Lecture 10 In-Class Worksheet

1 Learning Objectives

This worksheet is based on Lumetta course notes section 2.1.2–2.1.4. After completing this lesson, you will know how to:

- [S10-1] Determine whether an expression is a minterm or a maxterm.
- [S10-2] Determine whether an expression is an implicant or a prime implicant.
- [S10-3] Express a function as a sum of products (SOP) or product of sums (POS).
- [S10-4] Find the minimal SOP and POS form of Boolean function.

For additional practice with Karnaugh maps:

`https://lumetta.web.engr.illinois.edu/120-kmap/120-kmap.html`

2 Standard Forms of Boolean Expressions

A *literal* is either a variable or its negation. A *minterm* of a Boolean function is a product of literals (one or more literals **AND**'ed together) where each function variable occurs either as itself or as its negation. For example, $A\bar{B}C$ is a minterm of a function on three variables A, B, and C, while $A\bar{C}$ is not a minterm of the same function, because the variables B is not present in the product $A\bar{C}$. Similarly, a *maxterm* of a Boolean function is a sum of literals (one or more literals **OR**'ed together) where each function variable occurs either as itself or as its negation.

We can think of a minterm as a function in its own right. We know that this function is 1 at exactly one assignment of values to its variables. For example, the minterm $A\bar{B}C$ defines function on the variables A, B, and C. To have $A\bar{B}C = 1$, all of the literals must evaluate to 1. That is, $A = 1$, $\bar{B} = 1$, and $C = 1$. This happens only when $A = 1$, $B = 0$, and $C = 1$. At all other possible assignments of values, at least one of the literals evaluates to 0, and the product is 0.

Similarly, a maxterm evaluates to 0 at exactly one assignment of values to the variables in the maxterm; for all other possible assignments, it has the value 1. For example, the maxterm $A + \bar{B} + C$ evaluates to 0 only when $A = 0$, $\bar{B} = 0$, and $C = 0$. This happens only when $A = 0$, $B = 1$, $C = 0$. At all other assignments of values to A, B, and C, at least one of the literals evaluates to 1, and the entire sum evaluates to 1.

Q1. Find an assignment to the variables A, B, C, and D at which the expression $AB\bar{C}\bar{D}$ evaluates to 1.

.

Q2. Construct a Boolean expression on four variables (A, B, C, and D) that evaluates to 1 only when $A = 0$, $B = 1$, $C = 0$, and $D = 0$.

.

Q3. Find an assignment to the variables A, B, C, and D at which $\bar{A}+B+\bar{C}+D$ evaluates to 0.

.

Q4. Construct a Boolean expression on four variables (A, B, C, and D) that evaluates to 0 only when $A = 0$, $B = 1$, $C = 0$, and $D = 0$.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Given a truth table for an arbitrary Boolean function, we can construct a Boolean algebra expression that has the same truth table. We do this by summing (**OR**'ing) all minterms of the function where the function is 1. The resulting expression, called a sum of products (SOP), will evaluate to 1 at precisely those assignments of values to the function variables where the function is 1. For example, consider the function $F(A,B)$ given by the truth table on the right. It has value 1 when $A=0$ and $B=1$, and when $A=1$ and $B=0$. The minterms corresponding to those variable assignments are $\bar{A}B$ and $A\bar{B}$, because when $A=0$ and $B=1$, the minterm $\bar{A}B$ is 1. Similarly, when $A=1$ and $B=0$, the minterm $A\bar{B}$ is 1. Combining these two minterms using **OR**, we have

$$F_{\text{SOP}} = \bar{A}B + A\bar{B}.$$

We can confirm that $F_{\text{SOP}} = F$ given by the truth table by evaluating it at all possible assignments to A and B. For example, at $A=0$ and $B=0$:

$$\begin{aligned} F_{\text{SOP}}(0,0) &= \bar{0}\cdot 0 + 0\cdot\bar{0} \\ &= 1\cdot 0 + 0\cdot 1 \\ &= 0+0 \\ &= 0. \end{aligned}$$

