Math 114C Notes

Mealaud Mokhtarzad

This set of notes is very informal and tries its best to simplify often hard to digest ideas. Hopefully it's useful in your learning of the material.

FINISH: put figures in right places

Contents

1 Models of computation

test

2 Effective enumerations and undecidability

2.1 Recursive partial functions

In the previous section we talked about different models of computation (hence its title) and generally different ways to think about computability in terms of, well, computing (with computers). Now, we come across a new term: recursive. What recursive means is that the more pure-mathematical way to define computability without machines necessarily. Now let's get into our definition.

Definition 2.1. We define the `class` $\mathcal{R}$ `of recursive partial functions` to be the $\subseteq$-smallest collection of partial functions $f\colon \mathbb{N}^n \rightharpoonup \mathbb{N}$ such that

(1) $\mathcal{R}$ contains the zero function $\underline{0}$, the successor function, and the projection functions U^n_k.

(2) $\mathcal{R}$ is closed under definition by substitution, definition by recursion, and definition by minimalization.

We're also going to give a name to the class of functions we were working with before: the URM computable functions.

Definition 2.2. We define the `class` $\mathcal{C}$ `of URM computable partial functions` to be the class of partial functions $g\colon \mathbb{N}^n \rightharpoonup \mathbb{N}$ such that there exists a URM program P with $g = f_P^{(n)}$.

Our first big result of these notes indeed relating $\mathcal{R}$ and $\mathcal{C}$. In particular,

Theorem 2.3. *A partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is recursive if and only if it is URM-computable (i.e., $\mathcal{R} = \mathcal{C}$).*

We actually have one direction of this already (the proof is very short and follows mostly by definition).

Lemma 2.4. $\mathcal{R} \subseteq \mathcal{C}$.

Proof. Since $\mathcal{C}$ is closed under the same constraints that define $\mathcal{R}$, since $\mathcal{R}$ is the smallest such set, it is necessarily true that $\mathcal{R} \subseteq \mathcal{C}$. ■

We will prove the other direction in due time.

3 Computably enumerable sets

3.1 Partially decidable relations

In this section, we're going to use the Kleene T predicate a lot, so we're going to restate it here.

Theorem 3.1 (Kleene T predicate)**.** *For each positive integer n, there is a recursive $(n+2)$-ary relation $T_n(e, \mathbf{x}, y)$ and a computable totla function $U\colon \mathbb{N} \to \mathbb{N}$ which satisfy the following condition: for every $e \in \mathbb{N}$ and every $\mathbf{x} \in \mathbb{N}^n$,*

$$\varphi_e^{(n)}(\mathbf{x}) = U(\mu y[T_n(e, \mathbf{x}, y)]).$$

Remark. Let recall what the Kleene T predicate does: it takes a program code e, an input for the program $\mathbf{x}$, and a computation history y, and decides whether y is a computation history for the program defined by e with input $\mathbf{x}$ that halts. Recall that the Kleene T predicate is indeed decidable.

Definition 3.2. An n-ary relation $R \subseteq \mathbb{N}^n$ is `partially decidable` if and only if the partial function $f\colon \mathbb{N}^n \rightharpoonup \mathbb{N}$ defined by

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } R(\mathbf{x}) \text{ holds,} \\ \uparrow & \text{otherwise,} \end{cases}$$

is computable.

The function f above is called the `partial characteristic function` of R and any algorithm computing f is called a `partial decision procedure` for R.

Remark. Partially decidable has other names too; semi-decidable, semi-computable, and partially solvable are other common names. We will try to use partially decidable mainly though. I don't like the other names lmao.

Do we have any examples of partially decidable relations? We do! In fact, we have many, many such relations. Here's a canonical one, though.

Example 3.3. The (unary diagonal) halting problem (i.e., whether $\varphi_x(x)$ halts) is partially decidable.

Proof. We can define its partial characteristic function $f\colon \mathbb{N} \rightharpoonup \mathbb{N}$ as the following:

$$f(x) = 1 + 0 \cdot \mu y[T(x, x, y)].$$

Observe that f is computable since it is the minimalization of a decidable relation. Then, notice that if $\varphi_x(x)\downarrow$, then there does exist a halting computation history y and our minimalization will eventually return a value, so $f(x) = 1$ if $\varphi_x(x)\downarrow$. If $\varphi_x(x)\uparrow$, then there does not exist a halting computation history y and our minimalization never ends, so $f(x)\uparrow$. ■

Remark. This trick of multiplying a minimalization by 0 to encode divergence of a computation is SUPER useful. Be aware of its existence!

Example 3.4. Any decidable relation is partially decidable.

Proof. Suppose $R \subseteq \mathbb{N}^n$ is decidable. Then, $f\colon \mathbb{N}^n \to \mathbb{N}$ defined by

$$f(\mathbf{x}) = 1 + 0 \cdot \mu y[R(\mathbf{x}) \wedge y = y].$$

If $R(\mathbf{x})$ holds, then the minimalization ends and we get $f(\mathbf{x}) = 1$ as desired. IF $R(\mathbf{x})$ does not hold, then our minimalization never ends and the computation diverges, so $f(\mathbf{x})\uparrow$. ■

Remark. I told you the multiplying by 0 trick would be useful!

We know that there exist decidable relations, but do there exist not partially decidable relations? Yes, there do! We actually have another 'simple' example: the negation of the (unary diagonal) Halting Problem.

Example 3.5. Let $Q(x)$ hold if and only if $\varphi_x(x)\uparrow$. The relation Q is not partially decidable.

Proof. Suppose towards a contradiction that Q is partially decidable. In particular, suppose that the the partial characteristic function of Q, $f_Q\colon \mathbb{N} \rightharpoonup \mathbb{N}$, is computable. Morever, let $f_H\colon \mathbb{N} \rightharpoonup \mathbb{N}$ be the (computable) partial characteristic function of the (unary diagonal) Halting Problem. Now, by running $f_Q(x)$ and $f_H(x)$ at the same time, we may decide whether $\varphi_x(x)$ halts (as we will be able to decide if it does or if it does not based on the ouput of $f_H(x)$ and $f_Q(x)$, respectively). Since the Halting Problem is undecidable we have a contradiction and thus Q must not have been partially decidable. ■