FUNCTIONS AND RELATIONS

FRANK TSAI

Contents

1. Relations

Definition 1.1. An n-ary *relation* R on a set S can be encoded as a subset:

$$R \subseteq S^n$$

We write $R(a, \dots, z)$ whenever $(a, \dots, z) \in R$. Binary relations will be the main focus of this class. For these relations, it is customary to use infix notations. That is, we write aRb instead of $R(a, b)$.

Example 1.2. The substring relation $\sqsubseteq$ on $\{a, b\}^*$ is the subset

$$\{(\varepsilon, \varepsilon), (\varepsilon, a), \dots, (a, a), (a, ab), (a, ba), \dots\}$$

Example 1.3. The divisibility relation $\mid$ on $\mathbb{Z}$ is defined by

$$a \mid b \iff \exists c.\, b = ac$$

It is the subset

$$\{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid \exists c.\, b = ac\}$$

Example 1.4. The adjacency relation on a simple graph: two vertices u are v are adjacent if they are connected by an edge. It is the subset

$$\{(u, v) \in V \times V \mid (u, v) \in E \lor (v, u) \in E\}$$

Definition 1.5 (Reflexivity)**.** A binary relation R on a set S is *reflexive* if every element of S is related to itself by R.

$$\forall a.\, aRa$$

Example 1.6. The divisibility relation on $\mathbb{Z}$ is reflexive because every integer divides into itself once.

Definition 1.7 (Symmetry)**.** A binary relation R on a set S is *symmetric* if whenever a is related to b by R, then b is also related to a by R.

$$\forall a. \forall b.\, (aRb \Rightarrow bRa)$$

Date: October 15, 2023.

Example 1.8. The adjacency relation on a simple graph is symmetric. If a vertex u is adjacent to another vertex v, then v is also adjacent to u.

Definition 1.9 (Transitivity)**.** A binary relation R on a set S is *transitive* if for any three elements a, b, c of S, if aRb and bRc then aRc.

$$\forall a. \forall b. \forall c. \, (aRb \wedge bRc \Rightarrow aRc)$$

Example 1.10. The substring relation on $\{a, b\}^*$ is transitive. In fact, it is also reflexive, but it is not symmetric.

Definition 1.11 (Equivalence Relation)**.** A binary relation R on a set S is an *equivalence relation* if it is

(i) reflexive,
(ii) symmetric, and
(iii) transitive.

Proposition 1.12. *The congruence-modulo-2 relation on $\mathbb{Z}$ is defined by*

$$a \equiv b \mod 2 \iff 2 \mid (a - b)$$

It is an equivalence relation.

Proof. (Reflexivity). Let a be any integer. We need to prove that $a \equiv a \mod 2$. By definition, this is equivalent to proving $2 \mid (a - a)$, or equivalently, $2 \mid 0$. By definition again, this is equivalent to $\exists c. \, 0 = 2c$. Setting $c := 0$ yields $0 = 2 \cdot 0 = 0$ as desired.

(Symmetry). Let a, b be any integers. Assume that $a \equiv b \mod 2$. By definition, this hypothesis asserts that there's an integer c so that $a - b = 2c$. We need to prove $\exists k. \, b - a = 2k$. Setting $k := -c$ yields $b - a = -(a - b) = -2c = 2(-c)$ as desired.

(Transitivity). Let a, b, c be any integers. Assume that $a \equiv b \mod 2$ and that $b \equiv c \mod 2$. By definition, these two hypotheses assert that there are integers n, m so that $a - b = 2n$ and $c - b = 2m$. We need to show that $\exists c. \, a - c = 2c$. Setting $c := n - m$ yields $2(n - m) = 2n - 2m = a - b - (c - b) = a - b - c + b = a - c$ as desired. $\square$

Definition 1.13 (Antisymmetry)**.** A binary relation R on a set S is *antisymmetric* if for any two elements a, b of S, if aRb and bRa then $a = b$.

Example 1.14. The subset relation $\subseteq$ on $\mathcal{P}(S)$ is antisymmetric. Recall that two sets A and B are equal precisely when $A \subseteq B$ and $B \subseteq A$.

Remark 1.15. Antisymmetry does **not** imply **a**symmetry. For example, the indiscrete relation I on the singleton set $\{a\}$, defined as

$$I = \{(a, a)\}$$

is both antisymmetric and symmetric.

Definition 1.16 (Preorder)**.** A binary relation is a *preorder* if it is

(i) reflexive, and
(ii) transitive.

Definition 1.17 (Partial Order)**.** A *partial order* is a preorder that additionally satisfies antisymmetry.

Proposition 1.18. *The divisibility relation on $\mathbb{N}$ is a partial order.*

Proof. (Reflexivity): Exercise.

(Transitivity): Exercise. Hint: See Proposition 1.12.

(Antisymmetry): Let a, b be natural numbers so that $a \mid b$ and $b \mid a$. These hypotheses assert that there are natural numbers n, m so that $b = an$ and that $a = bm$. Thus, $b = (bm)n$. If $b = 0$, then since $a = bm = 0m = 0$, $a = b$ as desired. However, if $b \neq 0$, then $mn = 1$. Since n, m are natural numbers, $n = m = 1$. Thus, $a = b$ as desired. $\square$

Remark 1.19. Proposition 1.18 does not hold if we replace $\mathbb{N}$ with $\mathbb{Z}$ because $2 \mid -2$ and $-2 \mid 2$, but $2 \neq -2$. Although the divisibility relation on $\mathbb{Z}$ is not a partial order, it is a preorder.

2. Functions

3. Countable Sets and Uncountable Sets

Theorem 3.1. $\mathbb{N}^{\mathbb{N}}$ *is uncountable.*

Proof. Suppose that $\mathbb{N}^{\mathbb{N}}$ is countable, i.e., $\mathbb{N} \cong \mathbb{N}^{\mathbb{N}}$. A possible interpretation of this hypothesis is that every function $f : \mathbb{N} \to \mathbb{N}$ can be given a unique natural-number code. That is, there are functions

$$\mathsf{decode} : \mathbb{N} \to \mathbb{N}^{\mathbb{N}}$$
$$\mathsf{encode} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$$

that are mutual inverses. Consider the function

$$k : \mathbb{N} \to \mathbb{N}$$
$$k : n \mapsto \mathsf{decode}(n)(n) + 1$$

Given a code n, the function k decodes n, yielding a function $\mathbb{N} \to \mathbb{N}$, then evaluates that function at n, and finally adds 1 to the result.

The function k has a unique code given by $\mathsf{encode}(k)$. Now, let's evaluate k at its own code:

$$\begin{aligned} k(\mathsf{encode}(k)) &= \mathsf{decode}(\mathsf{encode}(k))(\mathsf{encode}(k)) + 1 \\ &= k(\mathsf{encode}(k)) + 1 \end{aligned}$$

This is a contradiction. $\square$

Theorem 3.1 tells us that some functions $f : \mathbb{N} \to \mathbb{N}$ are uncomputable: there are only countably many programs that one can write, but there are uncountably many endofunctions on $\mathbb{N}$. Thus, some of those functions do not have a corresponding program that computes it.