CSC207H1 SUMMER 2018: PROBLEM SET 1

BY: ERIC KOEHLI

Date: May 26, 2018.

1. Problem 1

Prove that $\forall n \in \mathbb{N}, n \geqslant 1 \implies 9 \mid 4^n + 15n - 1$.

Proof. Let $n \in \mathbb{N}$ and define $P(n) : 9 \mid 4^n + 15n - 1$. We will prove $P(n)$ is true for all natural numbers $n \geqslant 1$ by induction.

Base Case:

Let $n = 1$. Then $4^1 + 15(1) - 1 = 18$. Since $9 \mid 18$, we have shown $P(1)$ is true.

Inductive Step:

Let $k \in \mathbb{N}$ and assume $P(k)$ is true. With the definition of divisibility, we are assuming that $\exists c_k \in \mathbb{Z}, 4^k + 15k - 1 = 9c_k$. We want to show that $P(k+1)$ is also true. That is, $P(k+1) : \exists c_{k+1} \in \mathbb{Z}, 4^{k+1} + 15(k+1) - 1 = 9c_{k+1}$.

Let $c_{k+1} = \frac{1}{3}(4^k - 1) + 18 + 9c_k$. From the LHS, we have

$$\begin{aligned}
4^{k+1} + 15(k+1) - 1 &= 4 \cdot 4^k + 15k + 15 - 1 \\
&= 3 \cdot 4^k + 4^k + 15k + 15 - 1 \\
&= 3 \cdot 4^k - 3 + 18 + (4^k + 15k - 1) \\
\text{(By I.H.)} \qquad &= 3(4^k - 1) + 18 + 9c_k \\
&= 9 \cdot \Big[\frac{1}{3}(4^k - 1) + 2 + c_k\Big] \\
&= 9c_{k+1}
\end{aligned}$$

The final step is not justified until we can show that $3 \mid (4^k - 1)$ (i.e. is a multiple of 3). We will prove this by induction (albeit less formally than the current proof).

Base Case:

Let $k = 0$. Then $4^0 - 1 = 0$ and $3 \mid 0$. Thus $P(0)$ is true.

Inductive Step:

Assume $P(k) : 4^k - 1 = 3p$ for some $p \in \mathbb{Z}$. We will show $P(k+1) : 4^{k+1} - 1 = 3q$ for $q \in \mathbb{Z}$. Let $q = 4^k + p$. From the LHS, we have

$$\begin{aligned}
4^{k+1} - 1 &= 3 \cdot 4^k + (4^k - 1) \\
\text{(By the I.H.)} \qquad &= 3 \cdot 4^k + 3p \\
&= 3(4^k + p) \\
&= 3q
\end{aligned}$$

This shows that $4^k - 1$ is a multiple of three, which now justifies our final step in the original proof. Thus $P(k+1)$ follows from $P(k)$ and this completes the induction step. Having shown steps 1 and 2, we can conclude by the Principle of Mathematical Induction that $P(n)$ is true for all natural numbers $n \geqslant 1$.

$\square$

2. Problem 2

(a) Prove that every natural number $n \geqslant 1$ has a binary representation.
(b) Prove that the binary representation is unique. That is, $\forall n \in \mathbb{N}$, n is a binary representation $\implies$ n is unique.

Proof. Proof for (a). Since binary numbers are written as powers of 2, we will show that any number n (base 10) can be written as a sum of powers of 2. We will use strong induction to show that this is true. Let $n \in \mathbb{N}$ (base 10) and define $P(n) : n$ has a binary representation as the equation:

$$n = \sum_{i=0}^{r} b_i 2^i$$

where $r \in \mathbb{Z}^+$, and $b_i = 0, 1$ for $i = 0, 1, \ldots, r$.

<u>**Base Case**</u>:

We can actually show 0 and 1 are true (even though we are only asked for $n \geqslant 1$). Let $n = 0$, $r = 0$, $b_0 = 0$. Then

$$0 = \sum_{i=0}^{0} b_i 2^i = b_0 2^0 = 0 \cdot 1 = 0$$

To show $P(1)$, let $n = 1$, $r = 0$, $b_0 = 1$. Then

$$1 = \sum_{i=0}^{0} b_i 2^i = b_0 2^0 = 1 \cdot 1 = 1$$

Thus $P(0)$ and $P(1)$ are true.

<u>**Inductive Step**</u>:

Let $k \in \mathbb{N}$ (base 10) and assume $P(0), P(1), P(2), \ldots, P(k)$ are all true. That is,

$$P(j) : j = \sum_{i=0}^{r} b_i 2^i$$

for $0 \leqslant j \leqslant k$. We now want to show that $P(k+1)$ is true. We'll split the proof into two cases depending on whether $k+1$ is even or odd.

