Lecture #9: Emulation of Computations — and a "Universal" $\mathcal{S}$-Program

Lecture Presentation

Proving the Padding Lemma

Our goal is to prove the following result.

Claim (Padding Lemma)**.** *Let $f : \mathbb{N}^k \to \mathbb{N}$ be an $\mathcal{S}$-computable partial or total function (for a positive integer k). Then there exists infinitely many non-negative integers*

$$n_0, n_1, n_2, \ldots$$

such that, if $\mathcal{P}_i$ is an $\mathcal{S}$-program such that $\#(\mathcal{P}_i) = n_i$, then $\mathcal{P}_i$ computes the function f — for every number $i \in \mathbb{N}$.

Idea of Proof

Let $f : \mathbb{N}^k \to \mathbb{N}$ be an $\mathcal{S}$-computable partial or total function — so that there exists some $\mathcal{S}$-program $\mathcal{P}$ such that $\mathcal{P}$ computes f.

Let $\mathcal{P}_0 = \mathcal{P}$, so that $n_0 = \#(\mathcal{P}_0) = \#(\mathcal{P})$.

We know that appending a copy of the "dummy statement"

$$Y \leftarrow Y$$

to the end of $\mathcal{P}_0$ produces a different $\mathcal{S}$-program that computes the same function — but the encoding of the programs are the same.

What if we did *this*, instead?

Confirming That This Idea Would Work

Defining a Useful Function

Let *gen* : $\mathbb{N}^2 \to \mathbb{N}$ be a function that satisfies the following property:

A More Formal Definition of This Function.

Confirming That This Function is Primitive Recursive

Completing a Proof of the Padding Lemma

A “Padding Lemma” for Turing Machines