CS-472: Design Technologies for Integrated Systems

Exercise Problem Set 2 Solution — Date: 26/09/2024

Problem 1

Given the following function: $f(a,b,c) = a\bar{b}c + bc + \bar{a}b\bar{c}$

(a) Write the truth table.

Ans:

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

(b) Write the function in terms of its minterms.

Ans: $f(a,b,c) = \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}c + abc$

(c) Write the function using only NAND-2 gates.

Ans: $f(a,b,c) = ((a \bar{\wedge} a) \bar{\wedge} b) \bar{\wedge} (a \bar{\wedge} c)$

Problem 2

A truth table is a complete listing of all points in a Boolean input space and the corresponding output values. For example, the two-input AND function has the following truth table:

x_1	x_0	$x_0 x_1$
0	0	0
0	1	0
1	0	0
1	1	1

To compactly represent a truth table, we can use a bitstring $b_{2^n-1} \dots b_1 b_0$ where $b_x = f(w_1, \dots, w_n)$ when $w = (x_n \dots x_1)_2$. Each bit in the bitstring corresponds to the output of f evaluated at some assignment to its variables. The most-significant bit b_{2^n-1} corresponds to all variables assigned to 1, $f(1,\dots,1)$ and the least-significant bit b_0 correspond to $f(0,\dots,0)$. Using this compact representation, the two-input AND function is 1000.

Table 1: The sixteen logical operations on two variables, $f(x, y)$.

Truth table	Notation(s)	Name
0000	$\perp$	Contradiction; antilogy; constant 0
0001	$x \barvee y$, $x \downarrow y$	Nondisjunction; NOR
0010		Converse nonimplication
0011	x', $\neg x$, $\bar{x}$	Left complementation
0100		Nonimplication
0101	y', $\neg y$, $\bar{y}$	Right complementation
0110	$x \oplus y$	Exclusive disjunction; nonequivalence; XOR
0111	$x \barwedge y$, $\uparrow y$	Nonconjuction; NAND
1000	$x \cdot y$, xy, $x \wedge y$	Conjunction; AND
1001	$x \equiv y$, $x \Leftrightarrow y$	Equivalence
1010	y	Right projection
1011	$x \Rightarrow y$	Implication
1100	x	Left projection
1101	$x \Leftarrow y$	Converse implication
1110	$x + y$, $x \vee y$	Disjunction; OR
1111	$\top$	Affirmation; tautology; constant 1

(a) Suppose that on a remote country logicians use the symbol 1 for "false" and 0 for "true". How do our logical operations associate to theirs?

For example, they have a binary operation "or"

$$1 \text{ or } 1 = 1, \quad 1 \text{ or } 0 = 0, \quad 0 \text{ or } 1 = 0, \quad 0 \text{ or } 0 = 0$$

which we associate with AND (1000).

Ans: The *dual* of a Boolean function is the expression that can be obtained by interchanging AND and OR operations and interchanging 0's and 1's. The dual function of $f = x \circ y$ is denoted by $f^d = \overline{\bar{x} \circ \bar{y}}$ and therefore one truth table is the reverse of the complement of the other. Thus, the two countries associate by duality.

For instance, the dual function of $f = a\bar{b} + c$ is $f^d = (a + \bar{b})c$.

As an interesting case, some functions are self-dual. This means that $f = f^d$. For instance the majority operation $MAJ(a, b, c) = ab + ac + bc$ is self-dual.

(b) All operations in Table 1 can be expressed in terms of NAND. For each of the 16 operations in the table find a formula equivalent to the function that uses only two-input NAND. The formula should be as short as possible and not contain any constant.

Ans:

4. [*Trans. Amer. Math. Soc.* **14** (1913), 481–488.] (a) Start with the truth tables for L and R; then compute truth table $\alpha \bar{\wedge} \beta$ bitwise from each known pair of truth tables α and β, generating the results in order of the length of each formula and writing down a shortest formula that leads to each new 4-bit table:

