CS131 Fall 2025 Midterm 2 Practice Problem Solutions

Problem 1. Convert the following sets from roster notation to set-builder notation.

a) The power set of the natural numbers: $\{\emptyset, \{0\}, \{1\}, ..., \{0, 1, 2, 3, ...\}, ...\}$. Do not use $P(\mathbb{N})$.

Solution. $\{s : s \subseteq \mathbb{N}\}$

b) The set of odd numbers: $\{1, -1, 3, -3, 5, -5, ...\}$

Solution. $\{x : \exists k \in \mathbb{Z} \text{ s.t. } x = 2k + 1\}$

c) $\{1, 2, 3, 4, 5, 6\}$

Solution. $\{x \in \mathbb{N} : x > 0 \land x < 7\}$

d) $\{0, -1, 2, -3, 4, -5, 6, ...\}$

Solution. $\{x : \exists i \in \mathbb{N} \text{ s.t. } x = (-1)^i \cdot i\}$

e) The set of prime numbers: $\{2, 3, 5, 7, 11, 13, ...\}$. Remember that primes are strictly positive.

Solution. $\{x : \forall k \in \mathbb{Z}^+, k \mid x \to (k = 1 \lor k = x)\}$

Problem 2. Prove that $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (without using set identities; use the same format as on Problem 1 of Problem Set 7)

Solution.

$$\begin{aligned}
x \in A \cup (B \cap C) &\equiv x \in A \lor (x \in B \cap C) && \text{by definition of } \cup \\
&\equiv x \in A \lor (x \in B \land x \in C) && \text{by definition of } \cap \\
&\equiv (x \in A \lor x \in B) \land (x \in A \lor x \in C) && \text{by Distributive Law} \\
&\equiv (x \in A \cup B) \land (x \in A \cup C) && \text{by definition of } \cup \\
&\equiv x \in (A \cup B) \cap (A \cup C) && \text{by definition of } \cap
\end{aligned}$$

Problem 3. Prove that $A \cap (B \cup C) \subseteq A \cup B \cup C$ (without using set identities)

Solution. Suppose $x \in A \cap (B \cup C)$. Then $x \in A \land (x \in B \lor x \in C)$ (by definitions of $\cap$ and $\cup$), so $x \in A$, so $x \in A \lor x \in B \lor x \in C$.

Problem 4.

a) For each of the following functions, state whether it is injective and whether it is surjective. Fill in the box if yes, and leave it empty if no.

Solution.

	Domain	Target	Function	Injective? (1-to-1)	Surjective? (onto)
1.1	$\mathbb{Z}$	$\mathbb{Z}$	$2x$	■	□
1.2	$\mathbb{R}$	$\mathbb{R}$	$2x$	■	■
1.3	$\mathbb{N}$	$\mathbb{N}$	x^2	■	□
1.4	Dunkin Donuts restaurants	GPS coordinates on Earth	Location of the front door	■	□
1.5	Buses traveling now on Comm. Ave	People now in MA	Driver of the bus	■	□
1.6	Every eye in this room	Every person in this room	Owner of the eye	□	■

Explanatory Notes Not every integer is divisible by two (so 1.1 is not surjective) but every real is twice another real (so 1.2 is surjective) 1.3: Injective, because *nonnegative* square roots are unique (would not be injective over $\mathbb{Z}$, because positive squares have both positive and negative square roots); not surjective, because not every natural number is a perfect square. 1.4: Injective, because every Dunkin has different coordinates for its front door. 1.5: Injective, because every bus has a different driver. 1.6 Not injective, because at least some people have two eyes; surjective because no one is without eyes in this room.

b) (9 points) For each of the following relations, state whether it is reflexive, whether it is symmetric, and whether it is transitive. Fill in the box if yes, and leave it empty if no.

Solution.

	Domain	Relation between x and y	Refl?	Symm?	Trans?
2.1	BU Courses	x is scheduled at same time slot as y	■	■	■
2.2	BU Courses	$\exists$ a student who is in both x and y	■	■	□
2.3	BU Students	x is taller than y	□	□	■
2.4	BU Students	$\exists$ a course that x and y are both taking	■	■	□
2.5	BU Students	x lives in the same room as y	■	■	■
2.6	All Humans	x has sent an email to y	□	□	□

Explanatory Notes 2.2 is not transitive, because it could be that there is a student Andrew in x and y and another student Bella in y and z, but because these are different students, there is no student in both x and z. Similarly, 2.4 is not transitive, because x and y could be in CS 131 and y and z could be in CS 112, but x and z might not have a course in common.

Problem 5.

Here is a truth table of a boolean function:

p	q	r	
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a) Write the CNF of this formula.

Solution. $(p + q + \overline{r}) \cdot (\overline{p} + q + r) \cdot (\overline{p} + q + \overline{r})$

b) Write the DNF of this formula.

