Digital Logic Systems

Recitation 4: Propositional Logic

Guy Even   Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

November 22, 2011

The Logisim Software

- Can be downloaded from: `http://ozark.hendrix.edu/~burch/logisim/`.

Example

The the 3-bit carry function $c : \{0,1\}^3 \to \{0,1\}$ defined as follows.

$\forall (b_1, b_2, b_3) \in \{0,1\}^3 : c(b_1, b_2, b_3) = (b_1 \wedge b_2) \vee (b_1 \wedge b_3) \vee (b_2 \wedge b_3)$.

- Every gate implements a Boolean function. We compose Boolean functions to "construct" new ones.
- Generate the truth table of a Boolean function.
- Simulating a single input:
  - Light green = '1'.
  - Dark green = '0'.
- Substitution. Using a circuit as a module in another circuit.
- Printing designs and truth tables.
- "Minimization heuristics"...can be found in the book.

Substitution: Example #1

Recall that substitution was made using parse trees. Substitution can be applied directly without using a parse tree. Let us consider the following Boolean formulas.

$$
\begin{aligned}
\varphi &= (X_1 + X_2)\,, \\
\alpha_1 &= (X \cdot 0)\,, \\
\alpha_2 &= (\neg Y)\,.
\end{aligned}
$$

A substitution of $\alpha_1, \alpha_2 \in \mathcal{BF}(\{X, Y\}, \{\cdot, \neg\})$ in $\varphi \in \mathcal{BF}(\{X_1, X_2\}, \{+\})$ yields the Boolean formula $\varphi(\alpha_1, \alpha_2) \in \mathcal{BF}(\{X, Y\}, \{\cdot, +, \neg\})$.

Substitution: Example #1 (cont.)

We substitute α_1 for X_1, and α_2 for X_2, as follows. We apply a simple "find and replace" procedure, i.e., we replace every symbol X_1 in the string φ with the string $(X \cdot 0)$, and every symbol X_2 in the string φ with the string $(\neg Y)$, as follows:

1. The original formula: $\varphi = (X_1 + X_2)$.
2. Replacing X_1 with α_1 results with the formula: $((X \cdot 0) + X_2)$.
3. Replacing X_2 with α_2 results with the formula: $((X \cdot 0) + (\neg Y))$.

Substitution: Example #2

Let us consider the following Boolean formula.

$$\alpha \quad = \quad (X_1 \cdot X_2)\,.$$

A substitution of $\alpha \in \mathcal{BF}(\{X_1, X_2\}, \{\cdot\})$ in $\alpha \in \mathcal{BF}(\{X_1, X_2\}, \{\cdot\})$ yields the Boolean formula $\alpha(\alpha, X_2) \in \mathcal{BF}(\{X_1, X_2\}, \{\cdot\})$.

Substitution: Example #2 (cont.)

We substitute α for X_1 , as follows:

1. The original formula: $\alpha = (X_1 \cdot X_2)$.
2. Replacing X_1 with α results with the formula: $((X_1 \cdot X_2) \cdot X_2)$.

Note that we replaced X_1 with α ONCE and not RECURSIVELY forever...

Substituting again...

1. Let $\psi(X_1, X_2)$ denote $\varphi(\alpha, X_2)$.
2. A substitution of α in $\psi(X_1, X_2) \in \mathcal{BF}(\{X_1, X_2\}, \{\cdot\})$ yields the Boolean formula $\psi(\alpha, X_2) \in \mathcal{BF}(\{X_1, X_2\}, \{\cdot\})$, as follows:
3. Replacing X_1 with α results with the formula: $(((X_1 \cdot X_2) \cdot X_2) \cdot X_2)$.

Tautologies

Prove that the following formulas are tautologies: (i) addition: $\varphi_1 \triangleq (X \to (X + Y))$, and (ii) simplification: $\varphi_2 \triangleq ((X \cdot Y) \to X)$.

Proof.

The proof is by truth tables, The following figure depicts the tables of both formulas. Note that the row that represents $\hat{\tau}_v(\varphi_i)$ is a constant Boolean function, i.e., $\forall\ v \in \{0,1\}^2 : \hat{\tau}_v(\varphi_i) = 1$. Example 1, on page 85 implies that φ_i are tautologies, as required.

X	Y	$X + Y$	φ_1
0	0	0	1
1	0	1	1
0	1	1	1
1	1	1	1

X	Y	$X \cdot Y$	φ_2
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	1

Table: The truth tables of the addition and the simplification tautologies.

$\square$

Complete Set of Connectives

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

Recall the following definitions.

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 $\mathcal{BF}(\{X_1,\ldots,X_n\},\mathcal{C})$.

Complete Set of Connectives (cont.)

Theorem

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

Proof.

On the whiteboard. $\square$

$\mathcal{C} = \{\text{AND}, \text{OR}\}$ is not Complete Set of Connectives

Theorem

The set $\mathcal{C} = \{\text{AND}, \text{OR}\}$ is not a complete set of connectives.

Proof.

- We prove that the Boolean function NOT is not expressible by $\mathcal{BF}(\{X_1\}, \mathcal{C})$.
- How? we prove that every $p \in \mathcal{BF}(\{X_1\}, \mathcal{C})$, B_p is either the function 0, 1, or I, where I is the identity function.
- Proof by induction on the size of the parse tree of the Boolean formula p (on the whiteboard).
- Since NOT is not the function 0, 1, or I, it follows that NOT is not expressible by $\mathcal{BF}(\{X_1\}, \mathcal{C})$, as required.

$\square$