Establishing Closure

A ***complexity class*** is a set of languages. So for example, the regular languages are a class, as are the recursive languages.

As before, ***closure*** means if we have some language(s) in some class, then the result of doing some operation to the language(s) yields a language of the same class.

Here is a typical example.

> **Example.** *Given string z, define the sampling of z as the string consisting of every alternate character in z, starting with the first. For example, the sampling of* THEORY *is* TER. *For a language L, define the sampling of L, denoted L^s, as the set of samplings of all strings in L. Show that the class of r.e. languages is closed under sampling.*

So the task is:

Given a TM that accepts L (but might not halt), build a TM that accepts L^s.

What the new TM must answer is (using the TM for L as a subroutine): given a string w, is there a string in L whose sampling is w ?

A natural way to proceed is to take the input string w, and generate all possible original strings whose sampling would be that string. E.g., given TER, generate TAEAR, TAEARA, TAEBR, etc. (assuming the alphabet of L was the lower-case letters). This generation can easily be done by some looping.

This yields a long, but **finite**, collection of strings that might have been the original string. We then need to test each of these strings to see if any is in L. If at least one is in L, then w is in L^s. If none is in L, then w is not in L^s.

So we submit each possible original string to the TM for L. But because the TM for L is not guaranteed to halt, we must run these submissions in parallel. (Or more accurately, simulate each thread for one step, and repeat indefinitely).

If w is in L^s, then at least one of these runs will terminate with a positive answer, and we will be able to stop the process and answer yes. Otherwise the process might go on forever; but that is okay.