Turing Categories and Computability

Amittai Siavava

05/23/2024

Contents

Introduction

In this paper, we construct a turing category $\Bbbk$ and study the resulting implications on computability.

1 Preliminaries

1.1 Categories

Definition 1.1. A ***category*** $\mathscr{A}$ consists of:

1. A collection $\mathbf{ob}(\mathscr{A})$ of objects;
2. For each pair of objects $A, B \in \mathbf{ob}(\mathscr{A})$, a set $\mathscr{A}(A, B)$ of ***arrows*** or ***morphisms*** or ***maps*** from A to B;
3. For each $A, B, C \in \mathbf{ob}(\mathscr{A})$, a function

$$\circ_{A,B,C} : \mathscr{A}(B, C) \times \mathscr{A}(A, B) \to \mathscr{A}(A, C)$$
$$(f, g) \mapsto f \circ g$$

called ***composition***; where $(f \circ g)(x) = f(g(x))$ for all $x \in A$.

4. For each $A \in \mathbf{ob}(\mathscr{A})$, an arrow $\mathsf{id}_A \in \mathscr{A}(A, A)$ called the ***identity*** on A;

such that the following axioms hold:

1. associativity: for all $f \in \mathscr{A}(A,B)$, $g \in \mathscr{A}(B,C)$, and $h \in \mathscr{A}(C,D)$, $(h \circ g) \circ f = h \circ (g \circ f)$.

2. identity laws: for all $f \in \mathscr{A}(A,B)$, $f \circ \mathsf{id}_A = f = \mathsf{id}_B \circ f$.

Remark 1.2. As simplifications, we write:

(a) $A \in \mathscr{A}$ to mean $A \in \mathbf{ob}(\mathscr{A})$;

(b) $f : A \to B$ or $A \xrightarrow{f} B$ to mean $f \in \mathscr{A}(A,B)$;

(c) fg for $f \circ g$;

$\Diamond$

Examples 1.3. **1.** There is a category Set, where

(a) $\mathbf{ob}(\mathsf{Set})$ is the collection of all sets;

(b) $\mathsf{Set}(A,B)$ is the set of all functions from A to B;

(c) composition is ordinary function composition;

(d) the identity on A is the identity function on A.

2. There is a category Grp, where

(a) $\mathbf{ob}(\mathsf{Grp})$ is the collection of all groups;

(b) $\mathsf{Grp}(G,H)$ is the set of all group homomorphisms from G to H;

(c) composition is ordinary function composition;

(d) the identity on G is the identity homomorphism on G.

3. There is a category Top of topological space and continuous maps.

4. For each field k, there is a category Vect_k of vector spaces over k and linear maps between them.

$\Diamond$

Definition 1.4. A map $f : A \to B$ in a category $\mathscr{A}$ is an ***isomorphism*** if there exists a map $g : B \to A$ such that $fg = \mathsf{id}_A$ and $gf = \mathsf{id}_B$. Ee call g the ***inverse*** of f and write $f^{-1} = g$, and say that A and B are ***isomorphic*** if there exists an isomorphism between them.

Examples 1.5. **1.** In Set, isomorphisms are bijections.

2. In Grp and Ring, isomorphisms are group and ring isomorphisms respectively.

3. In Vect_k, isomorphisms are linear isomorphisms.

$\Diamond$

1.2 Restriction Categories

Definition 1.6. A ***restriction category*** is a category $\mathscr{A}$ with a ***restriction*** operation that assigns to each arrow $f : A \to B$ an arrow $\bar{f} : A \to A$ such that:

1. $\bar{f} \circ f = f$;
2. $\bar{f} \circ \bar{g} = \bar{g} \circ \bar{f}$ whenever $\mathbf{dom}\,(f) = \mathbf{dom}\,(g)$;
3. $\overline{f \circ \bar{g}} = \bar{g} \circ \bar{f}$ whenever $\mathbf{dom}\,(f) = \mathbf{dom}\,(g)$.
4. $\bar{g} \circ f = \bar{g} \circ f \circ \bar{g}$ whenever $\mathbf{dom}\,(f) = \mathbf{range}\,(g)$.

