Math 29: Simple Sets

April 25th, 2022

1 Immune Sets

Last time, we discussed the creative sets in an attempt to address Post's Problem. I.e., we are trying to find a c.e. set which does not give us enough information to solve the halting problem. As we saw, creative sets fail to achieve this. This leads us to look elsewhere.

A set is **immune** if it is infinite, but contains no infinite c.e. subset. A set is **simple** if it is c.e. and its complement is simple. Notice that, while we don't explicitly require a simple set to be infinite, all simple sets will be: If A were finite, then it is computable, and therefore its complement is cofinite and computable. However, this would make its complement an infinite c.e. set, so it cannot be immune.

In general, we can use this idea to prove that simple sets are different than what we have seen before.

Lemma 1. *Let S be a simple set.*

- *S is not computable.*
- *S is not creative.*
- *There is no total computable function f such that $f(e) \in S$ if and only if $e \in K$.*

Proof:

- If S were computable, its complement would be as well. But then its complement would be infinite and c.e., and therefore not immune.
- By Homework 4 Question 8, productive sets are not immune. Therefore, simple sets are not creative.

Proof: Suppose S is c.e. and there is f as stated in the lemma. By the s-m-n theorem, there is a total computable injection s such that $\varphi_{s(n)}(k) = \varphi_n(f(k))$. Then for all e, $W_{s(e)} = \{n : f(n) \in W_e\} = f^{-1}(W_e)$. We claim $f \circ s$ is a productive function for S^c.

Suppose $W_e \subseteq S^c$. Then $W_{s(e)} = f^{-1}(W_e) \subseteq K^c$. However, it cannot be the case that $\varphi_{s(e)}(s(e)) \downarrow$. If it did, then $s(e) \in K$ and $f(s(e)) \in S$. But, we also have $s(e) \in W_{s(e)} = f^{-1}(W_e)$. That means that there is $k \in W_e$ such that $f(s(e)) = k$. But since $W_e \subseteq S^c$, this is a contradiction. Therefore, $\varphi_{s(e)}(s(e)) \uparrow$, and it follows that $f(s(e)) \in S^c$. Furthermore, $\varphi_e(f(s(e))) = \varphi_{s(e)}(s(e)) \uparrow$, so $f(s(e)) \notin W_e$ as desired.

To summarize, if there is a function f with the assumed properties, then S is creative. Now suppose S is simple. It is not creative by the previous point, so by contrapositive such an f cannot exist for f.

Notice that in the proof for the final bullet point above, we did not use the fact that S was simple or that S^c is immune. This means it is true for arbitrary c.e. sets that such a function exists only if the set is creative.

Something one should always be careful about when introducing a mathematical definition is to make sure the definition is not vacuous. By the above, if they exist, the simple sets are outside of the scope of what we've seen before. They can't be computable, but they also cannot be essentially the same as K. But, we first need to prove that simple sets exist.

Theorem 2. *There exists a simple set S.*

Proof: We shall build a uniform sequence of finite c.e. sets S_t, and we will set $S = \bigcup_{t\in\omega} S_t$. We will meet all of the following requirements:

$$R_i : \ W_i \text{ infinite } \implies W_i \cap S \neq \emptyset$$

Notice that if we can meet these requirements, S will coimmune as no infinite c.e. set will be entirely contained within its complement

We start by defining $S_0 = \emptyset$. Given S_t, look for the least $e < t$ such that $W_{e,t} \cap S_t = \emptyset$ and there is some $t > x > 2e$ such that $x \in W_{e,t}$. If we find such an e, choose the least corresponding x and set $S_{t+1} = S_t \cup \{x\}$. If there is no such e, let $S_{t+1} = S_t$. Because time-bounded computation is all computable, and there are finitely many e and x to check at each stage, this process is computable. Continue to stage $t+2$. Set $S = \bigcup_{t\in\omega} S_t$, and this is a c.e. set.

Proof: (Cont.) All that is left to do is to check that we have met our requirements and verify that our set is coinfinite. (That is, its complement is infinite.) The latter is not hard to see, as $[0, 1, \ldots, 2n+1] \cap S$ contains at most n elements for all n because we always choose $x > 2e$ and we act on R_e at most once.

