Math 4350

Homework 1

(2.1) Let x be a Boolean string. What type of number does the Boolean string $x0$ represent?

---

solution: Since $x0$ has 0 in the ones position, $x0$ is an even number.

(2.2) Let x be a Boolean string with exactly one bit a 1. What can you say about the number it represents? Does this identification depend on using the canonical numbering of $\{0,1\}^n$, where n is the length of x?

---

solution: We have $x = 0\cdots010\cdots0$ where the length of x is n. If the 1 is in the position m spots from the right we have the number that x represents as

$$0\cdot 2^n + \cdots + 0\cdot 2^{m+1} + 1\cdot 2^m + 0\cdot 2^{m-1} + \cdots + 0\cdot 2^1 + 0\cdot 2^0 = 2^m$$

a power of 2. This doesn't depend on the canonical numbering.

(2.4) Let x be a Boolean string of even length. Can the Boolean string xxx ever represent a prime number in binary notation? *First find an x of odd length with xxx prime. Then either find an x of even length with xxx prime or prove that no such x exists.*

---

solution: If y is a binary string let $\overline{y}$ represent the corresponding number. If x has length n then a little reflection shows that

$$\overline{xxx} = (2^{2n} + 2^n + 1)\overline{x}$$

and $\overline{xxx}$ can only be a prime if $\overline{x} = 1$.

A few examples x of odd length with $\overline{xxx}$ a prime:

x	$\overline{xxx}$
1	7
001	73
000000001	262657

Now suppose that $x = 0\cdots01$ is of even length $n = 2m$. Then

$$\begin{aligned}\overline{xxx} &= (2^{2n} + 2^n + 1)\overline{x}\\ &= 4^n + 2^{2m} + 1\\ &= 4^n + 4^m + 1\\ &\equiv 1^n + 1^m + 1 \pmod 3\\ &\equiv 0 \mod 3\end{aligned}$$

Consequently $\overline{xxx}$ is divisible by 3 and, since it is clearly larger than 3, it is *not* a prime.

(2.6) Show that a function $f : \mathbb{N} \to \mathbb{N}$ is bounded by a polynomial in n, written $f(n) = n^{O(1)}$, if and only if there is a constant C such that for all sufficiently large n, $f(2n) \leq Cf(n)$. How does C relate to the exponent k of the polynomial? Thus, we can characterize algorithms that run in polynomial time as those for which the amount of work scales up only *linearly* as the size of the data grows linearly. Later we will use this criterion as a benchmark for *feasible* computation.

You may assume that f is an increasing function of n if that is of any use to you.

Oops. Neither direction of this problem is true without some modification. Sorry about that. The important part of the problem (the part we'll be using later in the course) is:

If $f : \mathbb{N} \to \mathbb{N}$ is a monotonically increasing function and there is a constant C such that $f(2n) \leq Cf(n)$ for all sufficiently large n, then $f(n) = O(n^k)$ for some constant k.

solution:

($\Leftarrow$) Suppose $f : \mathbb{N} \to \mathbb{N}$ is a monotonically increasing function and there is a constant C such that $f(2n) \leq Cf(n)$ for all $n \geq N = 2^M$ (some M). We'll show that $f(n) = O(n^k)$ for some k.

Let $m = \lceil \log_2(n) \rceil$, so that $n \leq 2^m < 2n$. Also let $k = \lceil \log_2(C) \rceil$. We'll show that $f(n) \leq Bn^k$ for all $n \geq N$ where $B = 2^k f(2^M)/C^M$. Consider

$$\begin{aligned}
f(n) &\leq f(2^m) && \text{since } f \text{ is increasing} \\
&\leq f(2^{m-1})C^1 && \text{by hypothesis} \\
&\vdots && \text{repeating} \\
&\leq f(2^M)C^{m-M} \\
&= \frac{f(2^M)}{C^M}C^m \\
&= \frac{f(2^M)}{C^M}2^{\log_2(C)m} \\
&= \frac{f(2^M)}{C^M}(2^m)^{\log_2(C)} \\
&< \frac{f(2^M)}{C^M}(2n)^k \\
&= \frac{2^k f(2^M)}{C^M} \cdot n^k \\
&= B \cdot n^k
\end{aligned}$$

This result doesn't hold without the monotonicity assumption (Example 1), and the reverse implication doesn't hold at all (Example 2).

