v) = 1$

$\iff p \leftrightarrow q$ TAUT. $\boxtimes$

Substitution is used to compose large formulas from smaller ones. For simplicity, we deal with substitution in formulas over two variables; the generalization to formulas over any number of variables is straightforward.

1. $\varphi \in \mathcal{BF}(\{X_1, X_2\}, \mathcal{C})$,
2. $\alpha_1, \alpha_2 \in \mathcal{BF}(U, \mathcal{C})$.
3. (G_φ, π_φ) denotes the parse tree of φ.

Definition

Substitution of α_i in φ yields the Boolean formula $\varphi(\alpha_1, \alpha_2) \in \mathcal{BF}(U, \mathcal{C})$ that is generated by the parse tree (G, π) defined as follows.
For every leaf of $v \in G_\varphi$ that is labeled by a variable X_i, replace the leaf v by a new copy of $(G_{\alpha_i}, \pi_{\alpha_i})$.

example: substitution

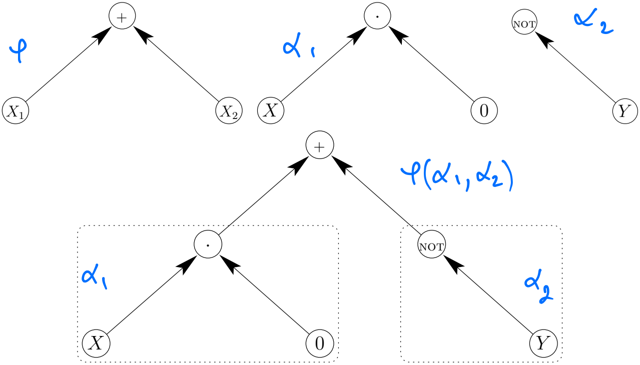
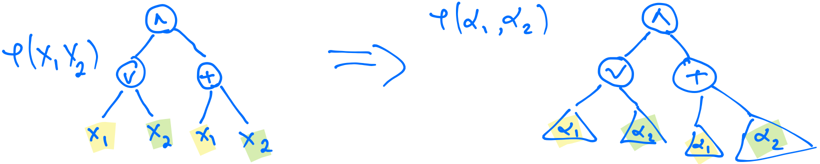


Figure: φ, α_1, α_2, $\varphi(\alpha_1, \alpha_2)$



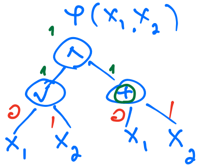
Substitution can be obtain by applying a simple "find-and-replace", where each instance of variable X_i is replaced by a copy of the formula α_i, for $i \in \{1, 2\}$.

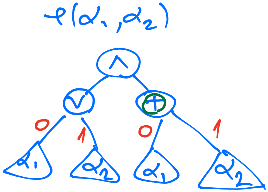
One can easily generalize substitution to formulas $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_k\}, \mathcal{C})$ for any $k > 2$. In this case, $\varphi(\alpha_1, \ldots, \alpha_k)$ is obtained by replacing every instance of X_i by α_i.

Lemma

For every assignment $\tau : U \to \{0,1\}$,

$$\hat{\tau}(\varphi(\alpha_1,\alpha_2)) = B_\varphi(\hat{\tau}(\alpha_1),\hat{\tau}(\alpha_2)). \quad (1)$$





$\forall \tau: \quad \hat{\tau}(\varphi(\alpha_1, \alpha_2)) = B_\varphi(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2))$

proof comp. ind. on #vertices in parse tree of φ. (called: "size" of φ)

base: #vertices = 1: $\varphi \in \{0, 1, x_1, x_2\}$

$\varphi = 0$: $\varphi(\alpha_1, \alpha_2) = 0$ & B_φ const 0

LHS: $\hat{\tau}(0) = 0$

RHS: $B_\varphi(\dots) = 0$

$\boxed{\text{Try to prove for } \varphi = 1}$

$\varphi = x_1$ $\quad \varphi(\alpha_1, \alpha_2) = \alpha_1$ & $B_\varphi(b_1, b_2) = b_1$

LHS: $\hat{\tau}(\varphi(\alpha_1, \alpha_2)) = \hat{\tau}(\alpha_1)$

RHS: $B_\varphi(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2)) = \hat{\tau}(\alpha_1)$

$\boxed{Q: \varphi = x_2}$

ind hyp: $\forall \varphi$: #vert. in parse tree $\leq n$
claim holds.

step: consider φ s.t. #vertices $= n+1$.

