pgf@stop

Computer Science Tripos 2014 Paper 2 Question 10 part ()(ii)
A Discussion Answer

Thomas Forster

March 9, 2021

I have a machine $\mathfrak{M}$ that has k states, and it accepts infinitely many words. The machine is only finite (poor thing) so there are only finitely many paths-without-loops from the start state to an accepting state, so (by the pigeonhole principle—you'd probably been wondering if that principle would ever come in useful) at least one of those paths is traversed by infinitely many of those words (detours allowed) so it must have a loop hanging off it. Let's call this path $\mathfrak{p}$ and the loop $\mathfrak{l}$. The lengths of $\mathfrak{p}$ and $\mathfrak{l}$ must both be less than k. Call them p and l.

We are asked to show that there is a word of length between k and $2k$ that is accepted by $\mathfrak{M}$. How about the word that sends M along $\mathfrak{p}$ and through $\mathfrak{l}$ precisely once? Do we have

$$k \leq^{(1)} p + l \leq^{(2)} 2k,$$

, as desired? If yes then we have reached our goal. The second inequality (2) is secure; it's the first, (1), that might not be. We might have $p + l < k$.

So how about we try the word that sends $\mathfrak{M}$ through $\mathfrak{l}$ *twice*, or even n times? Its length is $p + n \cdot l$. Pick n just big enough to ensure that $p + n \cdot l \geq k$, but $p + (n - 1) \cdot l < k$. We want $k \leq p + n \cdot l \leq 2k$.

Notice that, since $l < k$, if $p + (n - 1) \cdot l < k$, then $k \leq p + n \cdot l \leq 2k$.