CAB203

Tutorial 1

Foundations

1. Some simple algebra shows $(x+1)(x+2) = x^2 + 3x + 2$. Apply the definition of divides $(a|b)$ to obtain two new statements.

> **Solution:** $a|b$ means that there is some c such that $ac = b$. Here there are two ways that we can fill in a, b, c:
>
> - $a = x+1$, $b = x^2 + 3x + 2$, $c = x + 2$. Hence $x + 1 \mid x^2 + 3x + 2$.
> - $a = x+2$, $b = x^2 + 3x + 2$, $c = x + 1$. Hence $x + 2 \mid x^2 + 3x + 2$.

2. A *square number* is a natural number n such that $n = x^2$ for some integer x. Apply this definition to show that 25 and 121 are square numbers.

> **Solution:** We need to find x so that $x^2 = 25$. Clearly $x = 5$ works hence 25 is a square number. Likewise, we know that $11^2 = 121$ hence 121 is a square number.

3. Using only the partial list of axioms on slide 25 of Lecture 1, show that $(0 + a) + 0 = a$.

> **Solution:** There are many possibly ways of doing this. Here is one example.
>
> *Proof.*
> $$\begin{aligned} (0+a)+0 &= 0 + (a+0) && \text{by axiom 3} \\ &= 0 + a && \text{by axiom 4} \\ &= a + 0 && \text{by axiom 2} \\ &= a && \text{by axiom 4} \end{aligned}$$
>
> $\square$

Modular arithmetic

Calculate:

1. $17 \mod 5$

> **Solution:** We start by doing division, keeping track of the remainder
>
> $$17 = 5 \cdot 3 + 2$$
>
> The remainder is 2, so
>
> $$17 \mod 5 = 2.$$

2. $36 \mod 7$

Solution: Again, start with division:

$$36 = 5 \cdot 7 + 1$$

The remainder is 1, so

$$36 \mod 7 = 1.$$

3. $(5 + 7) \mod 3$

Solution: We'll try two different ways. First, we have

$$5 + 7 = 12$$

and

$$12 \mod 3 = 0$$

so

$$5 + 7 \mod 3 = 0.$$

A second way of doing this is to first find

$$5 \mod 3 = 2$$

$$7 \mod 3 = 1$$

from this we see

$$5 + 7 \mod 3 = 2 + 1 \mod 3 = 3 \mod 3 = 0.$$

4. $(5 \cdot 4) \mod 3$

Solution: Again, we'll try two different ways. First

$$5 \cdot 4 = 20$$

so

$$5 \cdot 4 \mod 3 = 20 \mod 3.$$

Now

$$20 = 6 \cdot 3 + 2$$

so

$$5 \cdot 4 \mod 3 = 2$$

Another way of doing this is to find

$$5 \mod 3 = 2$$

$$4 \mod 3 = 1$$

from which we find

$$5 \cdot 4 \mod 3 = 2 \cdot 1 \mod 3 = 2$$

5. Write a short Python function that takes two numbers, a and b, and returns True when $a|b$.

Solution: We will take advantage of the Lemma from the lectures stating that if $a \mod b = 0$ then $a|b$.

```
def divides(a,b):
   return b % a == 0
```

6. Write a short Python function that takes three numbers, a, b, c and returns True when $a \equiv b \pmod{c}$.

Solution: We will use the definition of modular equivalence. We need to test whether $c|(a-b)$. We can use the answer to the previous question.

```
def mod_equiv(a,b,c):
   return divides(c, a - b)
```

7. *Stretch question* Show that if r is the remainder when a is divided by b (i.e. $a = bq + r$) then $a \equiv r \pmod{b}$.

Solution: Rearrange to get $a - r = bq$. Then clearly b divides $a - r$ and so $a \equiv r \pmod{b}$

8. *Stretch question* Apply the definition of modular equivalence to show that if $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ then $a \equiv c \pmod{n}$.

Solution: We have $a - b = nq$ for some q and $b - c = np$ for some p. Then

$$\begin{aligned} a - c &= a - b + (b - c) \\ &= nq + np \\ &= n(q + p) \end{aligned}$$

so $a - c$ is divisible by n.

Exponents

Use the laws of exponents and logarithms to calculate the following (don't use a calculator):

1. $2^{15} \cdot 2^3$ bits expressed in kilobits

Solution: First, recall that a kilobit is $1024 = 2^{10}$ bits. Also, recall (or easily calculate) that $2^8 = 256$. Using properties of exponents, we find

$$\begin{aligned} 2^{15} \cdot 2^3 &= 2^{15+3} \\ &= 2^{18} \\ &= 2^{10+8} \\ &= 2^8 \cdot 2^{10} \\ &= 256 \cdot 2^{10} \end{aligned}$$

So this is 256 kilobits.

2. Express 8^5 with base 2

Solution:

$$\begin{aligned} 8^5 &= \left(2^3\right)^5 \\ &= 2^{3 \cdot 5} \\ &= 2^{15} \end{aligned}$$

3. $\log_2 \frac{256}{16}$

Solution: Using mostly properties of exponents:

$$\begin{aligned} \log_2 \frac{256}{16} &= \log_2 \frac{2^8}{2^4} \\ &= \log_2 2^4 \\ &= 4 \end{aligned}$$

Or using mostly properties of logs:

$$\begin{aligned} \log_2 \frac{256}{16} &= \log_2 256 - \log_2 16 \\ &= 8 - 4 \\ &= 4 \end{aligned}$$

4. $\log_2 8^3$

Solution: Using properties of exponents:

$$\begin{aligned} \log_2 8^3 &= \log_2 \left(2^3\right)^3 \\ &= \log_2 2^9 \\ &= 9 \end{aligned}$$

Or using base transformation, first rearrange to get

$$\log_a x = \log_a b \cdot \log_b x$$

then apply:

$$\begin{aligned} \log_2 8^3 &= \log_2 8 \cdot \log_8 8^3 \\ &= 3 \cdot 3 \\ &= 9 \end{aligned}$$

5. Write a short Python program that takes a number of addresses n and returns the minimum number of bits required to express that many unique addresses. Note that n might not be a power of 2. You may wish to us `math.ceil()`.

Solution: We want $\lceil \log_2 n \rceil$:

```
import math
def min_bits(n):
   return math.ceil(math.log2(n))
```

6. *Stretch question* Use the fact that $n^x \cdot n^y = n^{x+y}$ to show that $\log_n ab = \log_n a + \log_n b$ using only the basic properties of logarithms (i.e. $\log_n n^x = x$ and $x = n^{\log_n x}$).

Solution: First, from the definition of logarithms, we have $a = n^{\log_n a}$ and $b = n^{\log_n b}$. Now we apply the given property of exponents to find

$$ab = n^{\log_n a} \cdot n^{\log_n b} = n^{\log_n a + \log_n b}$$

Now we take the log of both sides of the equation to get

$$\log_n ab = \log_n a + \log_n b$$

where we have used the definition of logarithms to see that $\log_n n^x = x$ on the right hand side.