Math 29: Arslanov's Completeness Criterion

May 20th, 2022

1 Complete C.e. Sets

Recall that a c.e. set is complete if it computes every c.e. set. K is the prototypical complete set, but we have seen many others. We proved that creative sets are complete, and in fact 1-complete. We proved that, while not 1-complete, effectively simple sets are still complete. However, these are not all of the complete sets.

A total function f is **fixed point-free** if $W_{f(x)} \neq W_x$ for all x. Notice that this seems to be a slightly stronger requirement than being fixed point-free under the previous notion, i.e. $\phi_{f(x)} \neq \phi_x$ for all x. However, one can prove that these are essentially the same thing.

Lemma 1. *There is a fixed point-free function $f \leq_T X$ if and only if there is $g \leq_T X$ such that g is fixed point-free in the previous sense, i.e. $\varphi_{g(x)} \neq \varphi_x$.*

Proof. Homework 7 Question 4. □

Fixed point-free functions turn out to be the key to describing the complete sets.

Theorem 2. *(Arslanov's Completeness Criterion) A c.e. set X is complete if and only if it computes a fixed point-free function*

Proof: First, suppose that X is complete. Recall that $K \leq_1 Nem = \{e : W_e \neq \emptyset\}$ by the Index Set Theorem. Because $Nem \leq_1 K$ as it is c.e., $K \equiv_T Nem \equiv_T X$. Therefore the following function f obtained is X-computable:

$$W_{f(x)} = \begin{cases} \emptyset & x \in Nem \\ \{0\} & \text{otherwise} \end{cases}$$

It is easy to check that f is fixed point-free, so X computes a fixed point-free function.

Proof: (Cont.) Suppose that X computes a fixed point-free function f. Because X is c.e., $X \leq_T \emptyset'$, so f is limit computable via some computable $g(n, s)$ by the Limit Lemma.

Let $\{K_s\}_{s\in\omega}$ be an enumeration of K, and define the function $\theta(x)$ to be the least s such that $x \in K_s$ if $x \in K$, and $\theta(x) \uparrow$ otherwise. Using the recursion theorem with parameters, let $h(x)$ be such that

$$W_{h(x)} = \begin{cases} W_{g(h(x),\theta(x))} & x \in K \\ \emptyset & \text{otherwise} \end{cases}$$

That is, $W_{h(x)}$ waits for x to enter K, then enumerates the $g(h(x), \theta(x))$-th c.e. set if it does.

Now let m be a modulus for f. Then by the definition of a modulus we have $g(h(x), t) = f(h(x))$ for all $t \geq m(h(x))$. Thus, if $\theta(x) \downarrow > m(h(x))$, we have $f(h(x)) = g(h(x), \theta(x))$, and therefore $W_{h(x)} = W_{f(h(x))}$. But this is a contradiction of the fact that f is fixed point-free, so either $\theta(x) \uparrow$ (and thus $x \notin K$), or $\theta(x) \leq m(h(x))$. That means we can compute whether or not x is in K by checking if it is in $K_{m(h(x))}$. (This is similar to the proof of the fact that effectively simple sets compute K.)

Lastly, recall that we proved last time that if $f \leq_T X$ and X is c.e., then X computes a modulus for f. Therefore X can compute some m as above, and thus can compute K. Therefore, X is complete.

Notice that, along with the homework problem, this proves an incredible strengthening of the recursion theorem: if f is a total function computable from a non-complete c.e. set, then f has a fixed point! For example, the principle function of any non-complete c.e. set has a fixed point.

There are some generalizations of this theorem. While functions which can compute the halting problem will not necessarily have a fixed point, we can generalize the idea of a fixed point. An **almost fixed point** for a function f is a number e such that $W_e =^* W_{f(e)}$, where $=^*$ means that the sets agree on all but finitely many n.

Theorem 3. *If $f \leq_T \emptyset'$, then f has an almost fixed point.*

Proof: Suppose that $f \leq_T \emptyset'$. Then f is limit computable by the limit lemma, so there is a computable function g such that $f(n) = \lim_{s\to\infty} g(n,s)$. By the s-m-n theorem, there is a total computable function $\theta(x)$ such that

$$W_{\theta(x)} = \bigcup_{s\in\omega} W_{g(x,s),s}$$

That is, $\varphi_{\theta(x)}$ is the machine which starts enumerating each of the $W_{g(x,s),s}$-es for each s.

Then by the recursion theorem, θ has a fixed point e such that

$$W_e = W_{\theta(e)} = \bigcup_{s\in\omega} W_{g(e,s),s}$$

Now we claim that $W_e =^* W_{f(e)}$. Because f is limit computable, there is some k at which $g(e,t) = f(e)$ for all $t \geq k$. Then $W_{g(e,t),t} = W_{f(e),t}$ for all such t, so all of the places where $W_{f(e)}$ and $W_{\theta(e)} = W_e$ differ are, at worst, $\bigcup_{s<k} W_{g(e,s),s}$. This is a finite union of finite sets, and thus finite.

Therefore $W_e =^* W_{f(e)}$, so f has an almost fixed point.

This opens up a new question: The Arslanov Completeness Criterion extended the recursion theorem by proving that many more functions than the computable ones have fixed points. We proved above a recursion-like theorem in terms of almost fixed points: every jump-computable function has an almost fixed points. Can we extend this to more functions.

We won't prove this, but it turns out that we can. If X is c.e. in $\emptyset'$, i.e. is the domain of some Φ_e^X, then it can compute a function which has no almost fixed points if and only if it can compute $\emptyset''$. Recall that the proof that K is complete relativizes, so $\emptyset''$ is complete for the sets which are c.e. in $\emptyset'$. This shows the relationship with Arslanov's Completeness Criterion.

One final extension of this result is about **Turing fixed points.** A Turing fixed point for a function f is an e such that $W_e = W_{f(e)}$. One can show that every $\emptyset''$-computable function can compute a Turing fixed point.