2 cases: $\varphi = \text{not}(\varphi_1)$ (exercise)

$\varphi = \varphi_1 * \varphi_2$

bin. connective

Suppose $\varphi = \varphi_1 * \varphi_2$

incl. hyp. $\hat{\tau}(\varphi_i(\alpha_1, \alpha_2)) = B_{\varphi_i}(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2))$

$$\hat{\tau}(\varphi(\alpha_1, \alpha_2)) = B_*(\hat{\tau}(\varphi_1(\alpha_1, \alpha_2)), \hat{\tau}(\varphi_2(\alpha_1, \alpha_2)))$$

$$= B_*(B_{\varphi_1}(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2)), B_{\varphi_2}(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2)))$$

$$= B_{\varphi_1 * \varphi_2}(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2))$$

$\boxtimes$

Let

- $\varphi \in \mathcal{BF}(\{X_1, X_2\}, \mathcal{C})$,
- $\alpha_1, \alpha_2 \in \mathcal{BF}(U, \mathcal{C})$,
- $\tilde{\varphi} \in \mathcal{BF}(\{X_1, X_2\}, \tilde{\mathcal{C}})$,
- $\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{BF}(U, \tilde{\mathcal{C}})$.

Corollary

If α_i and $\tilde{\alpha}_i$ are logically equivalent, and φ and $\tilde{\varphi}$ are logically equivalent, then $\varphi(\alpha_1, \alpha_2)$ and $\tilde{\varphi}(\tilde{\alpha}_1, \tilde{\alpha}_2)$ are logically equivalent.

Example

$$\varphi = \neg(X_1 \cdot X_2) \qquad \tilde{\varphi} = \bar{X}_1 + \bar{X}_2$$

$$\alpha_1 = A \to B \qquad \tilde{\alpha}_1 = \bar{A} + B$$

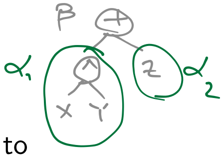
$$\alpha_2 = C \leftrightarrow D \qquad \tilde{\alpha}_2 = \neg(C \oplus D)$$

$\varphi(\alpha_1, \alpha_2) = \neg((A \to B) \cdot (C \leftrightarrow D))$ $\quad$ $\tilde{\varphi}(\tilde{\alpha}_1, \tilde{\alpha}_2) = \overline{(\bar{A} + B)} + \overline{\neg(C \oplus D)}$

example: changing connectives

Let $\mathcal{C} = \{\text{AND}, \text{XOR}\}$. We wish to find a formula $\tilde{\beta} \in \mathcal{BF}(\{X, Y, Z\}, \mathcal{C})$ that is logically equivalent to the formula

$$\beta \triangleq (X \cdot Y) + Z.$$



Parse β: $\varphi(\alpha_1, \alpha_2)$ with $\alpha_1 = (X \cdot Y)$ and $\alpha_2 = Z$.
Find $\tilde{\varphi} \in \mathcal{BF}(\{X_1, X_2\}, \mathcal{C})$ that is logically equivalent to $\varphi \triangleq (X_1 + X_2)$.

$$\tilde{\varphi} \triangleq X_1 \oplus X_2 \oplus (X_1 \cdot X_2).$$

$\varphi = x_1 + x_2$

$\alpha_1 = X \cdot Y = \tilde{\alpha}_1$

$\alpha_2 = Z = \tilde{\alpha}_2$

find $\tilde{\varphi}$!
(using $\wedge, \oplus$)

Apply substitution to define $\tilde{\beta} \triangleq \tilde{\varphi}(\alpha_1, \alpha_2)$, thus

$$\begin{aligned}\tilde{\beta} &\triangleq \tilde{\varphi}(\alpha_1, \alpha_2) \\ &= \alpha_1 \oplus \alpha_2 \oplus (\alpha_1 \cdot \alpha_2) \\ &= (X \cdot Y) \oplus Z \oplus ((X \cdot Y) \cdot Z)\end{aligned}$$

Indeed $\tilde{\beta}$ is logically equivalent to β.