Solution. $(\overline{p} \cdot \overline{q} \cdot \overline{r}) + (\overline{p} \cdot q \cdot \overline{r}) + (\overline{p} \cdot q \cdot r) + (p \cdot q \cdot \overline{r}) + (p \cdot q \cdot r)$

Problem 6.

Consider the English alphabet $\Sigma = \{a, b, \ldots z\}$. The set of any possible strings we can make from this alphabet, **excluding** the empty string ϵ, is denoted Σ^+.

The relation R on Σ^+ is defined as $R = \{(x, y) \mid x \text{ and } y \text{ have the same first letter}\}$.

a) Show that R is an equivalence relation.

Solution. We will begin by showing that R is reflexive. Consider that for any arbitrary $x \in \Sigma^+$, that x starts with the same letter as itself. Therefore, R is reflexive.

Suppose xRy for any $x, y \in \Sigma^+$. Then it must be that x starts with the same letter as y. Therefore, y starts with the same letter as x, so yRx, meaning R is symmetric.

Suppose xRy and yRz for any $x, y, z \in \Sigma^+$. Then x starts with the same letter as y which starts with the same letter as z, so x starts with the same letter as z, so xRz. Therefore, R is transitive.

Since R is reflexive, symmetric, and transitive, R is an equivalence relation.

b) How many equivalence classes does R have? Describe them using proper notation, but you do not have to explicitly list every single one.

Solution. There are 26 equivalence classes, one for each letter of the alphabet, i.e. $[a], [b], ...[z]$, where $[a]$ represents all the strings that start with the letter a, and so on.

Problem 7.

Prove the following using mathematical induction.

a) Prove for all $n \in \mathbb{N}$, that $2^n > n$

Solution. Base case: Let $n = 0$. $2^0 = 1 > 0$. Thus the base case holds.

Inductive step: Assume for some $k \in \mathbb{N}$, that $2^k > k$. Then

$$\begin{aligned}
2^{k+1} &= 2(2^k) \\
&= 2^k + 2^k \\
&> k + 2^k && \text{by the inductive hypothesis} \\
&\geq k + 1 && \text{because } 2^k \geq 1 \text{ for any positive } k
\end{aligned}$$

So $2^{k+1} > k + 1$, which is what we needed to prove for the inductive step.

b) Given the recurrence relation $u_n = 3u_{n-1} + 1$, where $u_1 = 1$, prove that for all integers $n \geq 1$, $u_n = \frac{3^n - 1}{2}$.

Solution. Base case: Let $n = 1$. $u_1 = 1$ and $\frac{3^1 - 1}{2} = 1$. Since $1 = 1$, the base case holds.

Inductive step: By the inductive hypothesis, assume $u_k = \frac{3^k - 1}{2}$ for some $k \geq 1$. Since $u_{k+1} = 3u_k + 1$ by definition, $u_{k+1} = 3u_k + 1 = 3(\frac{3^k - 1}{2}) + 1 = \frac{3^{k+1} - 3}{2} + 1 = \frac{3^{k+1} - 3}{2} + \frac{2}{2} = \frac{3^{k+1} - 1}{2}$, which is what we needed to prove for the inductive step.

Problem 8.

a) Consider the following partial proof in Lean.

```
example (a b c : Nat) : (a = b) → (a - b = c - c) := by
  intro h
  rw [h]
```

What is the proof state at the end of this partial proof?

Solution.

```
a b c : Nat
h : a = b
|- b - b = c - c
```

b) Consider the following partial proof in Lean.

```
example (a b c : Nat) : (a = b) → (c = a - b) → (a - b = c) := by
  intro h
  rw [h]
  intro h2
```

What is the proof state at the end of this partial proof?

Solution.

```
a b c : Nat
h : a = b
h2 : c = b - b
|- b - b = c
```

Problem 9.

Consider the following statement in Lean.

```
example (a b c : Nat) : (a = b + c) → (a - b = c)
```

and the following auxiliary theorems:

```
theorem Nat.sub_self (n : Nat) : n - n = 0
theorem Nat.zero_sub (n : Nat) : 0 - n = 0
theorem Nat.sub_zero (n : Nat) : n - 0 = n
theorem Nat.add_sub_cancel (a b : Nat) : (a + b) - b = a
theorem Nat.add_assoc (a b c : Nat) : a + b + c = a + (b + c)
theorem Nat.add_comm (a b : Nat) : a + b = b + a
theorem Nat.add_left_comm (a b c : Nat) : a + (b + c) = b + (a + c)
```

How can we prove the statement in Lean? (Multiple choice — choose one of the three options below.)

1.
```
by
rw [add_comm]
intro h
rw [h]
apply Nat.add_sub_cancel
```

2.
```
by
intro h
rw [h]
rw [add_comm b c]
rw [add_assoc]
rw [Nat.sub_self]
apply Nat.sub_zero
```

3.
```
by
intro h
rw [h]
rw [add_comm]
rw [Nat.sub_self]
apply Nat.sub_zero
```

Solution. Using 1.