Case 1: k + 1 is even

In this case, $\frac{k+1}{2}$ is an integer and $0 \leqslant \frac{k+1}{2} \leqslant k$. Then we can use the induction hypothesis:

$$\frac{k+1}{2} = \sum_{i=0}^{r} b_i 2^i$$

(multiply both sides by 2)

$$k + 1 = \sum_{i=0}^{r} b_i 2^{i+1}$$

Case 2: k + 1 is odd

In this case $\frac{k}{2}$ is an integer and $0 \leqslant \frac{k}{2} \leqslant k$, so the induction hypothesis applies and we get

$$
\begin{aligned}
\frac{k}{2} &= \sum_{i=0}^{r} b_i 2^i \\
k &= \sum_{i=0}^{r} b_i 2^{i+1} \\
k+1 &= \sum_{i=0}^{r} b_i 2^{i+1} + 2^0
\end{aligned}
$$

(since $2^0 = 1$). Thus $P(k+1)$ follows from $P(k)$ and this completes the induction step. Having showing steps 1 and 2, we can conclude by the Principle of Strong Induction that $P(n)$ is true for all natural numbers n (i.e. for every natural number, there *exists* has a binary representation).

□

Proof. Proof of (b), the uniqueness of binary numbers. We will prove that the binary representation of n is unique by contradiction.

Suppose that n is not unique. Then there must exist some $m \in \mathbb{N}$ (base 10) such that $m \neq n$, but m and n have the same binary representation:

$$
\sum_{i=0}^{r} b_i 2^i = \sum_{i=0}^{s} c_i 2^i
$$

for some positive arbitrary integers r and s and "bits" $b_i, c_i \in \{0, 1\}$. We may now assume that $r > s$ (or $r < s$). Then we can show that

$$
\begin{aligned}
&\text{(by geometric series)} & 2^r &> 2^{s+1} - 1 \\
&\text{(geometric series expanded)} & &= 1 + 2 + \cdots + 2^{s-1} + 2^s \\
& & &= \sum_{i=0}^{s} 2^i \\
& & &\geqslant \sum_{i=0}^{s} c_i 2^i
\end{aligned}
$$

The last step follows since some c_i *can* equal 0. Therefore, this shows that

$$
\sum_{i=0}^{r} b_i 2^i > \sum_{i=0}^{s} c_i 2^i
$$

which contradicts our assumption that n is not unique. Then it must follow that the binary representation of n is indeed unique. □

3. Problem 3

Prove $\forall n \in \mathbb{N}$, $n > 1 \implies 2^{n/2} \leqslant g(n) \leqslant 2^n$.

Proof. Let $n \in \mathbb{N}$ and define $P(n) : 2^{n/2} \leqslant g(n) \leqslant 2^n$. We will prove that $P(n)$ is true for all natural numbers $n > 1$ by complete induction.

Base Case:

Let $n = 2$. Then

$$\begin{aligned} 2^{2/2} = 2^1 = 2 \leqslant g(2) & \\ &= g(2-2) + g(2-1) = g(0) + g(1) = 4 \\ &\leqslant 2^2 = 4 \end{aligned}$$

Thus $P(2)$ is true.

Inductive Step:

Let $k \in \mathbb{N}$ and assume $P(2), P(3), \ldots, P(k)$ is true. For reference, we are assuming that the following holds:

(1) $2^{k/2} \leqslant g(k) \leqslant 2^k$
(2) $2^{(k-1)/2} \leqslant g(k-1) \leqslant 2^{k-1}$

We want to show $P(k+1) : 2^{(k+1)/2} \leqslant g(k+1) \leqslant 2^{k+1}$. That is, we want to show the following two conditions hold:

i) $2^{(k+1)/2} \leqslant g(k+1)$
ii) $g(k+1) \leqslant 2^{k+1}$

We begin by showing *(i)* as follows

$$\begin{aligned} g(k+1) &= g(k) + g(k-1) \\ \text{(by (1) and (2))} \qquad &\geqslant 2^{k/2} + 2^{(k-1)/2} \\ &= 2^{k/2} + 2^{k/2} \cdot 2^{-1/2} \\ &= 2^{k/2} \cdot (1 + \frac{1}{\sqrt{2}}) \\ \text{(by calculator)} \qquad &\geqslant 2^{k/2} \cdot 2^{1/2} \\ &= 2^{(k+1)/2} \end{aligned}$$

This shows that the first condition holds.

For *(ii)* we have

$$
\begin{aligned}
g(k+1) &= g(k) + g(k-1) \\
\text{(by (1) and (2))} \qquad &\leqslant 2^k + 2^{k-1} \\
&\leqslant 2^k + 2^k \\
&= 2^{k+1}
\end{aligned}
$$

By showing *(i)* and *(ii)* holds, we can conclude that $P(k+1)$ follows from $P(2), P(3), \ldots, P(k)$, and this completes the induction step. Having shown steps 1 and 2 we can conclude by complete induction that $P(n)$ is true for all natural numbers $n > 1$.

$\square$

4. Problem 4

Let $n \in \mathbb{N}$ and let $T(n)$ denote the time complexity of $L(n)$ for inputs n. The base case is when $n < 7$ and the function $L(n)$ runs in constant time. Thus $T(n) = c$ in this case.

If $n \geqslant 7$, then $L(n)$ makes a recursive call. To analyze the runtime we need to consider the recursive and non-recursive parts separately.

For the non-recursive part, only a constant number of steps occur (the `if` check, the addition, and the `return`), so we can say that the non-recursive part takes d steps.