Remark 1.7. It follows from the definition that $\bar{f}$ is ***idempotent***. That is, $\bar{f} \circ \bar{f} = \bar{f}$. Furthermore, the operation $f \mapsto \bar{f}$ is also monotonic, with $\bar{\bar{f}} = \bar{f}$. ◊

Examples 1.8. Here are a few examples of restriction categories. [2]

1. All categories admit the trivial restriction operation that maps $f : A \to B$ to $\bar{f} = \mathsf{id}_A$.
2. The category Par of partial functions between sets admits a restriction operation that maps $f : A \rightharpoonup B$ to $\bar{f} = \mathsf{id}_{\mathbf{dom}(f)}$.

◊

2 Turing Categories

A Turing category is a cartesian restriction category $\mathscr{T}$ equipped with:

1. cartesian products — to pair (the codes of) data and programs;
2. a restriction structure representing the notion of partiality — for represent programs (morphisms) which do not necessarily halt;
3. and a ***Turing object*** A — to represent the "codes" of all programs. A Turing object is an object A such that for any $X, Y \in \mathscr{T}$, there is a universal application morphism $\tau_{X,Y} : A \times X \to Y$ that represents the application of a program (in A) to data (in X) to produce a result (in Y). [1]

Turing categories provide an abstract framework for computability: a "category with partiality" equipped with a "universal computer", whose programs and codes thereof constitute the objects of interest. [1]

2.1 Basic Properties of Turing Categories

Definition 2.1. Given two objects $A, B \in \mathscr{C}$, A is a ***retract*** of B if there exist morphisms $s : A \to B$ and $r : B \to A$ such that $r \circ s = \mathsf{id}_A$. s is called a ***section*** and r is called a ***retraction***.

$$\begin{tikzcd} A \arrow[r, bend left, "s"] & B \arrow[l, bend left, "r"] \end{tikzcd}$$

Lemma 2.2. In a Turing category $\mathscr{C}$ with a Turing object A, every object $B \in \mathscr{C}$ is a retract of A.

Examples 2.3. Here are some examples of Turing categories:

1. The classical recursion category $\mathscr{R}$, where objects are sets and morphisms are partial computable ("recursive") functions. Since Turing machine (or register-machine)–computable functions are exactly the partial computable functions, one may consider the codes of Turing machines corresponding to the partial computable functions as the objects of $\mathscr{R}$, with $\varphi_i : \mathbb{N} \to \mathbb{N}$ representing the machine with code i. [1] When $f = \varphi_e$, we say that e is a *code* for f.

Key properties of classical recursion include:

(a) The existence of a universal partial computable function Φ such that for each $e \in \mathbb{N}$,

$$\Phi(e, x_1, x_2, \ldots, x_n) = \varphi_e(x_1, x_2, \ldots, x_n).$$

(b) The **s-m-n Theorem**: There are computable and injective functions s^n_m for each $m, n > 0$ such that

$$\varphi_e(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n) = \varphi_{s^n_m(e, x_1, x_2, \ldots, x_m)}(y_1, y_2, \ldots, y_n).$$

Define

$$\begin{aligned} \bullet : \mathbb{N} \times \mathbb{N} &\to \mathbb{N} \\ (e, x) &\mapsto \varphi_e(x). \end{aligned}$$

Consider the category $\mathsf{Comp}(\mathbb{N})$ with the following properties:

(a) $\mathbf{ob}(\mathsf{Comp}(\mathbb{N})) = \{\mathbb{N}^i \mid i \in \mathbb{N}\}$;

(b) $f : N^k \to \mathbb{N}^m$ is an m-tuple of partial computable functions of k variables each.

$\Diamond$

Some of the key results in computability theory carry over to Turing categories, including the following.

Theorem 1. (smn) For $\varphi : A \to B$ partial computable, there exists a partial computable function $s : \mathbb{N} \to \mathbb{N}$ such that $\varphi(\langle i, n \rangle) = \varphi_{s(i)}(n)$ for all $x \in \mathbf{dom}(\varphi)$.

References

[1] J.R.B. Cockett and P.J.W. Hofstra, *Introduction to turing categories*, Annals of Pure and Applied Logic **156** (2008), no. 2, 183–209.

[2] Tom Leinster, *Basic category theory*, 2016.