CSC110 Lecture 17: Modular Arithmetic

Hisbaan Noorani

October 21, 2020

Contents

For your reference, here is the definition of modular equivalence.

> Let $a, b, n \in \mathbb{Z}$, with $n \neq 0$. We say that a **is eqvuialent to** b **modulo** n when $n \mid a - b$. In this case, we write $a \equiv b \pmod{n}$.

1 Ex 1: Modular arithmetic practice

1. Expand the statement $14 \equiv 9 \pmod{5}$ into a statement using the divisibility predicate. Is this statement True or False?

   $= 5 \mid (14 - 9)$

   $= 5 \mid 5$

   This is true ✓

2. Expand the statement $9 \equiv 4 \pmod{3}$ into a statement using the divisibility predicate. Is this statement True or False?

   $3 \mid (9 - 4$

   $3 \mid 5$

   This is false. $3 \neq 5 \cdot k$ for any integer k.

3. Prove the following statement using *only* the definitions of divisibility and modular equivalence (and no other statements/theorems):

   WTS $\forall a, b, c \in \mathbb{Z},\ \forall n \in \mathbb{Z}^+, a \equiv b \pmod{n} \Rightarrow ca \equiv cb \pmod{n}$

   Let $a, b, c \in \mathbb{Z}$

   Let $n \in \mathbb{Z}^+$

   Assume $a \equiv b \pmod{n}$. This implies $n \mid a - b$. This again implies $a - b = np_1, p \in \mathbb{Z}$

   Prove $ca \equiv cb \pmod{n}$.

   We can rewrite this as: $n \mid ca - cb$

   This means:

   $ca - cb = np_2, p_2 \in \mathbb{Z}$

   $c(a - b) = ncp_1, p1 \in \mathbb{Z}$

$a - b = np_1 \in \mathbb{Z}$. We have arrived at our assumption.

We have thus proven that $ca \equiv cb \pmod{n}$ as needed. ■

2 Ex 2: Modular division

Recall that last class, we implemented the following function:

```
def extended_gcd(a: int, b: int) -> Tuple[int, int, int]:
    """Return the gcd of a and b, and integers p and q such that
    gcd(a, b) == p * a + b * q.

    >>> extended_gcd(10, 3)
    (1, 1, -3)
    """
    ...
```

This class, we proved that for any $n \in \mathbb{Z}^+$ and $a \in \mathbb{Z}$, a has an inverse modulo n as long as $\gcd(a, n) = 1$. The proof we wrote can be turned into an algorithm for actually computing this modular inverse. To solidify your knowledge of this proof, complete the following function *using* `extended_gcd` *as a helper*. Make sure to include the appropriate precondition(s) based on the statement of the theorem!

```
def modular_inverse(a: int, n: int) -> int:
    """Return the inverse of a modulo n, in the range 0 to n - 1 inclusive.

    Preconditions:
        - extended_gcd(a, n)[0] == 1

    >>> modular_inverse(10, 3)  # 10 * 1 is equivalent to 1 modulo 3
    1
    >>> modular_inverse(3, 10)  # 3 * 7 is equivalent to 1 modulo 10
    7
    """
    gcd, p, q = extended_gcd(a, n)

    assert gcd == 1

    if p > 0:
        return p
    else:
        return n + p

    # You could also use range(0, n - 1) here to get p here
    # by testing every one until one works. I would have
    # done it that way but mario's solution looked good so...
```

3 Ex 3: Exponentiation and order

Consider modulo 5, which has the possible remainders $0, 1, 2, 3, 4$. In each table, fill in the value for remainder b, where $0 \leq b < 5$, that makes the modular equivalence statement in each row True. The first table is done for you. Use Python as a calculator if you would like to. (Or write a comprehension to calculate them all at once!)

1. Powers of 2.

Power of 2	Value for b
$2^1 \equiv b \pmod 5$	2
$2^2 \equiv b \pmod 5$	4
$2^3 \equiv b \pmod 5$	3
$2^4 \equiv b \pmod 5$	1
$2^5 \equiv b \pmod 5$	2
$2^6 \equiv b \pmod 5$	4

2. Powers of 3.

Power of 3	Value for b
$3^1 \equiv b \pmod 5$	3
$3^2 \equiv b \pmod 5$	4
$3^3 \equiv b \pmod 5$	2
$3^4 \equiv b \pmod 5$	1
$3^5 \equiv b \pmod 5$	3
$3^6 \equiv b \pmod 5$	4

3. Powers of 4.

Power of 4	Correct value for b
$4^1 \equiv b \pmod 5$	4
$4^2 \equiv b \pmod 5$	1
$4^3 \equiv b \pmod 5$	4
$4^4 \equiv b \pmod 5$	1
$4^5 \equiv b \pmod 5$	4
$4^6 \equiv b \pmod 5$	1

4. Using the tables above, write down the *order* of 2, 3, and 4 modulo 5:

n	$\text{ord}_5(n)$
2	4
3	4
4	2

4 Additional Exercises

1. Using only the definition of divisibility and the definition of congruence modulo n, prove the following statements.

   (a) $\forall a, b, c, d \in \mathbb{Z},\ \forall n \in \mathbb{Z}^+,\ a \equiv b \pmod n \land c \equiv d \pmod n \Rightarrow a + c \equiv b + d \pmod n$

   (b) $\forall a, b \in \mathbb{Z},\ \forall n \in \mathbb{Z}^+,\ (0 \leq a < n) \land (0 \leq b < n) \land (a \equiv b \pmod n) \Rightarrow a = b.$

2. Implement the following function, which is the modular analog of division. Use your `modular_inverse` function from above. Once again, figure out what the necessary precondition(s) are for this function.

```python
def modular_divide(a: int, b: int, n: int) -> int:
    """Return an integer k such that ak = b (mod n).

    The return value k should be between 0 and n-1, inclusive.

    Preconditions:

    >>> modular_divide(7, 6, 11)  # 7 * 4 is equivalent to 6 modulo 11
    4
    """
```