CORO: $\varphi \Leftrightarrow \hat{\varphi}$, $\alpha_i \Leftrightarrow \hat{\alpha}_i$ $\Longrightarrow$ $\varphi(\alpha_1, \alpha_2) \Leftrightarrow \tilde{\varphi}(\tilde{\alpha}_1, \tilde{\alpha}_2)$

proof: suffice to prove

$$\forall v \in \{0,1\}^{|U|}: \quad \hat{\tau}_v(\varphi(\alpha_1, \alpha_2)) = \hat{\tau}_v(\tilde{\varphi}(\tilde{\alpha}_1, \hat{\alpha}_2))$$

indeed:

$$\hat{\tau}_v(\varphi(\alpha_1, \alpha_2)) = B_\varphi(\hat{\tau}_v(\alpha_1), \hat{\tau}(\alpha_2))$$

$$= B_{\tilde{\varphi}}(\hat{\tau}_v(\tilde{\alpha}_1), \hat{\tau}(\tilde{\alpha}_2))$$

$$= \hat{\tau}_v(\tilde{\varphi}(\tilde{\alpha}_1, \tilde{\alpha}_2))$$

$\boxtimes$

p formula $\longmapsto$ B_p Boolean func.

Every Boolean formula can be interpreted as Boolean function. In this section we deal with the following question: Which sets of connectives enable us to express every Boolean function?

Definition

A Boolean function $B : \{0,1\}^n \to \{0,1\}$ is expressible by $\mathcal{BF}(\{X_1, \ldots, X_n\}, \mathcal{C})$ if there exists a formula $p \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \mathcal{C})$ such that $B = B_p$.

Definition

A set $\mathcal{C}$ of connectives is complete if every Boolean function $B : \{0,1\}^n \to \{0,1\}$ is expressible by $\mathcal{BF}(\{X_1, \ldots, X_n\}, \mathcal{C})$.

$\forall B \; \exists p : \quad B = B_p$

Completeness of $\{\neg, \text{AND}, \text{OR}\}$

Theorem

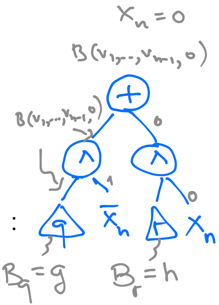
The set $\mathcal{C} = \{\neg, \text{AND}, \text{OR}\}$ is a complete set of connectives.

Proof Outline: Induction on n (the arity of Boolean function).

1. Induction basis for $n = 1$. (exercise)
2. Induction step for $B : \{0,1\}^n \to \{0,1\}$ define:

$g,h : \{0,1\}^{n-1} \to \{0,1\}$

$$g(v_1, \ldots, v_{n-1}) \triangleq B(v_1, \ldots, v_{n-1}, 0),$$
$$h(v_1, \ldots, v_{n-1}) \triangleq B(v_1, \ldots, v_{n-1}, 1).$$



3. By induction hyp. $\exists r, q \in \mathcal{BF}(\{X_1, \ldots, X_{n-1}\}, \mathcal{C})$: $h = B_r$ and $B_q = g$
4. Prove that $B_p = B$ for the formula p defined by

$$p \triangleq (q \cdot \bar{X}_n) + (r \cdot X_n)$$

Theorem: changing connectives

Theorem

If the Boolean functions in $\{\text{NOT}, \text{AND}, \text{OR}\}$ *are expressible by formulas in* $\mathcal{BF}(\{X_1, X_2\}, \tilde{\mathcal{C}})$*, then* $\tilde{\mathcal{C}}$ *is a complete set of connectives.*

Proof Outline:

1. Express $\beta \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \mathcal{C})$ by a logically equivalent formula $\tilde{\beta} \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \tilde{\mathcal{C}})$.
2. How? induction on the parse tree that generates β.

