Let us play with Simple Binary S-states Turing Machines (in short SBTM).

Simple means : half infinite (to the left) initially blank tape with the head in state 0 (States from 0 to S-1). No explicit halting state : the SBTM halts when the head try to enter the forbidden (right) part of the tape.

Binary means K=2 characters, 0 (white) and 1 (black).

My preferred numbering scheme differs slightly from Wolfram's : in the instructions table, doubles->triples, states and characters (present in doubles) are written in descending order. As a pleasant consequence, odd SBTM halt in one state. To fix the ideas, here is the 4095th (thus last) 2-states SBTM instructions table :

If you don't like it, you can easily translate my numbering in Wolfram's : use the (reversible) ChangeNumbering instruction :

It is possible to run all S-states SBTM with S = 2, 3 and 4 (When S=4, Mathematica becomes very slow but a tedious C++ GPU programming helps reducing the computation time to 300s not too bad). Here are the (presumably correct) results :

Comments : 1) It is easy to devise a valid criterion that decides looping SBTM (same left part of the tape, current character included, same state of the head and the assurance that the head is in a rightmost position). Stated otherwise a proof of the looping is possible in the frame of arithmetic. So the "Looping" line in the table above is surely OK.

2) I don't know a similar decision criterion for nested SBTM, do you ? When I have a look at SBTM's evolution I see when its behavior is nested but I can't easily prove it nor decide it by program. A proof (or a program) would be needed however.

I have a second question : with S states there are exactly 4S possible triples and each triple correspond to a figure in base 4S. Considering SBTM[nb] nb is simply the decimal translation of the figures-sequence. Equivalently, SBTM[nb] evolution is fully characterized by the ordered sequence of "activated" triples, 0.t1t2t3t4 …, a real number, r, between 0 and 1. If the machine halts or loops, r is rational. What if it is nested ? r is surely irrational but what class of irrational ? Only in very special cases I succeeded to prove a relationship with theta functions.

It is uneasy to check visually that the evolution of the 1536744 SBTM that doesn't halt or loop are effectively nested. Is it possible to devise (and thus program) a criterion that decide a nested behavior in the same way that we can do it for looping machines (easy in this case) ?