The recursive call is $L(n/7)$, which has a worst-case runtime of $T(n/7)$. Thus when $n \geqslant 7$, we get the recurrence relation $T(n) = T(n/7) + d$. Therefore the full recursive definition of T is:

$$T_1(n) = \begin{cases} c & \text{if } n < 7 \\ T(n/7) + d & \text{otherwise} \end{cases}$$

Now we want to find the closed form of T. We first make the substitution $m = \log_7(n)$. Then $n = 7^m$.

$$\begin{aligned}
&(\text{m} = 0) && T(7^0) = T(1) = c + 0d \\
&(\text{m} = 1) && T(7^1) = T(7) = T(1) + d = c + 1d \\
&(\text{m} = 2) && T(7^2) = T(49) = T(7) + d = c + 2d \\
& && \vdots \\
& && T(7^m) = \cdots = c + md
\end{aligned}$$

Then when we convert back to n, we get the closed form shown below

$$T_2(n) = c + d \cdot \log_7(n)$$

Proof of Correctness: Prove $\forall n \in \mathbb{N},\ n \geqslant 7 \implies T_2(n) = c + d \cdot \log_7(n)$.

Proof. Let $n \in \mathbb{N}$ and define $P(n) : T_1(n)$ is equivalent to the closed form $T_2(n)$. We want to prove that $T_2(n)$ is the closed form of $T_1(n)$.

Base Case:

Let $n = 7$. Then $T_1(7) = T_1(7/7) = T_1(1) = c + d$. Also, $T_2(7) = c + d \cdot \log_7(7) = c + d$. Thus $P(7)$ is true.

Inductive Step:

Let $m, k \in \mathbb{N}$ and assume $P(m)$ is true for $7 \leqslant m \leqslant k$. That is, $P(m) : T_1(m)$ is equivalent to $T_2(m) = c + d \cdot \log_7(m)$. We want to show $P(k+1) : T_1(k+1)$ is equivalent to $T_2(k+1) = c + d \cdot \log_7(k+1)$. From our time complexity function, we have

$$T_1(k+1) = \begin{cases} c & \text{if } k+1 < 7 \\ T(\frac{k+1}{7}) + d & \text{otherwise} \end{cases}$$

For reference we note that (a): $7 \leqslant \frac{k+1}{7} \leqslant k$. Then we have

$$\begin{aligned} T_1(k+1) &= T_1(\frac{k+1}{7}) + d \\ \text{(by I.H. and (a))} \qquad &= c + d \cdot \log_7(\frac{k+1}{7}) + d \\ &= c + d \cdot \left[\log_7(k+1) - \log_7(7)\right] + d \\ &= c + d \cdot \left[\log_7(k+1) - 1\right] + d \\ &= c + d \cdot \log_7(k+1) - d + d \\ &= c + d \cdot \log_7(k+1) \\ &= T_2(k+1) \end{aligned}$$

Thus $P(k+1)$ follows from $P(m)$ by complete induction. Since the basis and induction steps have been shown, we can conclude that $P(n)$ is true for all natural numbers $n \geqslant 7$. Thus we have shown that the closed form T_2 is equivalent to the time complexity function T_1. This completes the proof of correctness.

$\square$

5. Problem 5

Define the set F recursively as follows:

(a) $7 \in F$

(b) if $u, v \in F$, then $u + v \in F$

(c) Nothing else is in F

Prove by structural induction that $\forall w \in F$, $w \bmod 7 = 0$ (i.e. $7 \mid w$).

Proof. We will prove by structural induction that $\forall w \in F$, $7 \mid w$. Given any element in F, let the property P be the claim that 7 divides the element. That is, define $P(w) : 7 \mid w$.

Base Case:

We want to show that each element in the base of F satisfies P. The only element in the base of F is 7, and we know that $7 \mid 7$. Thus, $P(7)$ is true.

Inductive Step:

We now want to show that for each rule in the recursion for F, if the rule is applied to an element in F that satisfies the property P, then the element defined by the rule also satisfies the property P.

The recursive rule for F consists of one step denoted by (b) above.

Suppose $u, v \in F$ are any two elements such that $7 \mid u$ and $7 \mid v$. (This is our induction hypothesis). That is, expanding the definition of divisibility, we have $u = 7k_1$ and $v = 7k_2$ for $k_1, k_2 \in \mathbb{Z}$. When rule (b) is applied to u and v, the result is $u + v$, which we know is also in F (by the recursive definition of F). So we must now show that $u + v$ is divisible by 7. That is, $u + v = 7k_3$ for $k_3 \in \mathbb{Z}$. Let $k_3 = k_1 + k_2$. Then by substitution, we have

$$\begin{aligned} u + v &= 7k_1 + 7k_2 \\ &= 7(k_1 + k_2) \\ &= 7k_3 \end{aligned}$$

This shows that $7 \mid u + v$. Thus when the recursive rule is applied to any element divisible by 7 in F, the result is another element (say $w = u + v$), that is also divisible by 7. Therefore all elements in F are divisible by 7 (i.e. $P(w)$ is true for all $w \in F$).

$\square$