THM: B_{NOT}, B_{OR}, B_{AND} express. in $BF((x_1,x_2),\tilde{C})$

$\Rightarrow$ $\tilde{C}$ is a complete set of connec.

comp of $\{\neg, OR, AND\} \Rightarrow$

proof: $\forall$ func B $\exists \beta \in BF(\{x_i\}_i, \{NOT, OR, AND\})$

s.t. $B = B_\beta$.

goal: find $\tilde{\beta} \Longleftrightarrow \beta$ where $\tilde{\beta} \in BF(\{x_i\}_i, \tilde{C})$.

how? comp. ind. on size n of parse tree of β (#vert.)

base: $n=1$ (exercise). $\beta \in \{0, 1, x_i\}$

hyp: holds if $\text{size}_\beta \leq n$.

step: $\beta = \alpha_1 \wedge \alpha_2$.

let $\tilde{\alpha}_i \in BF(\{x_i\}_i, \tilde{C})$ s.t $\alpha_i \Longleftrightarrow \tilde{\alpha}_i$. (ind. hyp.)

let $\varphi_{AND} \in BF(\{x_i\}_i, \tilde{C})$ s.t. $x_1 \cdot x_2 \Longleftrightarrow \varphi_{AND}$
$\{x_1, x_2\}$

so: $\beta = \alpha_1 \wedge \alpha_2 \Longleftrightarrow \varphi_{AND}(\tilde{\alpha}_1, \tilde{\alpha}_2)$.

NOT, OR, ex.

Important Tautologies

φ TAUT : $\forall\, \tau$: $\hat{\tau}(\varphi) = 1$
truth assign.

Theorem

The following Boolean formulas are tautologies.

1. *law of excluded middle:* $X + \bar{X}$
2. *double negation:* $X \leftrightarrow (\neg\neg X)$
3. *modus ponens:* $(((X \rightarrow Y) \cdot X) \rightarrow Y)$
4. *contrapositive:* $(X \rightarrow Y) \leftrightarrow (\bar{Y} \rightarrow \bar{X})$
5. *material implication:* $(X \rightarrow Y) \leftrightarrow (\bar{X} + Y)$.
6. *distribution:* $X \cdot (Y + Z) \leftrightarrow (X \cdot Y + X \cdot Z)$.

how can we verify that these are tautologies?

Substitution in Tautologies

$X_1 + \bar{X}_1$ TAUT

Recall the lemma: φ TAUT $\Leftrightarrow$ $B_\varphi \equiv 1$

Lemma

For every assignment $\tau : U \to \{0,1\}$,

$$\hat{\tau}(\varphi(\alpha_1, \alpha_2)) = B_\varphi(\hat{\tau}(\alpha_1), \hat{\tau}(\alpha_2)). \tag{2}$$

question

Let α_1 and α_2 be any Boolean formulas.

1. Consider the Boolean formula $\varphi \triangleq \alpha_1 + \text{NOT}(\alpha_1)$. Prove or refute that φ is a tautology.
2. Consider the Boolean formula $\varphi \triangleq (\alpha_1 \to \alpha_2) \leftrightarrow (\text{NOT}(\alpha_1) + \alpha_2)$. Prove or refute that φ is a tautology.

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

proof of de Morgan Law (for sets!)

$$\overline{A \cup B} = \bar{A} \cap \bar{B}$$

$$
\begin{aligned}
x \in \overline{A \cup B} &\iff x \notin A \cup B \\
&\iff \text{not}\,(x \in A \cup B) \\
&\iff \text{not}\,\big(\underbrace{(x \in A)}_{Y \hat{=}} \ \text{OR} \ \underbrace{(x \in B)}_{Z \hat{=}}\big) \\
&\iff \text{not}\,(Y + Z) \\
&\iff \bar{Y} \wedge \bar{Z} \\
&\iff \text{not}\,(x \in A) \wedge \text{not}\,(x \in B) \\
&\iff x \in \bar{A} \wedge x \in \bar{B} \\
&\iff x \in \bar{A} \cap \bar{B} \qquad \boxtimes
\end{aligned}
$$