($\Rightarrow$) Suppose $f(n) = \Theta(n^k)$ for some k. We'll show that there is a constant C such that $f(2n) \leq Cf(n)$ for $n \geq N$ (some N).

Since $f(n) = \Theta(n^k)$ for some k, there are positive constants R and S with $Rn^k \leq f(n) \leq Sn^k$ for $n \geq N$ (some N). We'll show that $f(2n) \leq Cf(n)$ for $n \geq N$ where $C = {}^{2^k S}/_{R}$.

$$\begin{aligned}
f(2n) &\leq S(2n)^k && 2n > n \geq N \\
&= 2^k S \cdot n^k \\
&\leq 2^k S \cdot \frac{f(n)}{R} && n \geq N \\
&\leq \frac{2^k S}{R} \cdot f(n) \\
&\leq C \cdot f(n)
\end{aligned}$$

This result doesn't hold if we weaken the hypothesis to $f(n) = O(n^k)$ for some k (Example 2), and the reverse implication doesn't hold at all (Example 3).

Example 1. We exhibit a function $f : \mathbb{N} \to \mathbb{N}$ that satisfies $f(2n) \leq f(n)$ but is not bounded by a polynomial in n.

$$f(n) = \begin{cases} 1 & \text{if } n \text{ is even} \\ 2^n & \text{if } n \text{ is odd} \end{cases}$$

Clearly, because of its values for odd n, $f(n)$ is not bounded by any polynomial in n, but $f(2n) = 1 \leq f(n)$ for any n.

Example 2. We exhibit a monotonically increasing function $f : \mathbb{N} \to \mathbb{N}$ that is $O(n)$, but there is no constant C such that $f(2n) \leq Cf(n)$ for large n.

$$f(n) = m_k \text{ if } m_k \leq n < m_{k+1}$$

where the m_k are defined recursively:

$$\begin{aligned} m_0 &= 1 \\ m_1 &= 2 \\ m_{k+1} &= m_k^2 \qquad \text{for } k \geq 1 \end{aligned}$$

In particular, the m_k are powers of 2. Note that $f(n) \leq n$ for all n with equality iff $n = 1$ or $n = m_k$ for some k.

Now consider $n = {}^{m_{k+1}}\!/_2 \in \mathbb{N}$ since m_{k+1} is even. Then

$$\begin{aligned} f(n) &= f({}^{m_{k+1}}\!/_2) = m_k && \text{since } m_k \leq {}^{m_{k+1}}\!/_2 < m_{k+1} \\ f(2n) &= f(m_{k+1}) = m_{k+1} \\ {}^{f(2n)}\!/_{f(n)} &= {}^{m_{k+1}}\!/_{m_k} \\ &= {}^{m_k^2}\!/_{m_k} \\ &= m_k \end{aligned}$$

Since $m_k \to \infty$ as $k \to \infty$ there is no constant C that satifies $f(2n) \leq Cf(n)$ for all large n.

Example 3. We exhibit a monotonically increasing function $f : \mathbb{N} \to \mathbb{N}$ such that $f(2n) \leq 3f(n)$ for all n but $f(n) \neq \Theta(n^k)$ for any k.

$$f(n) = \begin{cases} 1 & \text{if } n \leq 4 \\ \lfloor n^{1 - \frac{1}{\log_2 \log_2 n}} \rfloor & \text{otherwise} \end{cases}$$

Since $1 - \frac{1}{\log_2 \log_2 n} < 1$ we clearly have $f(n) = O(n)$. On the other hand $f(n) \neq \Omega(n)$ since

$${}^{n}\!/_{f(n)} \geq n^{\frac{1}{\log_2 \log_2 n}} \to \infty \text{ as } n \to \infty$$

Clearly, for any $k > 1$, $f(n) = O(n^k)$ and $f(n) \neq \Omega(n^k)$. Also, for any $\epsilon > 0$, similar limits show that $f(n) = \Omega(n^{1-\epsilon})$ but $f(n) \neq O(n^{1-\epsilon})$. So $f(n)$ is not $\Theta(n^k)$ for any k.

The fact that $f(2n) \leq 3f(n)$ is left as an exercise for the reader.