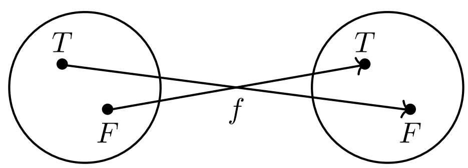
Assignment 0

CS 7480: Categories for PL, Fall 2025
Steven Holtzen and John M. Li

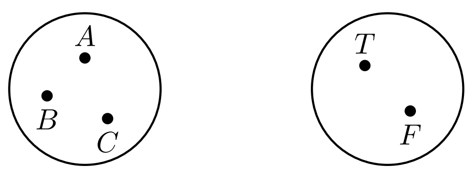
Due Monday, Sept 15 11:59PM EST

Problem 1 (Functions). A function $f : A \to B$ is a binary relation between inputs and outputs: $f \subseteq A \times B$. This relation satisfies a special property called *functionality*, which says that to each input a in A there is exactly one output b in B such that the pair (a, b) is in f. The common notation $f(a) = b$ then abbreviates $(a, b) \in f$, with the functionality of f ensuring that this notation does not lead to contradiction.

Part 1.1. A function can be depicted as a collection of arrows connecting each point in its domain to the corresponding output in its codomain. For instance, the following picture depicts the negation function on the two-element set of Booleans $\{T, F\}$.



Draw all functions between the following two finite sets:



Then, pick your favorite function and write it out as a set of input-output pairs.

Part 1.2. Let $f : \mathbb{R} \to \mathbb{R}$ be the squaring function $f(x) = x^2$. As a set of pairs,

$$f = \{(x, x^2) \mid x \in \mathbb{R}\}. \tag{1}$$

As with any such set, one can swap the order of elements in each pair to obtain a new set

$$g = \{(x^2, x) \mid x \in \mathbb{R}\}. \tag{2}$$

Is this new set g a function? What if x^2 were replaced by x^3?

Problem 2 (Equivalence Relations)**.** An *equivalence relation* $\approx$ on a set X is a binary relation $\approx \subseteq X \times X$, satisfying the following three conditions:

- (Reflexive) $x \approx x$ for each $x \in X$;
- (Transitive) $x \approx y$ and $y \approx z$ implies $x \approx z$ for every $x, y, z \in X$;
- (Symmetric) $x \approx y$ implies $y \approx x$ for every $x, y \in X$.

The *equivalence class for an element* x, written $[x]$, is the set of elements of X that are equivalent to x, i.e. $[x] = \{y \in X \mid x \approx y\}$. The set of all equivalence classes is written $X/_{\approx} = \{[x] \mid x \in X\}$, and is called the *quotient* of X by $\approx$.

Part 2.1. Show that the relation $p \subseteq X \times X/_{\approx}$ defined by $p = \{(x, [x]) \mid x \in X\}$ is functional, making p a function $X \to X/_{\approx}$.

Part 2.2. Let $\approx$ be the equivalence relation on $\mathbb{Z}$ defined by $i \approx j$ if and only if $i - j$ is even. Determine how many elements are in $\mathbb{Z}/_{\approx}$, and give an English description describing what the function $p : \mathbb{Z} \to \mathbb{Z}/_{\approx}$ does.

Problem 3 (STLC)**.** Recall the definition of the simply-typed λ-calculus (STLC). Terms, types, and contexts in STLC are formed from the following grammar:

$$\begin{aligned}
\mathsf{Type} \ni A, B &::= 1 \mid A \times B \mid A \to B \\
\mathsf{Term} \ni M, N &::= \langle\rangle \mid x \mid \langle M, N\rangle \mid \mathsf{proj}_1\ M \mid \mathsf{proj}_2\ M \mid \lambda x : A.M \mid MN \\
\mathsf{Ctx} \ni \Gamma &::= \bullet \mid \Gamma, x : A
\end{aligned}$$

where $\bullet$ denotes the empty context. For simplicity, we require that every variable in a context Γ is distinct. This grammar is accompanied by the following standard typing rules:

$$\frac{}{\Gamma \vdash \langle\rangle : 1}\ (T1) \qquad \frac{\Gamma(x) = A}{\Gamma \vdash x : A}\ (T2) \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N\rangle : A \times B}\ (T3) \qquad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \mathsf{proj}_1\ M : A}\ (T4)$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \mathsf{proj}_2\ M : B}\ (T5) \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}\ (T6) \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}\ (T7)$$