Finally, we argue that R_i is met for all i. If W_i is finite, then we succeed automatically. If W_i is infinite, then there will be t large enough so that $t > i$ and there is some $t > x > 2i$ such that $\varphi_{i,t}(x) \downarrow$. Then eventually, either $S_{t+1} \cap W_i \neq \emptyset$ at some point, or i will eventually be the least index ready to act, at which point it does. In both cases, we get $S \cap W_i \neq \emptyset$ as necessary.

Thus, simple sets exist. Furthermore, we can't use a total, computable function to reduce the halting problem to a simple set. However, as we shall see, we still have not satisfied Post's Problem.

A simple set S is **effectively simple** if there is a computable function f such that whenever $W_e \subseteq S^c$, $f(e) \downarrow$ and W_e contains no more than $f(e)$ elements. Notice that the above set is effectively simple with $f(n) = 2n + 1$: If there were $x \in W_e$ with $x > 2e$, then the least one would be added to S when it was R_e's turn to act.

Lemma 3. *There is an effectively simple set with $f(n) = n$. This is called the **Canonical Simple Set**.*

Proof: We will use the same requirements R_i. As we have argued above, if we meet them, then the resulting set will be simple. Therefore we just need to build our set and prove that it is effectively simple as witnessed by the identity.

Let $S_0 = \emptyset$. Given S_t, let s_i^t represent the i-th element of S_t^c in increasing order. In other words, $S_t^c = \{s_t^0 < s_t^1 < \ldots\}$. Find the least $e < t$ such that $W_{e,t} \cap S_t = \emptyset$ and there is some $t > i > e$ with $s_i^t \in W_{e,t}$. Let $S_{t+1} = S_t \cup \{s_i^t\}$ for the least such i. If there is no such e and i, let $S_{t+1} = S_t$.

Proof: (Cont.) Let $S = \bigcup_{t\in\omega} S_t$. Suppose $W_e \subseteq S^c$. We need to show that $|W_e| \leq f(e)$. We act on requirement R_e at most once, so eventually there is some t large enough that $s_i^t = s_i$. In other words the compliment of S stabilizes to its correct value below any n by some stage m. If $|W_e| > e$, then for some $i \geq e$ we have $s_i^m = s_i \in W_e$. But this is a contradiction, as we would then have put s_i into S.

It turns out that, while we can't use a single injective function, in general there are simple sets which are able to solve the halting problem.

1.1 Domination

A function g **dominates** f is, for all but finitely many n, either $f(n)\uparrow$ or $f(n)\downarrow < g(n)\downarrow$. That is, g is larger than f almost everywhere.

Notice that how big a function is relates to how much information it contains. Consider the function

$$f(e) = \begin{cases} s & s \text{ is the least number such that } \varphi_{e,s}(e)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Then suppose total g dominates f. We can then use f to compute K as follows: given e, check $g(e)$. Run $\varphi_{e,g(e)}(e)$. If it converges, then $e \in K$. If it doesn't, then by definition $\varphi_e(e)\uparrow$, so $e \notin K$. Thus we can use any total function which dominates f to compute K.

We can construct a simple set which can compute a total function which dominates every partial computable function. In particular, it can dominate f above, and therefore compute K.

Lemma 4. *There is a simple set S such that the principle function of its complement dominates every partial computable function.*

Proof: Construct the canonical simple set, but in addition to the s_i^t's enumerated, enumerate a_n^t into S_{t+1} whenever $\varphi_{e,t}(n)\downarrow \geq s_n^t$ and $e \leq n$. Then s_n^t will only be enumerated by $e \leq n$ by construction, and there are only finitely many such e's and n's. Thus s_n still stabilises, but $s_k > \varphi_e(k)$ for all $k \geq e$.

In other words, ensure that S^c contains only the maximum along the diagonal of $\varphi_e(n)$.

Later, we will see that all effectively simple sets can solve the halting problem. However, it turns out that not all simple sets can! In fact, every noncomputable c.e. set computes a simple set which is not effectively simple. This will use a technique known as **permitting**, which we have somewhat already seen.

In a permitting construction, we generally are trying to compute something from a c.e. set. We will set up our construction so that seeing an element being enumerated gives us **permission** to perform some operation on our construction. Then the c.e. set will be able to compute the constructed set by letting us run the construction up to the listed stage.