Title

Whitaker Thompson

November 17, 2025

Question: For an FPT algorithm, if you lift the requirement that the function of k in the runtime is computable, why is that a problem?

Maybe a simpler question to start with would be this: between any two functions $f(n), g(n) : \mathbb{N} \to \mathbb{N}$ such that $f(n) = o(g(n))$, is it the case that there exists an uncomputable function in between them? I.e. does there exist a function $h(n)$ such that $h(n) = \Omega(f(n))$ and $h(n) = o(g(n))$?

Ok, it seems like the answer is largely pedantic, that according to this post, in the original parameterized algorithms textbook, Downey and Fellows actually address this problem.

Definition (Fundamental definitions)**.** *Let A be a parameterized problem*

(i) We say that A is uniformly fixed-parameter tractable *if there exists an algorithm Θ, a constant c, and an arbitrary function $f : \mathbb{N} \to \mathbb{N}$ such that*

  (a) The running time of $\Theta(\langle x, k\rangle)$ is at most $f(k)|x|^c$,

  (b) $\langle x, k\rangle \in A$ iff $\Theta(\langle x, k\rangle) = 1$

(ii) We say that A is strongly uniformly fixed-parameter tractable *if A is uniformly fixed-parameter tractable and*