Math 29: Creative Sets

April 22nd, 2022

1 Productive Sets

We'd like to see some more examples of noncomputable c.e. sets, in furtherance of exploring Post's Problem. Notice that if we are to build X c.e. but not computable, then it suffices to ensure that X^c is not c.e. (We proved on the homework that X is computable if and only if it and its complement are c.e.) In other words, for each e, we need to ensure that X^c is not equal to W_e. One way to do this would be to ensure that X^c is not a subset of W_e.

We say that a set P is **productive** if it has a **productive function**. A productive function is a (partial) computable function ψ such that, whenever $W_e \subseteq P$, $\psi(e)\downarrow$ and $\psi(e) \in P \setminus W_e$. That is, a productive function is able to produce a witness to the fact that $P \neq W_e$ whenever $W_e \subseteq P$. Then it is immediate that productive sets are not c.e., so finding a c.e. set whose complement is productive will necessarily be a noncomputable c.e. set.

A set C is said to be creative if it is c.e. and its complement is productive. It is a fair question to ask if creative sets even exist. In fact, they do, and we have already seen one.

Lemma 1. *K is creative.*

Proof: We need to give a productive function for K^c. That is, a computable function ψ such that whenever $W_e \subseteq K^c$, we have that $\psi(e) \downarrow\in K^c \setminus W_e$. Let ψ be the identity function, i.e. $\psi(n) = n$ for all n.

Firstly, $\psi(e)\downarrow$ for all e. Now suppose $W_e \subseteq K^c$. Then $e \notin W_e$: if it were, then $\varphi_e(e)\downarrow$ since $W_e = dom(\varphi_e)$. But this would imply $e \in K$, contradicting that $W_e \subseteq K^c$. Thus $e \notin W_e$, which means $\varphi_e(e)\uparrow$ and thus $e \in K^c$. As e was arbitrary, ψ is a productive function for K^c.

While the definition of productive sets only requires a partial computable function, we can prove that a stronger condition is true.

Lemma 2. *If P is productive, then it has an injective, total computable productive function.*

Proof: We first create a total productive function q. Let ψ be a productive function for P. By the s-m-n theorem, there is a total computable function $s : \omega \to \omega$ such that

$$\varphi_{s(x)}(n) = \begin{cases} \varphi_x(y) & \text{if } \psi(x) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Then notice if $\psi(x) \uparrow$, then $W_{s(x)} = \emptyset$. Therefore $W_{s(x)} \subseteq P$, so $\psi(s(x)) \downarrow$. Therefore, define $q(x)$ to be the first of $\psi(x)$ and $\psi(s(x))$ to converge. (If $\psi(x) \downarrow$, then $\psi(s(x))$ need not necessarily converge, so we cannot simply use that.)

We now create an injective productive function p from q. By the s-m-n theorem, there is a total computable function h such that

$$\varphi_{h(x)}(n) = \begin{cases} 1 & n = q(x) \\ \varphi_x(n) & \text{otherwise} \end{cases}$$

Then $W_{h(x)} = W_x \cup \{q(x)\}$, $W_{h(h(x))} = W_x \cup \{q(x), q(h(x))\}$, $W_{h(h(h(x)))} = W_x \cup \{q(x), q(h(x)), q(h(h(x)))\}$, etc. Now define $p(0) = q(0)$. To define $p(n+1)$, list out the set

$$\{q(n+1), q(h(n+1)), q(h(h(n+1))), \dots, q(h^k(n+1)), \dots\}$$

There are two cases. Either we will see some $q(h^k(n+1))$ such that it does not appear in $p(0), \dots, p(n)$, in which we set $p(n+1)$ to the first one we see. Or, if we never see this, we will eventually see a repetition, i.e. the set is finite because $q(h^k(n+1)) = q(h^m(n+1))$ for some $k \neq m$. In this case, we set $p(n+1)$ equal to the least number not appearing in $p(0), \dots, p(n)$. Then this construction ensures that p is total, computable, and injective.

Finally, we need to justify that p is a productive function for P. If $p(x) = q(h^{k+1}(x))$ by the first case, then $W_x \subseteq W_{h^k(x)} \subseteq P$ for some k, and therefore $q(h^k(x)) \in P \setminus W_{h^k(x)} \supseteq P \setminus W_x$. If the value of $p(x)$ is set by the second case, then it must be the case that $W_x \not\subseteq P$: If it were, then by induction $W_{h^k(x)} \subseteq P$ for all k, but for some specific e we would have $q(h^e(x)) \in W_{h^e(x)}$, contradicting that q is a productive function for P.

Of course, this is not particularly helpful for addressing Post's Problem: we wanted to find a problem that couldn't be used to solve the halting problem, not the halting problem again. Can we find some other creative set which is weaker? It turns out that the answer is no: a set is creative if and only if it solving it solves the halting problem. This is a result due to John Myhill.

Theorem 3. *Let P be any productive set. Then there is an injective, total computable function f such that $x \in K^c$ if and only if $f(x) \in P$.*

Proof: By the above lemma, let p be an injective, total productive function for P. Consider the computable function

$$\varphi_e(x, y, n) = \begin{cases} 0 & \varphi_y(y) \downarrow \text{ and } n = p(x) \\ \uparrow & \text{otherwise} \end{cases}$$

By the s-m-n theorem, there is a total computable s such that $\varphi_{s(x,y)}(n)\varphi_e(x, y, n)$. Then $W_{s(x,y)} = \{p(x)\}$ if $y \in K$ and $\emptyset$ otherwise. By the recursion theorem with parameters, there is a total, injective, computable function t such that

$$W_{t(y)} = W_{s(t(y),y)} = \begin{cases} \{p(t(y)) & y \in K \\ \emptyset & \text{otherwise} \end{cases}$$

Now suppose $y \in K$. Then $W_{t(y)} = \{p(t(y))\}$, so $W_{t(y)}$ cannot be a subset of P: this would contradict the fact that p is productive for P. Therefore, $p(t(y)) \in P^c$.

Conversely, suppose $y \notin K$. Then $W_{t(y)} = \emptyset$, so $W_{t(y)} \in P$ and therefore $p(t(y)) \in P \backslash W_{t(y)} = P$. Thus $p \circ t$ is a function with the desired property. (It is total and injective as the composition of total, injective functions.)

Thus we have been able to reduce the problem of figuring out if something is in K by figuring out if something is not in P^c. To determine membership in K, apply $p \circ t$ then check whether or not it is an element of P^c. Showing that we can solve one problem via simplifying it to a second problem is an example of a **reducibility**. We'll see some formal definitions of various reducibilities after the midterm, but this is a common idea in mathematics: we solve new problems by showing that the answer must relate to that of a question we have already solved.