In the past, you may have seen the semantics of STLC given by a stepping relation. However, it is also possible to assign a semantics to STLC using an equational theory, defined inductively by the following rules [1]:

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M \equiv M : A}\ (1) \qquad \frac{\Gamma \vdash M \equiv N : A}{\Gamma \vdash N \equiv M : A}\ (2) \qquad \frac{\Gamma \vdash M \equiv N : A \quad \Gamma \vdash N \equiv O : A}{\Gamma \vdash M \equiv O : A}\ (3)$$

$$\frac{\Gamma \vdash M \equiv M' : A \quad \Gamma \vdash N \equiv N' : B}{\Gamma \vdash \langle M, N\rangle \equiv \langle M', N'\rangle : A \times B}\ (5)$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \mathsf{proj}_1\ \langle M, N\rangle \equiv M : A}\ (6) \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \mathsf{proj}_2\ \langle M, N\rangle \equiv N : B}\ (7)$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \langle \mathsf{proj}_1\ M, \mathsf{proj}_2\ M\rangle \equiv M : A}\ (8) \qquad \frac{\Gamma \vdash M : A \to B \quad x \notin \mathsf{dom}(\Gamma)}{\Gamma \vdash (\lambda x : A.Mx) \equiv M : A \to B}\ (9)$$

$$\frac{\Gamma \vdash M \equiv M' : A \to B \quad \Gamma \vdash N \equiv N' : A}{\Gamma \vdash MN \equiv M'N' : B}\ (10) \qquad \frac{\Gamma, x : A \vdash M \equiv N : B}{\Gamma \vdash (\lambda x : A.M) \equiv (\lambda x : A.N) : A \to B}\ (11)$$

$$\frac{\Gamma \vdash M : A \quad \Gamma, x : A \vdash N : B}{\Gamma \vdash (\lambda x : A.N)M \equiv N[M/x] : B}\ (12) \qquad \frac{\Gamma \vdash M : 1}{\Gamma \vdash M \equiv \langle\rangle : 1}\ (13)$$

In this exercise, we will review typing derivations and get practice working with the equational theory of STLC.

Part 3.1. Exhibit a derivation for the following STLC typing judgement:

$$\bullet \vdash \mathsf{proj}_1\ ((\lambda x : 1 \times 1.x)\langle\langle\rangle, \langle\rangle\rangle) : 1$$

Label each applied rule.

Part 3.2. Exhibit a derivation for the following STLC typing judgement:

$$y : 1 \vdash (\lambda f : 1 \to 1.\langle f\,\langle\rangle, y\rangle)((\lambda x : 1.\lambda z : 1.x)\langle\rangle) : 1 \times 1$$

Label each applied rule.

Part 3.3. Show that $\equiv$ forms an equivalence relation on STLC terms.

Part 3.4. Exhibit a derivation showing that $((\lambda\, x : \mathsf{Unit} \times (\mathsf{Unit} \to \mathsf{Unit}).\, \mathsf{proj}_1\, x)\ \langle\langle\rangle, \lambda\, y : \mathsf{Unit}.\, y\rangle) \equiv \langle\rangle$ using the equational rules. Label each applied rule. Give one other member of the equivalence class $[\langle\rangle]$.

Part 3.5. Show that the typing rules are *deterministic*, i.e., show that there is exactly one typing derivation tree $\Gamma \vdash M : A$ if M is well-typed according to a typing context Γ. Your proof should be by structural induction on syntax.

Part 3.6. Are the equational laws deterministic? If they are, prove it. If they aren't, give a counterexample.

References

[1] Eugenio Moggi. "Notions of computation and monads". In: *Information and computation* 93.1 (1991), pp. 55–92.