Problem Set #9 – Solutions

Problem 1. Finite Functions on a Computer

We consider functions $F : \{0,1\}^n \to Y$, where $|\{0,1\}^n| = 2^n$.

1. Output {0,1}. For each of the 2^n inputs, the function can independently output either 0 or 1. The number of possible functions is $2^{(2^n)}$.

2. Output {-1,0,1}. For each of the 2^n inputs, there are 3 possible outputs. The number of possible functions is $3^{(2^n)}$.

3. Output $\{0,1\}^m$. Each output is an m-bit vector. There are 2^m possible outputs for each input. The number of possible functions is $2^{(m \cdot 2^n)}$.

Decision tree interpretation. A decision tree for inputs of size n has 2^n leaves. Each leaf can be labeled by an output value. Counting functions corresponds to counting all possible labelings.

Problem 2. NAND implies AND, OR, NOT

The NAND operation is defined as $A \uparrow B$ = not (A and B). NOT can be implemented as not A = $A \uparrow A$. AND can be implemented as $(A \uparrow B) \uparrow (A \uparrow B)$. OR can be implemented as $(A \uparrow A) \uparrow (B \uparrow B)$.

Since AND, OR, and NOT can be built from NAND, NAND is universal. Each gate can be replaced by at most 3 NAND gates. Thus, a circuit of size n can be simulated using at most 3n NAND gates.

Problem 3. Universality of Boolean Circuits

Let $F : \{0,1\}^n \to \{0,1\}$. For each input x in $\{0,1\}^n$, define $\delta_x(y) = 1$ if $y = x$ and 0 otherwise. Each δ_x can be implemented using $O(n)$ gates.

The function F can be written as an OR over all x such that $F(x) = 1$ of $\delta_x(y)$. There are at most 2^n such terms. Therefore, F can be computed using $O(n \cdot 2^n)$ gates.