$\perp$: $(x \bar{\wedge} (x \bar{\wedge} x)) \bar{\wedge} (x \bar{\wedge} (x \bar{\wedge} x))$	$\bar{\vee}$: $(x \bar{\wedge} (x \bar{\wedge} x)) \bar{\wedge} ((y \bar{\wedge} y) \bar{\wedge} (x \bar{\wedge} x))$
$\wedge$: $(x \bar{\wedge} y) \bar{\wedge} (x \bar{\wedge} y)$	$\equiv$: $(x \bar{\wedge} y) \bar{\wedge} ((y \bar{\wedge} y) \bar{\wedge} (x \bar{\wedge} x))$
$\bar{\supset}$: $(x \bar{\wedge} (x \bar{\wedge} y)) \bar{\wedge} (x \bar{\wedge} (x \bar{\wedge} y))$	$\bar{\mathrm{R}}$: $y \bar{\wedge} y$
L: x	$\subset$: $y \bar{\wedge} (x \bar{\wedge} x)$
$\bar{\subset}$: $(y \bar{\wedge} (x \bar{\wedge} x)) \bar{\wedge} (y \bar{\wedge} (x \bar{\wedge} x))$	$\bar{\mathrm{L}}$: $x \bar{\wedge} x$
R: y	$\supset$: $x \bar{\wedge} (x \bar{\wedge} y)$
$\oplus$: $(y \bar{\wedge} (x \bar{\wedge} x)) \bar{\wedge} (x \bar{\wedge} (x \bar{\wedge} y))$	$\bar{\wedge}$: $x \bar{\wedge} y$
$\vee$: $(y \bar{\wedge} y) \bar{\wedge} (x \bar{\wedge} x)$	$\top$: $x \bar{\wedge} (x \bar{\wedge} x)$

(c) Similarly, find 16 short formulas when constants are allowed.

Ans:

(b) In this case we start with four tables $\perp$, $\top$, L, R, and we prefer formulas with fewer occurrences of variables whenever there's a choice between formulas of a given length:

$\perp$: 0	$\bar{\vee}$: $1 \bar{\wedge} ((y \bar{\wedge} 1) \bar{\wedge} (x \bar{\wedge} 1))$
$\wedge$: $(x \bar{\wedge} y) \bar{\wedge} 1$	$\equiv$: $(x \bar{\wedge} y) \bar{\wedge} ((y \bar{\wedge} 1) \bar{\wedge} (x \bar{\wedge} 1))$
$\bar{\supset}$: $((y \bar{\wedge} 1) \bar{\wedge} x) \bar{\wedge} 1$	$\bar{\mathrm{R}}$: $y \bar{\wedge} 1$
L: x	$\subset$: $y \bar{\wedge} (x \bar{\wedge} 1)$
$\bar{\subset}$: $(y \bar{\wedge} (x \bar{\wedge} 1)) \bar{\wedge} 1$	$\bar{\mathrm{L}}$: $x \bar{\wedge} 1$
R: y	$\supset$: $(y \bar{\wedge} 1) \bar{\wedge} x$
$\oplus$: $(y \bar{\wedge} (x \bar{\wedge} 1)) \bar{\wedge} ((y \bar{\wedge} 1) \bar{\wedge} x)$	$\bar{\wedge}$: $x \bar{\wedge} y$
$\vee$: $(y \bar{\wedge} 1) \bar{\wedge} (x \bar{\wedge} 1)$	$\top$: 1

(d) For each of the 16 operations in the table try to find a formula equivalent to the function using only the two-input XOR operator ($\oplus$) and without the use of constants 0 or 1. Is it possible? Why?

Ans: It is not possible since XOR is not functionally complete.

(e) Consider the previous exercise using the material implication ($\Rightarrow$) as basis operator instead of $\oplus$.

Ans: Only some solutions are possible:

(a) $\perp$: $x \bar{\subset} x$; $\wedge$: $(x \bar{\subset} y) \bar{\subset} y$; $\bar{\supset}$: $y \bar{\subset} x$; L: x; $\bar{\subset}$: $x \bar{\subset} y$; R: y;

(f) If the use of constants is allowed, how would it affect the answer for (e)?

Ans:

$\perp$: 0

$\wedge$: $(y \,\bar{\subset}\, 1) \,\bar{\subset}\, x$

$\bar{\supset}$: $y \,\bar{\subset}\, x$

L: x

$\bar{\subset}$: $x \,\bar{\subset}\, y$

R: y

$\oplus$: $((y \,\bar{\subset}\, x) \subset ((x \,\bar{\subset}\, y) \,\bar{\subset}\, 1)) \,\bar{\subset}\, 1$

$\vee$: $(y \subset (x \,\bar{\subset}\, 1)) \,\bar{\subset}\, 1$

$\bar{\vee}$: $y \subset (x \,\bar{\subset}\, 1)$

$\equiv$: $(y \,\bar{\subset}\, x) \subset ((x \,\bar{\subset}\, y) \,\bar{\subset}\, 1)$

$\bar{\mathsf{R}}$: $y \,\bar{\subset}\, 1$

$\subset$: $(x \,\bar{\subset}\, y) \,\bar{\subset}\, 1$

$\bar{\mathsf{L}}$: $x \,\bar{\subset}\, 1$

$\supset$: $(y \,\bar{\subset}\, x) \,\bar{\subset}\, 1$

$\bar{\wedge}$: $((y \,\bar{\subset}\, 1) \,\bar{\subset}\, x) \,\bar{\subset}\, 1$

$\top$: 1

(Solutions in exercises 2.b, 2.c, 2.e, and 2.f are taken from the book *The Art of Computer Programming, Donald Knuth*, thus some of the symbols are different.)