Digital Logic Design: a rigorous approach ©

Chapter 6: Propositional Logic

part 2

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

Syntax vs. Semantics

- Syntax - grammatic rules that govern the construction of Boolean formulas (rules: parse trees + inorder traversal)
- Semantics - functional interpretation of a formula

Syntax has a purpose: to provide well defined semantics!

Syntax vs. Semantics

TABLE  

Logical connectives have two roles:

- Syntax: building block for Boolean formulas ("glue").
- Semantics: define a truth value based on a Boolean function.

To emphasize the semantic role: given a k-ary connective $*$, we denote the semantics of $*$ by a Boolean function

$$B_* : \{0,1\}^k \to \{0,1\}$$

.

Example

- $B_{\mathrm{AND}}(b_1, b_2) = b_1 \cdot b_2$.
- $B_{\mathrm{NOT}}(b) = 1 - b$.

Semantics of Variables and Constants

- The function B_X associated with a variable X is the identity function $B_X(b) = b$.
- The function B_σ associated with a constant $\sigma \in \{0, 1\}$ is the constant function $B_\sigma(b) = \sigma$.

$$B_0(1) = 0$$

$$B_1(0) = 1$$

A = "today is Monday"
B = "this is written in blue"
$\tau(A) = 1, \quad \tau(B) = 0$

Let U denote the set of variables.

Definition

A truth assignment is a function $\tau : U \to \{0, 1\}$.

Our goal is to extend every assignment $\tau : U \to \{0, 1\}$ to a function

$$\hat{\tau} : \mathcal{BF}(U, \mathcal{C}) \to \{0, 1\}$$

Thus, a truth assignment to variables, actually induces truth values to every Boolean formula.

$\hat{\tau}(A \vee B) = 1, \quad \hat{\tau}(A \wedge B) = 0, \quad \hat{\tau}(\bar{B}) = 1$

The extension $\hat{\tau} : \mathcal{BF} \to \{0,1\}$ of an assignment $\tau : U \to \{0,1\}$ is defined as follows.

Definition

Let $p \in \mathcal{BF}$ be a Boolean formula generated by a parse tree (G, π). Then,

$$\hat{\tau}(p) \triangleq \mathrm{EVAL}(G, \pi, \tau),$$

where EVAL is listed in the next slide.

EVAL is also an algorithm that also employs inorder traversal over the parse tree!

Algorithm 2 EVAL(G, π, τ) - evaluate the truth value of the Boolean formula generated by the parse tree (G, π), where (i) $G = (V, E)$ is a rooted tree with in-degree at most 2, (ii) $\pi : V \to \{0,1\} \cup U \cup \mathcal{C}$, and (iii) $\tau : U \to \{0,1\}$ is an assignment.

1. Base Case: If $|V| = 1$ then
   1. Let $v \in V$ be the only node in V.
   2. $\pi(v)$ is a constant: If $\pi(v) \in \{0,1\}$ then return $(\pi(v))$.
   3. $\pi(v)$ is a variable: return $(\tau(\pi(v)))$.
2. Reduction Rule:
   1. If $deg_{in}(r(G)) = 1$, then *(in this case $\pi(r(G)) = \text{NOT}$)*
      1. Let $G_1 = (V_1, E_1)$ denote the rooted tree hanging from $r(G)$.
      2. Let π_1 denote the restriction of π to V_1.
      3. $\sigma \leftarrow$ EVAL(G_1, π_1, τ).
      4. Return $(\text{NOT}(\sigma))$.
   2. If $deg_{in}(r(G)) = 2$, then
      1. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ denote the rooted subtrees hanging from $r(G)$.
      2. Let π_i denote the restriction of π to V_i.
      3. $\sigma_1 \leftarrow$ EVAL(G_1, π_1, τ).
      4. $\sigma_2 \leftarrow$ EVAL(G_2, π_2, τ).
      5. Return $(B_{\pi(r(G))}(\sigma_1, \sigma_2))$.

$$\mathrm{EVAL}(G, \tau)$$

parse tree of p

truth assign

omitted π on purpose

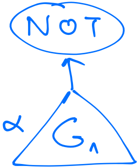
base:

$0 \longmapsto 0$

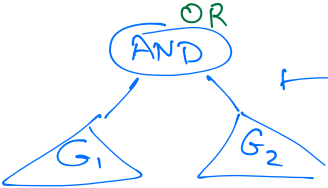
$1 \longmapsto 1$

$x \longmapsto \tau(x) \in \{0,1\}$

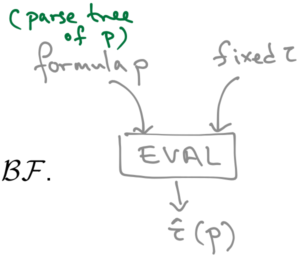
reduction:



