Lecture 17: Modular Arithmetic

Wednesday, October 19, 2022 11:03 PM

CSC110 Lecture 17: Modular Arithmetic

Print this handout

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 equivalent to** b **modulo** n when $n \mid a - b$. In this case, we write $a \equiv b \pmod n$.

Exercise 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?

$5 \mid 14-9$ True

$\hookrightarrow 5 \mid 5$

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

$3 \mid 9-4$ False

$\hookrightarrow 3 \mid 5$

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

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

Let $a,b,c \in \mathbb{Z}$. Let $n \in \mathbb{Z}^+$

Assume $a \equiv b \pmod n$, i.e., $\exists k_1 \in \mathbb{Z},\ b-a = k_1 n$.

We want to show: $ca \equiv cb \pmod n$, We want to show: $ca \equiv cb \pmod{n}$,

i.e., $\exists k_2 \in \mathbb{Z},\ cb - ca = k_2 n$.

Let $k_2 = \underline{ck_1}$.

...

Rough

(given)

$b - a = k_1 n$

$\times c$

(want)

$cb - ca = \underline{ck_1} n$

Exercise 2: Modular division

In lecture, we proved the following theorem about the existence of modular inverses. For your reference, we've also included an abridged proof with just the key steps shown.

> *Modular inverse theorem*:
>
> $\forall n \in \mathbb{Z}^+,\ \forall a \in \mathbb{Z},\ \ \gcd(a, n) = 1 \Rightarrow \big(\exists p \in \mathbb{Z},\ ap \equiv 1 \pmod{n}\big).$
>
> *Key proof steps*:
>
> - Assuming $\gcd(a, n) = 1$, by the GCD Characterization Theorem there exist $p, q \in \mathbb{Z}$ such that $1 = ap + qn$.
> - Then $qn = 1 - ap$
> - Then $ap \equiv 1 \pmod{n}$.

Now, you'll turn this proof into an algorithm. In the code below, we've provided the `extended_euclidean_gcd` function from last class, as well as the specification for a new `modular_inverse` function. Your task is to complete `modular_inverse` by *writing appropriate precondition(s)* and then *writing the function body*. Recall that last class, we implemented the following function:

$10 \times 5 \equiv 1 \pmod{7}$

$10 \times 5 \equiv 2 \pmod{3}$

WTS $\exists p \in \mathbb{Z},\ \underbrace{ap \equiv 1 \pmod{n}}$

$n \mid 1 - ap$

$\Downarrow$

$\exists k \in \mathbb{Z},\ 1 - ap = kn$

$10 \times$ ______ $\equiv 1 \pmod{1112257}$

$ak = b$

$k = \frac{b}{a}$

${}^{p\cdot}ak \equiv {}^{p\cdot}b \pmod{n}$

$pak \equiv pb \pmod{n}$

$k \equiv pb \pmod{n}$

(since $pa \equiv 1 \pmod{n}$)