// Canonical Sum of Minterms Primer

Background::

It is imperative that the technique described here is not embraced as a substitute for understanding the fundamental Boolean Algebra that supports it. While it's fun and exciting to find shortcuts and workarounds, we can all agree that there's no replacement for the satisfaction of knowing that there is a fundamental reason why the shortcut works. Before you commit to learning this technique, be sure that are comfortable with the Boolean Algebra Laws. In particular, review the Complementation, Identity, and Distributive Laws in detail. These three laws are the backbone of converting Boolean equations into canonical sum of minterms form.

As a review of the lecture material, I will walk through the Boolean Algebra approach to deriving the canonical sum of minterms form of a sample Boolean equation. After you are comfortable with this process, I encourage you to follow along as I demonstrate a methodology in finding the sum of minterms form that will certainly help you when it's crunch time during an exam or while working on assignments.

Purpose::

Provide an alternative methodology to deriving the canonical sum of minterms form from a Boolean equation. Everyone will respond to this technique differently and certainly some of you will find the Boolean Algebra approach more effective or more straight forward. Regardless, this exercise will introduce an approach that may even reveal a relationship between the truth table representation of Boolean equations and the equation itself.

Boolean Algebra Derivation::

The canonical sum of minterms form of an equation can be derived by successively applying the Complementation and Identity rules followed by a distribution event via the Distribution Law. The final step (leveraging the Idempotence Law) is to remove duplicate minterms. Note: The original equation must be in sum of products form to apply this succession of steps.

F(a,b,c) = ab' + a' + a'b'c + bc + c'

ab'**(c + c')** + a'**(b + b')** + a'b'c + bc**(a + a')** + c'**(a + a')**

ab'**c** + ab'**c'** + a'**b** + a'**b'** + a'b'c + **a**bc + **a'**bc + **a**c' + **a'**c'

ab'c + ab'c' + a'b**(c + c')** + a'b'**(c + c')** + a'b'c + abc + a'bc + ac'**(b + b')** + a'c' **(b + b')**

ab'c + ab'c' + a'b**c** + a'b**c'** + a'b'**c** + a'b'**c'** + a'b'c + abc + a'bc + a**b**c' + a**b'**c' + a'**b**c' + a'**b'**c'

ab'c + ab'c' + a'bc + a'bc' + a'b'c + a'b'c' + ~~**a'b'c**~~ + abc + ~~**a'bc**~~ + abc' + ~~**ab'c'**~~ + ~~**a'bc'**~~ + ~~**a'b'c'**~~

F(a,b,c) = ab'c + ab'c' + a'bc + a'bc' + a'b'c + a'b'c' + abc + abc'

Alternative Method::

With a firm understanding of how the Laws of Boolean Algebra are leveraged to derive the sum of minterms form, you are in a position to examine the technique. Note: In order to use this method, the equation must be in sum of products form.

Let’s take the simple equation F(a,b,c) = a’ as an example. This equation has a single term, so we need only consider this one term but for an equation with multiple terms, we would treat them all individually.

Step 1:

Identify the variables that do not appear in the term. Consider only the variables, not which literals appear. Here, we see that the variables *b* and *c* do not appear in the term.

Step 2:

Form a list of all possible permutations of these missing variables while considering their true or complemented form (i.e. all the possible *literals*). This is the crucial step of the technique. It is effective to think of the literals as 1’s and 0’s and generate the list as if you are creating a truth table. The following is the permutation list alongside a “binary representation” for clarity:

b'c' (00)

b'c (01)

bc' (10)

bc (11)

Step 3:

Simply add the original term (regardless of how many variables are present within it) to each one of these permutations. This generates all of the individual minterms needed to represent the original term. See below:

a'b'c'

a'b'c

a'bc'

a'bc

Note: The representation (true or complemented form) of the variable in the original term is unchanged. This will be true regardless of the number of variables in the original term.

Step 4:

Combine all the minterms together with OR gates. See below:

F(a,b,c) = a'b'c' + a'b'c + a'bc' + a'bc

[complete Steps 5 – 6 if the original equation has more than a single term]

Step 5:

Examine all of the lists of minterms for duplicates and remove them.

Step 6:

Combine all minterm lists together with OR gates.

Worked Example::

What was demonstrated was a simplified version of the technique. Below, you will find a more complex example in a format that may be indicative of how you will perform the technique in practice. The Boolean equation used is the same as the equation used to demonstrate the algebraic method, F(a,b,c) = ab' + a' + a'b'c + bc + c'.

ab'		a'		a'b'c'		bc		c'
c'	+	b'c'	+		+	a'	+	a'b'
c		b'c				a		a'b
		bc'						ab'
		bc						ab

Treating each term separately, we generate the list of permutations of the missing variables (considering their true and complemented form).

ab'		a'		a'b'c'		bc		c'
ab'c'	+	a'b'c'	+	a'b'c'	+	a'bc	+	a'b'c'
ab'c		a'b'c				abc		a'bc'
		a'bc'						ab'c'
		a'bc						abc'

We combine each permutation with the original term to generate an exhaustive list of minterms. Hint: Don’t forget original terms that are already minterms!

ab'		a'		a'b'c'		bc		c'
ab'c'	+	a'b'c'	+	~~a'b'c'~~	+	~~a'bc~~	+	~~a'b'c'~~
ab'c		a'b'c				abc		~~a'bc'~~
		a'bc'						~~ab'c'~~
		a'bc						abc'

We examine the list and remove the duplicates. Combining all of these minterms with OR gates gives us the canonical sum of minterms.

F(a,b,c) = ab'c' + ab'c + a'b'c' + a'b'c + a'bc' + a'bc + abc + abc'