$\longmapsto \mathrm{NOT}(\underbrace{\mathrm{EVAL}(G_1, \tau)}_{\in \{0,1\}})$



$\longmapsto$ AND / OR $\left(\overbrace{\mathrm{EVAL}(G_1, \tau)}^{\in \{0,1\}}, \underbrace{\mathrm{EVAL}(G_2, \tau)}_{\in \{0,1\}}\right)$

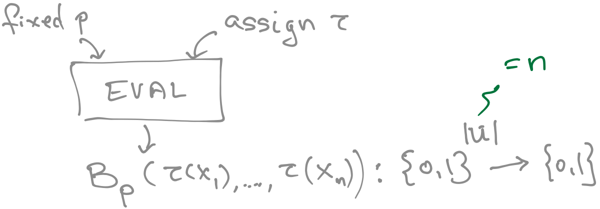


Evaluation:

- Fix a truth assignment $\tau : U \to \{0,1\}$.
- Extended τ to every Boolean formula $p \in \mathcal{BF}$.

Formula as a function:

- Fix a Boolean formula p.
- Consider all possible truth assignments $\tau : U \to \{0,1\}$.



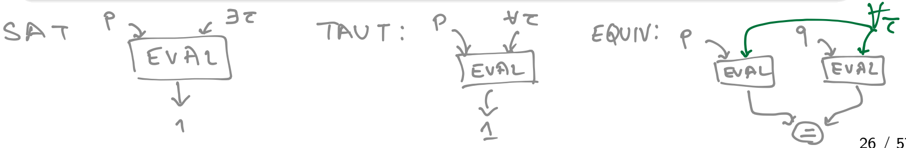
Definition

Let p denote a Boolean formula.

1. p is satisfiable if there exists an assignment τ such that $\hat{\tau}(p) = 1$.
2. p is a tautology if $\hat{\tau}(p) = 1$ for every assignment τ.

Definition

Two formulas p and q are logically equivalent if $\hat{\tau}(p) = \hat{\tau}(q)$ for every assignment τ.



$\tau(X) = 0$

$\tau(Y) = 1$

$\hat{\tau}(X \oplus Y) = 1$

$0 \oplus 1$

1. Show that $\varphi \triangleq (X \oplus Y)$ is satisfiable.
2. Let $\varphi \triangleq (X \vee \neg X)$. Show that φ is a tautology.

"TO BE"

OR

"NOT TO BE"

$\tau(X)$	$\text{NOT}(\tau(X))$	$\hat{\tau}(X \vee \neg X)$
0	1	1
1	0	1

Let $\varphi \triangleq (X \oplus Y)$, and let $\psi \triangleq (\bar{X} \cdot Y + X \cdot \bar{Y})$. Show that φ and ψ are logically equivalent.
We show that $\hat{\tau}(\varphi) = \hat{\tau}(\psi)$ for every assignment τ. We do that by enumerating all the $2^{|U|}$ assignments.

$\tau(X)$	$\tau(Y)$	$\text{AND}(\text{NOT}(\tau(X)), \tau(Y))$	$\text{AND}(\tau(X), \text{NOT}(\tau(Y)))$	$\hat{\tau}(\varphi)$	$\hat{\tau}(\psi)$
0	0	0	0	0	0
1	0	0	1	1	1
0	1	1	0	1	1
1	1	0	0	0	0

Table: There are two variables, hence the enumeration consists of $2^2 = 4$ assignments. The columns that correspond to $\hat{\tau}(\varphi)$ and $\hat{\tau}(\psi)$ are identical, hence φ and ψ are equivalent.

Lemma

Let $\varphi \in \mathcal{BF}$, then

$$\varphi \textit{ is satisfiable} \Leftrightarrow (\neg\varphi) \textit{ is not a tautology .}$$

Proof.

All the transitions in the proof are "by definition".

$$\begin{aligned}
\varphi \text{ is satisfiable} &\Leftrightarrow \exists\tau : \hat{\tau}(\varphi) = 1 \\
&\Leftrightarrow \exists\tau : \text{NOT}(\hat{\tau}(\varphi)) = 0 \\
&\Leftrightarrow \exists\tau : \hat{\tau}(\neg(\varphi)) = 0 \\
&\Leftrightarrow (\neg\varphi) \text{ is not a tautology .}
\end{aligned}$$

/ NOT

by def EVAL

by def of TAUT

□

Assume that $U = \{X_1, \ldots, X_n\}$.