We can also express F as a product of sums (POS) by multiplying (**AND**'ing) all of the maxterms that correspond to the rows of the truth table where $F=0$. Specifically, $F=0$ when $A=0$ and $B=0$, and when $A=1$ and $B=1$. The corresponding maxterms are $A+B$ and $\bar{A}+\bar{B}$. Multiplying these maxterms gives

$$F_{\text{POS}} = (A+B)(\bar{A}+\bar{B}).$$

Q5. Give a SOP and POS expression for the function G given by the following truth table:

A	B	C	G
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

The SOP expression constructed using minterms and the POS expression constructed using maxterms are said to be in *canonical form*. Given two functions in SOP or POS canonical form, we can easily determine if the functions are the same, provided we impose a consistent ordering on the variables and minterms.

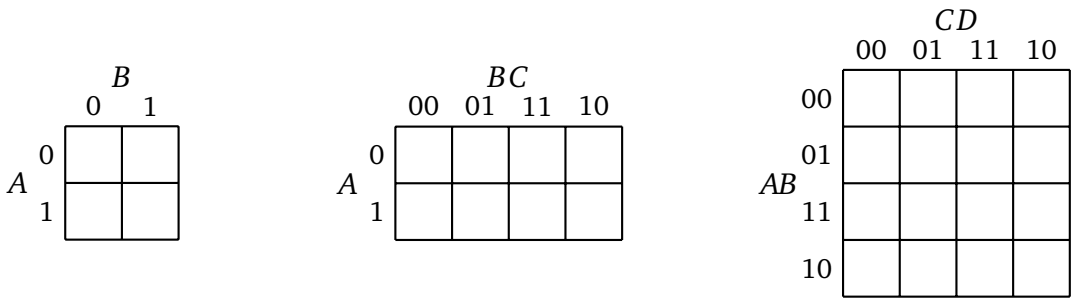
3 Optimizing Logic Expressions

A Boolean function G is said to be an *implicant* of another Boolean function F if:

- G is a function on the same set of variables as F,
- G is product of literals, and
- Whenever G is 1, F is also 1.

A *prime implicant* is an implicant from which no literals can be dropped from the product without the expression no longer satisfying the third property above (whenever the implicant is 1, the original function is also 1). A function is said to be in *minimal SOP form* when it is a sum of the fewest possible prime implicants.

A *Karnaugh map* (or *K-map*) is a way of writing the truth table of a function in a way that makes it easy to identify its prime implicants. The figures below show the Karnaugh map templates for functions of two (left), three (center), and four (right) Boolean variables. Note the unusual labeling of the columns in the larger Karnaugh maps: 00, 01, 11, 10.



To find a minimal SOP expression for a function using a Karnaugh map, **we enclose all groups of ones using rectangular loops.** The rules for these loops are:

- **1.** Each loop must be rectangular.
- **2.** Each loop must enclose exactly 1, 2, 4, 8, or 16 *ones*.
- **3a.** In a three-variable K-map, loops may wrap around left to right.
- **3b.** In a four-variable K-map, loops may wrap around top to bottom and left to right.
- **4.** Each loop must be as large as possible (subject to the above).
- **5.** Each *one* must be inside some loop.

Any loop that satisfies conditions 1–3 represents an implicant of the function. If the loop also satisfies condition 4, it is a prime implicant. To find the Boolean expression for the implicant, we read off the set of variables which are constant in the area enclosed by the loop. We then choose the smallest set of loops that enclose all ones in the Karnaugh map and write the function as the sum of the corresponding prime implicants.

Q6. Express the function H, given by the truth table below, in minimal SOP form.

A	B	C	H
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

We can also use Karnaugh maps to express a function in *minimal POS form*. The process is the same, except:

- Instead of grouping ones, we group zeroes.
- In a POS term, the literals are the complement of what they would be in an SOP term.

Q7. Express the function U in minimal SOP form.

A	B	C	D	U	$\bar{U}$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Q8. Express the function U in minimal POS forms.

A	B	C	D	U	$\bar{U}$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Q9. Express the functions $\bar{U}$ in minimal SOP form.

A	B	C	D	U	$\bar{U}$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Q10. Express the function $\bar{U}$ in minimal POS forms.

A	B	C	D	U	$\bar{U}$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1