Exercise 3: SOLUTIONS

Honor Notice: SOLUTION SET, for use by CS3102 at UVA students in Spring 2020 only. Anyone reading this outside of that course for that semester is in violation of the honor code. If you view these solutions, you may not (re) submit the corresponding exercises

Exercise 3-2: Equal to Constant Function (TCS Exercise 5.3)

For every $k \in \mathbb{N}$ and $x' \in \{0,1\}^k$, show that there is an $O(k)$ line *NAND-CIRC* program that computes the function $EQUALS_{x'} : \{0,1\}^k \to \{0,1\}$ that on input $x \in \{0,1\}^k$ outputs 1 if and only if $x = x'$.

We can implement $EQUALS_{x'}$ in the following way:

$$EQUALS_{x'}(x_0, ..., x_{k-1}) = \begin{cases} 0 & x'_0 == x_0 \\ EQUALS_{x'_1 \dots x'_k}(x_1, ..., x_{k-1}) & otherwise \end{cases}$$

where $EQUALS_0(x) = NAND(x, 0)$ and $EQUALS_1(x) = NOT(NAND(x, 0))$

We could write a program for $EQUALS_{x'}$ as follows:

```
def EQUALSxprime(x0, ..., xkminus1):
    notequal0 = XOR(x0, xprime0)
    restequal = EQUALxprimerest(x1, ..., xkminus1)
    return IF(notequal0, 0, restequal)
```

Where `xprimerest` refers to the string consisting of all but the first bit of `xprime`, and `xprime0` refers to the first bit of `xprime`.

We can show that the number of gates required for $EQUALS_{x'}$ is $O(k)$ by showing that there is some constants n_0, c such that $\forall n > n_0$ we have that the number of gates required to do $EQUALS_{x'}$ where x' is n bits long is less than or equal to $c \cdot n$. We will use $n_0 = 1, c = 20$ We will show this by induction.

Base Case: Our base cases for $n = 1$ (we have two) are $EQUALS_0(x)$ and $EQUALS_1(x)$. Each of those require fewer than 20 gates by inspection.

Inductive Hypothesis: Assume there is some constant c such that $EQUALS_{x'}$ can be implemented in no more that $20 \times k$ gates, where the length of x' is k.

Inductive Step: We will use the inductive hypothesis above to show that we can implement $EQUALS_{x'}$ within $20 \cdot (k+1)$ gates provided that the length of x' is $k+1$

In the code above, we can see that to compute $EQUALS_{x'}$ we need to invoke XOR, IF, and $EQUAL_{rest}$. XOR requires fewer than 10 gates, IF requires fewer than 10 gates, and since the string $rest$ has length k our inductive hypothesis says this requires at most $20k$ gates. This means that $EQUALS_{x'}$ requires no more than $10 + 10 + 20k = 20(k+1)$ gates.

Exercise 3-4: Asymptotic Operators

For each sub-problem, indicate if the statement is *true* or *false* and support your answer with a convincing argument.

(a) $17n \in O(723n + \log n)$

(b) $\min(n^n, 3012) \in O(1)$

(c) $n^2 \in \Theta(n^3)$

(d) $2.0001^n \in O(2^n)$

(e) $log_n 10 \in \Theta(log_{2n} 17)$

a) True. Let $n_0 = 1$ and $c = 1$. We must show that $\forall n > n_0$ we have that $17n \leq c \cdot (723n + \log n)$. First, to show that $17n \leq 723n + \log n$, it suffices to show $17n \leq 723n$, which is true by inspection

b) True Let $n_0 = 5$ and $c = 3102$. We must show that $\forall n > n_0$ we have that $\min(n^n, 3012) \leq 3102 \cdot 1$. First, note that for $n > n_0$ we have that $n^n > 3012$, meaning $\min(n^n, 3012) = 3012$, which is less than 3102.

c) False because $n^2 \notin \Omega(n^3)$. To show this we need to show that for any choice of c, we can find a large enough n_0 such that $n_0^2 < c \cdot n_0^3$. Consider an arbitrary choice of c. whenever $\frac{1}{n_0} < c$ we have that $n_0^2 < c \cdot n_0^3$, and so $n^2 \notin \Omega(n^3)$.

d) False. We will show this by demonstrating that for any choice of c, we can find a large enough n_0 such that $2.0001^n < c \cdot 2^n$. To show this, it suffices to show that for large enough n we have that $\frac{2.0001^n}{2^n} < c$. Note that $\frac{2.0001^n}{2^n} = (\frac{2.0001}{2})^n$ and since $\frac{2.0001}{2} > 1$, raising it to a high enough power will eventually allow it to exceed any constant.

e) True. To begin, we will eliminate the n terms from the bases of the logs, because that's difficult to reason about. To do this, we will use the change of base formula, which states that $\log_b a = \frac{\log_x a}{\log_x b}$ We will use base 10 for our logs:

$$\log_n 10 = \frac{\log_{10} 10}{\log_{10} n} = \frac{1}{\log_{10} n}$$
$$\log_{2n} 17 = \frac{\log_{10} 17}{\log_{10} 2n}$$

To show $log_n 10 \in \Theta(log_{2n} 17)$ we must show that $log_n 10 \in O(log_{2n} 17)$ and $log_n 10 \in \Omega(log_{2n} 17)$.

Big-Omega to show $log_n 10 \in O(log_{2n} 17)$ we will show that there is some c such that for large enough n such that:

$$\begin{aligned}\frac{1}{\log_{10} n} &\geq c \cdot \frac{\log_{10} 17}{\log_{10} 2n}\\ \log_{10} 2n &\geq c \cdot \log_{10} 17 \cdot \log_{10} n\\ \log_{10} 2n &\geq c \cdot 2 \cdot \log_{10} n\\ \log_{10} 2n &\geq \log_{10} n^{2c}\\ 2n &\geq n^{2c}\end{aligned}$$

Which is true for $c = \frac{1}{4}$ and $n > 1$

Big-Oh to show $log_n 10 \in O(log_{2n} 17)$ we will show that there is some c such that for large enough n such that:

$$\begin{aligned}\frac{1}{\log_{10} n} &\leq c \cdot \frac{\log_{10} 17}{\log_{10} 2n}\\ \log_{10} 2n &\leq c \cdot \log_{10} 17 \cdot \log_{10} n\\ \log_{10} 2n &\leq c \cdot \log_{10} n\\ 2n &\leq n^{c}\end{aligned}$$

which is true for $c = 2$ and $n > 2$