Math 29: The Limit Lemma

May 18th, 2022

1 Limit Lemma

The **limit** $\lim_{n\to\infty} f(n)$ of a function f from the naturals to the naturals exists if and only if the function is eventually constant. This coincides with the usual ϵ definition of limits because once we choose $\epsilon < 1$ and the N such that the difference between $f(n)$ and $f(m)$ is no larger than ϵ for $n, m \geq N$, the function can do nothing but stay at the value of $f(N)$.

$f : \omega \to \omega$ is **limit computable** if there is a computable function $g(n, s)$ such that $f(n)$ equals the limit along s of g for all n. In other words,

$$f(n) = \lim_{s\to\infty} g(n, s)$$

This means that, for a given n, the value of g can change finitely many times as s increases, but must stop and remain constant at some point. We think of this as being some computable process which is guessing at the final, correct value for $f(n)$: it makes finitely many mistakes in which it keeps getting different answers, but it eventually settles on the correct answer.

Notice that, in general, not all limit computable functions are computable. For example, every c.e. set is limit computable: Given W_e, define $g_e(n, s)$ to be 0 if $\phi_{e,s}(n) \uparrow$, and 1 if $\varphi_{e,s}(n) \downarrow$. This will change at most one time, so the limit exists for all n and is either 0 or 1. Finally, it is not hard to see that $\chi_{W_e}(n) = \lim_{s\to\infty} g_e(n, s)$. Thus every c.e. set is limit computable. This includes K itself: there is a way to approximate the halting problem which gets more and more accurate as time increases, but there is no computable way to know how accurate it is at any fixed time.

Notice that c.e. sets are limit computable in a very specific way: the approximation always starts at 0, and it changes at most once. Because of this, we can construct a limit computable set which is not c.e.

Lemma 1. *There is a set B which is limit computable but not c.e.*

Proof: Define $f(e,s)$ to be 1 if $\varphi_{e,s}(e)\uparrow$ and 0 if $\varphi_{e,s}(e)\downarrow$, and let $\chi_B(e) = \lim_{s\to\infty} f(e,s)$. This exists because each $f(e,s)$ changes at most once when e increases. However, notice that $B \neq W_e$ for any e, as $B(e) = 1 - \chi_{W_e}(e)$ for all e.

Notice that the above set is exactly the complement of K, and thus clearly computable from K. In fact, by the same technique, the complement of any c.e. set is also limit computable.

Notice that every example we have seen is computable from the jump. It turns out that this is not a coincidence.

Lemma 2. *(The Limit Lemma) A function f is limit computable if and only if $f \leq_T \emptyset'$.*

To prove this lemma, we return to the idea of domination. That is, functions which grow fast enough are capable of computing various sets. If $f(n) = \lim_{s\to\infty} g(n,s)$, then a **modulus of convergence** for g is a function m such that $g(n,s) = f(n)$ for all $s \geq m(n)$. That is, $m(n)$ tells us a sufficient amount of time to wait to get the correct value of f, so we can use m and g to compute f. The least modulus is the function which outputs the minimal such s for all n.

Notice that, if we have *some* modulus, we can compute the least modulus: Just check the finitely many s values below $m(n)$, and pick the smallest one that matches $g(n, m(n))$ and doesn't change in between.

Lemma 3. *If X is c.e. and $f \leq_T X$, then f is limit computable. Furthermore, X computes a modulus for f.*

Proof: Because $f \leq_T X$, let Φ_e^X. Let k be such that $X = W_k$. Define $g(n,s)$ via

$$g(n,s) = \begin{cases} \Phi_{e,s}^{W_{k,s}}(n) & \text{if } \Phi_{e,s}^{W_{k,s}}\downarrow \\ 0 & \text{otherwise} \end{cases}$$

Then g is computable because time bounded computation is always computable, and thus time-bounded approximation of W_k is as well. Furthermore, $f(n) = \lim_{s\to\infty} g(n,s)$, as $\Phi_e^X(n) = f(n)$, and after large enough s, $\Phi_{e,s}^{W_{k,s}}(n)$ converges to the correct value because of the finite use principle. Thus f is limit computable.

Proof: Define $m(n)$ to be the least s such that there is a $x < s$ with $W_{k,s}$ correct up to x, i.e. $W_{k,s} \upharpoonright x = W_k \upharpoonright x$, and $\Phi_{e,s}^{W_{k,s} \upharpoonright x}(n) \downarrow$. This guarantees that the above computation matches $\Phi_e^X(n)$ and never changes, so $g(n, s)$ is the same value for all larger s. Therefore $m(n)$ is a modulus for f, and it is computable from X because X can tell when the approximations are correct.

We are now ready to prove the limit lemma.

Lemma 4. *(The Limit Lemma) A function f is limit computable if and only if $f \leq_T \emptyset'$.*

Proof: Suppose $f \leq_T \emptyset'$. As $\emptyset' = K$ is c.e., f is limit computable with modulus computable from the jump by the previous lemma.

Suppose f is limit computable via g, i.e. $f(n) = \lim_{s\to\infty} g(n, s)$ for all n. Consider the following set:

$$G = \{\langle n, s\rangle : \exists t \geq s \text{ with } g(n, t) \neq g(n, t + 1)\}$$

That is, the set of all $\langle n, s\rangle$ such that $g(n, s)$ has not stabilized yet. Note that this set is c.e., as we can start running $g(n, t)$ for $t \geq s$ until we see a change. Because $\emptyset'$ is complete for the c.e. sets, it can compute G. Furthermore, G, and hence the jump, can compute the least modulus simply by finding the least $\langle n, s\rangle$ which is not in G, as if $\langle n, s\rangle \in G$, then $\langle n, t\rangle \in G$ for all $t \leq s$.

Thus, the jump can compute f using the least modulus.

One can check that the limit lemma relativizes: $f \leq_T X'$ if and only if f is limit X-computable.

This characterizes the sets computable from the jump as exactly the limit computable sets. Tomorrow during the X-hour, Ben will prove a further result as part of Post's Theorem. A set is computable from the jump if and only if it is Δ_2^0, i.e. expressible as either a $\forall\exists$ or $\exists\forall$ formula. Then the limit lemma shows that a set is Δ_2^0 if and only if it is limit computable.