Homework 1 Solutions
Qiyam Tung
June 22, 2014

1 Problem 1
If a computer could represent all real numbers, then function will never terminate. If there exists such a number \( x \) such that \( x/2 = 0 \), then that implies that \( 0 \times 2 = x \). But that number can only be 0, and therefore since the input is a non-zero number, the function will never terminate. Alternatively, it can be proved by arguing that a number exists between every two numbers on the real number line.

2 Problem 2

\[
\log_b a = \frac{\log_{10} a}{\log_{10} b}
\]  

Assume that \( \log_b a = y \), then by raising both sides with a base of \( b \), we get

\[
a = b^y
\]

Now take the \( \log_{10} \) of both sides

\[
\log_{10} a = y \cdot \log_{10} b
\]

\[
\frac{\log_{10} a}{\log_{10} b} = y = \log_b a
\]

3 Problem 3

3.1 2.19a

Proof (Inductive):

Basis: \( n=1 \)
\( 1^2 - 1 = 0 \) and is therefore even.
Inductive: \( n^2 - n \) is even \( \rightarrow (n + 1)^2 - (n + 1) \) is even

\[
(n + 1)^2 - (n + 1) = n^2 + 2n + 1 - n - 1 = (n^2 - n) + 2n
\]

From our hypothesis, we know that \( n^2 - n \) is even. And since \( 2n \) is even and addition of even numbers is even, \( (n + 1)^2 - (n + 1) \) is even.

QED

3.2 2.19b

Proof (deductive):

First, without loss of generality, we assume that \( n \) is even.
Then, \( n^2 - n = n(n - 1) \). Now, since \( n - 1 \) is odd and even multiplied by odd is even, \( n^2 - n \) is always even.

QED

3.3 2.19c
Proof (deductive):
By definition, every 3rd number is divisible by 3.
\[ n^3 - n = n(n^2 - 1) = n(n - 1)(n + 1) = (n - 1)n(n + 1) \]
Given any value \( n \), either \( n - 1 \), \( n \), or \( n + 1 \) is a multiple of 3.
QED

3.4 2.19d

Proof (Inductive):
Basis: \( n=1 \)
\[ 1^5 - 1 = 0 \] and is divisible by 5.
Inductive: \( n^5 - n \) is even \( \rightarrow (n + 1)^5 - (n + 1) \) is even
\[ (n + 1)^5 - (n + 1) = n^5 + 5n^4 + 10n^3 + 10n^2 + 5n + 1 - n - 1 = (n^5 - n) + 5(n^4 + 2n^3 + 2n^2 + n) \]
From our hypothesis, we know that \( n^5 - n \) is even. The second term in parenthesis is a
also divisible by 5, making the whole expression divisible by 5.
QED

4 Problem 4
Let \( \sqrt{3} = \frac{a}{b} \), then \( 3 = \frac{a^2}{b^2} \) or \( a^2 = 3b^2 \) [1]. We now consider the following cases.

case 1: \( b \) is even. In this case \( b^2 \) is also even. Since \( a^2 = 3b^2 \), we can conclude that \( a^2 \) is even, and hence \( a \) is even. Since \( a \) and \( b \) can be reduced by canceling a common factor of 2, we have a contradiction.

case 2: \( b \) is odd. In this case \( b^2 \) is also odd. Since \( a^2 = 3b^2 \), we can conclude that \( a^2 \) is odd, and hence \( a \) is odd. Let
\[ a = 2m + 1 \] and \( b = 2n + 1 \). Substituting in the equation [1] we get
\[ 3(4m^2 + 4n + 1) = 4m^2 + 4m + 1 \], or
\[ 6n^2 + 6n + 1 = 2(m^2 + m) \]. Since the left hand side is an odd number and the right hand side is an even number
we have a contradiction.

Hence \( \sqrt{3} \) is not rational.

Here we show that \( a^2 \) and \( a \) are of same parity. To show this we consider \( a^2 - a = a(a - 1) \). Since either \( a \) or \( a - 1 \)
is even, we get \( a^2 - a \) is even, which implies \( a^2 \) and \( a \) are of same parity.

5 Problem 5
If we represent every object the position on a bit vector and use 0’s to represent absence and 1’s to indicate presence,
then 001010 would represent a subset of the power set containing only the 2nd and 4th element for a set containing 6
objects. Therefore, representing all possible subsets is the same as asking all possible permutations of the bit string,
which is \( 2^*2*2*2*2 = 2^6 \) for a set with 6 objects. Then, for any arbitrary \( n \), the total number of subsets is then \( 2^n \).

6 Problem 6

```c
function reverse_list(Node head)
{
    if head == NULL
        return head
    else if head.next == NULL
        return head
    else
        Node cur = head
        Node prev = cur
        Node next = cur.next
        cur.next = NULL
        cur = next
        next = prev
        next.next = cur
        return head
```
while next != NULL
    prev = cur
    cur = next
    next = cur.next
    cur.next = prev

    return cur
}