Quiz 2: Universality

Collaboration: This quiz is open note but individual. You may use any resources that were provided to you by the course, or that you had recorded as part of your own notes prior to when you first viewed this quiz. You may not discuss this quiz with your cohort-mates, class-mates, tutor, friends, family, or anyone else except the course staff (who will answer clarification questions only).

Problem 1: Comparing Computing Models

Determine whether the gate set $\{\texttt{MAJ}, \texttt{XOR}, 1\}$ is equivalent to $\{\texttt{AND}, \texttt{OR}, \texttt{NOT}\}$. Support your answer with a proof.

Problem 2: Counting Gates

In straightline.py from PS3 you implemented the function $\texttt{COMP}_2 : \{0,1\}^4 \to \{0,1\}$ defined such that $\texttt{COMP}_2(a_1, a_0, b_1, b_0)$ is 1 provided that the natural number represented by the string a_1a_0 is greater than the natural number represented by b_1b_0. Here is an example of a (sugary) way that we could implement $\texttt{COMP}_2$.

```
def COMP2(a1, a0, b1, b0):
    msb_diff = XOR(a1, b1)
    rest_greater = COMP1(a0, b0)
    return IF(msb_diff, a1, rest_greater)
```

We will define the function $\texttt{COMP}_n : \{0,1\}^{2n} \to \{0,1\}$ such that $\texttt{COMP}_n(a_{n-1}, \ldots, a_0, b_{n-1}, \ldots, b_0)$ returns 1 provided that the natural number represented by the string $a_{n-1} \ldots a_0$ is greater than the natural number represented by $b_{n-1} \ldots b_0$. Use induction to show that for every $n \in \mathbb{Z}^+$ we can implement the function $\texttt{COMP}_n$ using no more than $8n$ `NAND` gates. (*Hint:* You can use without proof that `XOR` and `IF` can each be done using no more than 4 `NAND` gates. I'll leave it open to you to determine the number `NAND` gates that $\texttt{COMP}_1$ requires.)