Truth Table, ANF and Trace Definitions and Input Examples

Truth Table

The truth table for an n-variable Boolean function 'f' should be in lexicographical form, i.e., $f = (f(0), f(1), f(2), \ldots, f(2^{n-1}))$. Because the truth table length might be big we represent it in hexadecimal rather than binary bits. To represent a truth table in hexadecimal, starting from the first bit we replace each four bits by their corresponding hexadecimal.For example,if $f = (0,0,1,1,1,1,1,1)$, then to represent f, we take the first four bits (0011) and replace them by 3 and take the second four bits (1111) and replace them by F. Therefore, our hexadecimal representation for f = (0,0,1,1,1,1,1,1) is 3F. **So to enter (0,0,1,1,1,1,1,1), you should just write 3F**.

ANF

ANF, is an abbreviation for Algebraic Normal Form. The ANF of an n-variable Boolean function is in the following form: $f(x_0, x_1, \ldots, x_{n-1}) = c_0x_0 + c_1x_1 + \ldots + c_{0,1}x_0x_1 + \ldots + c_{0,n-1}x_0x_1 + \ldots + c_{0,1,\ldots,n-1}x_0x_1 \cdots x_{n-1}$ where $c_0, c_1, \ldots, c_{0,1}, \ldots, c_{0,n-1}, \ldots, c_{0,1,\ldots,n-1}$ are binary coefficients and $x_0, x_1, \ldots, x_{n-1}$ are the Boolean variables. To enter an ANF of a Boolean function, we make a label for each variable. The variables x_i, where $0 \leq i \leq 9$, are labeled by i. The variables $x_{10}, x_{11}, \ldots, x_{36}$, are labeled by the letters $a, b, c, d, e, \ldots, z$. In fact our website is able to cope for Boolean functions with variables up to 21. **For example, to enter** $f(x_0, x_1, x_2, \ldots, x_{10}, x_{11}) = x_0x_{11} + x_1 + x_0x_2 + x_8x_{10}$**, you should just write** $0b, 1, 02, 8a$**. Another example, to enter** $f(x_0, x_1, x_2) = x_0x_1 + x_1 + x_2 + 1$**, you should write either *,01,1,2 or ,,01,1,2**.

Trace

In the theory of finite fields, the trace function on the finite field F_{p^n} is the function $Tr : F_{p^n} \rightarrow F_p$ defined by $Tr(c) = c + c^p + c^{p^2} + c^{p^3} + \ldots + c^{p^{(n-1)}}$. Here we are considering the case when $p = 2$, that is, when our finite field is the binary field F_{2^n}. So our trace is a function $Tr : F_{2^n} \rightarrow F_2$. Define the function $Tr(x^{at+b})$ on F_{2^n} for $0 \leq t \leq 2^{n-2}$ and integers a, b. Let $p(x)$ be a primitive polynomial over F_{2^n}, then x can generate F_{2^n}, i.e., x^t where

$0 \leq t \leq 2^n - 2$ are all nonzero the elements of F_{2^n}. From the theory of finite fields, we know that each element in F_{2^n} can be represented by a binary string of length n, and we also know that F_{2^n} consists of all the possible binary strings of length n. This means that for each value of x^t, we have a corresponding binary string. By evaluating $Tr(x^{at+b})$ for $0 \leq t \leq 2^{n-2}$, we obtain $2^n - 1$ binary values. Now, for each t, let $Tr(x^{at+b})$ be an element in the truth table at the position corresponding to the decimal representation of the binary string corresponding to the element x^t. Now, if we set a value at position 0, we will have a complete truth table. This value can be either true or false, but by convention we set it as false.The general form of the trace function we are dealing with, is $Tr(x^{a_1t+b_1}) + Tr(x^{a_2t+b_2}) + Tr(x^{a_3t+b_3}) + \ldots + Tr(x^{a_kt+b_k})$, where $a_1, a_2, a_3, \ldots, a_k$ are different integers. We enter it in the following form $a_1, b_1/a_2, b_2/a_3, b_3/\ldots/a_k, b_k$. There is a restriction on the values of $b_1, b_2, b_3, \ldots, b_k$ depending on $a_1, a_2, a_3, \ldots, a_k$ respectively. In the trace calculations, you need to enter a primitive polynomial $p(x)$. We have a simple way to enter it, by typing the number of variables in the primitive polynomials textbox, n, you will immediately see a list containing all the primitive polynomials of degree n from which you can select your primitive polynomial. **For example, to enter $Tr(x^{3t+2}) + Tr(x^{2t})$, you should just write 3,2/2,0**.