Week 4: Eval

Authors: Benjamin (aqn9yv, dlb2ru, ht6xd, iad4de, jmn4fms, lw7jz)

Problem 1 *Finite vs. Infinite functions (briefly on section 3)*

The cardinality of the set of all infinite functions with binary input is uncountably infinite. This cardinality can be proven by using Cantor's diagonalization argument. In order to use Cantors's diagonalization, the functions must be labeled or enumerated in a way that they can be mapped to real numbers.

Imagine each infinite function can be represented as an infinite binary string of possible inputs followed by their corresponding output(s). Begin enumerating these strings. Then Cantor's diagonal argument can be used to show that when you change the n^{th} value in the n^{th} string, a representation of a new function is produced that could not have been included in the enumeration, that the n^{th} value of that function is different than the value in the same location of the n^{th} function. This shows how the infinite functions cannot be counted.

Problem 6 *Equal to Constant Function (TCS exercise 5.3 and* Defining EVAL *video)*

Definition 1 $EQUALS_{x'} : \{0,1\}^k \rightarrow \{0,1\}$ *where* $EQUALS_{x'}(x) = 1$ *if and only if* $x = x'$.

Proof. We will proceed by induction on the number of inputs k of the function $EQUALS_{x'}$

Base Case For k $\in \mathbb{N}^+$ let $k = 1$. For two single-bit numbers $a, b \in \{0,1\}$, $a = b \iff$ XNOR(a,b) = 1. XNOR requires 5 NAND gates, so for $k = 1$ we have a NAND-CIRC straightline program of no more than $c \cdot k$ lines when $c = 5$.

Inductive Hypothesis Assume there's a constant c such that $c \cdot k$ is an upper bound on the number of gates required to compute $EQUALS_{x'}\{0,1\}^k$.

Consider $EQUALS_{x'}\{0,1\}^{k+1}$.

If x_k and x'_k are numbers with k bits, in order to compute $EQUALS_{x'_{k+1}}(x_{k+1})$ where x'_{k+1} and x_{k+1} have one additional bit we must:

1. compute $EQUALS_{x'_k}(x_k)$
2. compare the additional $(k+1)^{th}$ bits using a single XNOR-gate
3. combine parts 1. and 2. to check the overall equality using an additional AND-gate.

Thus, $EQUALS_{x'_{k+1}}(x_{k+1})$ requires the number of NAND-gates from $EQUALS_{x'_k}(x_k)$, 5 NAND-gates for XNOR, and 3 NAND-gates for AND.

From our assumption we know $EQUALS_{x'_k}(x_k)$ is bounded above by $c \cdot k$ NAND-gates, so $EQUALS_{x'_{k+1}}(x_{k+1})$ will be bounded above by $c \cdot k$ + 8.

Let's say this number of additional lines is less than or equal to the constant c, i.e. $c \geq 8$. Then the number of lines to compute $EQUALS_{x'_{k+1}}(x_{k+1})$ is at most $c \cdot k + c = c(k+1)$.

Conclusion We know that for $k = 1$, $\exists$ a constant c such that a NAND-CIRC program which computes $EQUALS_{x'}(x)$ requires no more than $c \cdot k$ lines. We also know that a NAND-CIRC program P which $EQUALS_{x'_k}(x_k)$ is bounded by at most $c \cdot k$ lines implies a program P' which computes $EQUALS_{x'_{k+1}}(x_{k+1})$ is bounded by $c(k+1)$ lines, for $c \geq 8$. So by the principle of induction, $\forall$ NAND-CIRC programs P which compute $EQUALS_{x'}(x)$, $\exists$ a constant c such that the number of lines in P is no more than $c \cdot k$ $\forall k \in \mathbb{N}^+$.

$\square$