Definition

Given a binary vector $v = (v_1, \ldots, v_n) \in \{0,1\}^n$, the assignment $\tau_v : \{X_1, \ldots, X_n\} \to \{0,1\}$ is defined by $\tau_v(X_i) \triangleq v_i$.

Example

Let $n = 3$. $\mathcal{U} = \{X_1, X_2, X_3\}$

$$v[1:3] = 011$$
$$\tau_v(X_1) = v[1] = 0$$
$$\tau_v(X_2) = v[2] = 1$$
$$\tau_v(X_3) = v[3] = 1$$

$v \mapsto \tau_v$ is a bijection from $\{0,1\}^n$ to truth assignments ex ∇_0

$$\{\tau \mid \tau : \{X_1, \ldots, X_n\} \to \{0,1\}\}\ .$$

Every Boolean Formula Represents a Function

Syntax Semantics

Assume that $U = \{X_1, \dots, X_n\}$.

Definition

A Boolean formula p over the variables $U = \{X_1, \dots, X_n\}$ defines the Boolean function $B_p : \{0,1\}^n \to \{0,1\}$ by

$v \triangleq (v_1, \dots, v_n)$

$$B_p(v_1, \dots v_n) \triangleq \hat{\tau}_v(p).$$

Example

$$p = X_1 \vee X_2$$

$$B_p(\underbrace{0}_{\tau(X_1)=0}, \underbrace{0}_{\tau(X_2)=0}) = 0, \quad B_p(\underbrace{0}_{\tau(X_1)=0}, \underbrace{1}_{\tau(X_2)=1}) = 1, \dots$$

Assume that $U = \{X_1, \ldots, X_n\}$.

Definition

A Boolean formula p over the variables $U = \{X_1, \ldots, X_n\}$ defines the Boolean function $B_p : \{0,1\}^n \to \{0,1\}$ by

$$B_p(v_1, \ldots v_n) \triangleq \hat{\tau}_v(p).$$

The mapping $p \mapsto B_p$ is a function from $\mathcal{BF}(U, \mathcal{C})$ to set of Boolean functions $\{0,1\}^{(\{0,1\}^n)}$. Is this mapping one-to-one? is it onto?

$\forall f \; \exists p : p \longmapsto f$

$(B_p = f)$

$p_1 \neq p_2$ ← diff. parse trees!

$p_i \longmapsto f_i$

$\Rightarrow f_1 \neq f_2$

Every Tautology Induces a Constant Function

$B_p(v) \doteq \hat{\tau}_v(p)$

Claim

A Boolean formula p is a tautology if and only if the Boolean function B_p is identically one, i.e., $B_p(v) = 1$, for every $v \in \{0,1\}^n$.

Proof.

$\forall \tau \; \exists v$

$\forall v \; \exists \; \tau = \tau_v$

$$\begin{aligned} p \text{ is a tautology} \quad &\Leftrightarrow \quad \forall\, \tau : \hat{\tau}(p) = 1 \\ &\Leftrightarrow \quad \forall\, v \in \{0,1\}^n : \hat{\tau}_v(p) = 1 \\ &\Leftrightarrow \quad \forall\, v \in \{0,1\}^n : B_p(v) = 1\,. \end{aligned}$$

□

what about a satisfiable formula?

$B_p(v) \hat{=} \hat{\tau}_v(p)$

Claim

A Boolean formula p is a satisfiable if and only if the Boolean function B_p is not identically zero, i.e., there exists a vector $v \in \{0,1\}^n$ such that $B_p(v) = 1$.

Proof.

$$
\begin{aligned}
p \text{ is a satisfiable} \quad &\Leftrightarrow \quad \exists\, \tau : \hat{\tau}(p) = 1 \\
&\Leftrightarrow \quad \exists\, v \in \{0,1\}^n : \hat{\tau}_v(p) = 1 \\
&\Leftrightarrow \quad \exists\, v \in \{0,1\}^n : B_p(v) = 1\,.
\end{aligned}
$$

□

equivalent formulas

$B_p(v) \hat{=} \hat{\tau}_v(p)$

Claim

Two Boolean formulas p and q are logically equivalent if and only if the Boolean functions B_p and B_q are identical, i.e., $B_p(v) = B_q(v)$, for every $v \in \{0,1\}^n$.

Proof.

p and q are logically equivalent

$$\begin{aligned} &\Leftrightarrow \quad \forall\, \tau : \hat{\tau}(p) = \hat{\tau}(q) \\ &\Leftrightarrow \quad \forall\, v \in \{0,1\}^n : \hat{\tau}_v(p) = \hat{\tau}_v(q) \\ &\Leftrightarrow \quad \forall\, v \in \{0,1\}^n : B_p(v) = B_q(v)\,. \end{aligned}$$

$\square$