A TURING MACHINE FOR EXPONENTIAL FUNCTION *f(x,y)* = x^y

Pedro Lemos

Centre for Logic, Epistemology and the History of Science (CLE-Unicamp)

Contact: pedromendeslemos@gmail.com

This is a Turing Machine which computes the exponential function $f(x,y) = x^y$, where $x, y \in$ N. Instructions format and operation of this machine are intended to best reflect the basic intuitions and conditions outlined by Alan Turing in his *On Computable Numbers, with an Application to the Entscheidungsproblem* (1936), using a version in essence due to Kleene (1952) and Carnielli & Epstein (2008). Hence, a complete instruction will consist of a quadruple (q_i, S, Op, q_j), where q_i is the current state, $S \in \{0, 1\}$ is the current symbol (read by the head), $Op \in \{1, 0, R, L\}$ is an operation, and q_j is the new state. This machine is composed by 4 basic task machines: one which checks if exponent y is zero, a second which checks if base x is zero, a third that is able to copy the base, and a fourth able to multiply multiple factors (in this case, factors will be all equal). They were conveniently separated in order to ease the reader's task to understand each step of its operation. We adopt the convention that a number n is represented by a string of n+1 symbols "1". Thus, an entry (x, y) will be represented by two respective strings of x+1 and y+1 symbols "1", separated by a single "0" (or a blank), and as an output, this machine will generate a string of x^y+1 symbols "1". Some instructions are followed by a brief description of what's going on. This machine can be tested on the internet. We adapted the instructions to a particular format, so it could be implemented on a java based TM emulator. These are presented right after our TM instructions.

(I) MACHINE (A) – The Zero Exponent Checker

q_1 1 R q_1 – Head passes through the base
q_1 0 R q_2
q_2 1 R q_3 – Checks if exponent is zero

→ q_3 0 1 q_4 – Exponent is zero, it erases the exponent and writes 1, resulting string "...11..."
q_4 1 L q_4
q_4 0 L q_5
q_5 1 O q_5 – Starts to erase the base
q_5 0 L q_6
q_6 1 1 q_5
q_6 0 0 q_7
q_7 0 R q_7
q_7 1 1 q_{66} – Halts the machine in standard position, that is, in the leftmost symbol "1" of the string

→ q_3 1 1 q_8 – Exponent is not zero, goes back to base and implements Machine (B)
q_8 1 L q_8
q_8 0 L q_9

(II) MACHINE (B) – *The Zero Base Checker*

q_9 1 L q_{10}

→ q_{10} 0 R q_{11} – Base is zero, erases the exponent and halts in standard position
q_{11} 1 R q_{11}
q_{11} 0 R q_{12}
q_{12} 1 0 q_{12}
q_{12} 0 R q_{13}
q_{13} 1 1 q_{12}
q_{13} 0 0 q_{14}
q_{14} 0 L q_{14}
q_{14} 1 1 q_{66}

→ q_{10} 1 1 q_{15} – Base is not zero, goes to the rightmost "1" of exponent and implements Machine (C)

q_{15} 1 R q_{15}
q_{15} 0 R q_{16}
q_{16} 1 R q_{16}
q_{16} 0 L q_{17}

(III) MACHINE (C) *The Base copier*

q_{17} 1 L q_{18} – First this machine checks if exponent is, or has reached, number 1. If it has just started, and it finds out 1, it enters state q_{19}, and erases exponent. If it verifies it has not reached string "..0110..", this stage will work until exponent is reduced to number 1.
q_{18} 1 L q_{19}

→ q_{19} 0 R q_{37} – Exponent is, or has reached, number 1. It now erases exponent.
q_{37} 1 0 q_{37}
q_{37} 0 R q_{38}
q_{38} 1 0 q_{39}
q_{39} 0 L q_{39}
q_{39} 1 1 q_{40} – Exponent is now erased and the head is positioned on the rightmost "1" of the rightmost factor (This is the moment when we have *y* copies of the base, and we can implement Machine (D), the multiplier of multiple factors).

q_{19} 1 R q_{20} – Exponent is not, or hasn't reached, number 1; Goes back to the rightmost "1", and starts (or continues) to copy the base

q_{20} 1 R q_{21}

q_{21} 1 0 q_{21} – This is the moment when we erase a symbol "1" of the exponent, move the head to the leftmost copy of the base, and duplicate it. If we have just started, our own base will be the "leftmost copy", otherwise, the head will move until it reaches the leftmost copy of the base, and replicate it.

q_{21} 0 L q_{22}
q_{22} 1 L q_{22}
q_{22} 0 L q_{23}
q_{23} 1 1 q_{22} – Every time it finds a string, the head passes through it

q_{23} 0 1 q_{24} – We found the leftmost copy, the head writes a symbol "1", so it knows where the new copy should be written.

q_{24} 1 R q_{24}
q_{24} 0 R q_{25}
q_{25} 1 R q_{26} – Notice that when returning to the string which is being copied, it first tests if it was already reduced to a single symbol "1". If this is the case, then it starts to rewrite the symbols which were erased from the original string

q_{26} 1 1 q_{27} – Implements, or continues, the process of erasing symbols
q_{27} 1 R q_{27}
q_{27} 0 L q_{28}
q_{28} 1 0 q_{28}
q_{28} 0 L q_{29}
q_{29} 1 L q_{29}
q_{29} 0 L q_{30}
q_{30} 1 L q_{30}
q_{30} 0 0 q_{23}

q_{26} 0 1 q_{31} – It found a single "1", starts rewriting the symbols, and then goes back to the exponent
q_{31} 1 R q_{32}
q_{32} 0 0 q_{26}
q_{32} 1 L q_{33}

q_{33} 1 0 q_{33}
q_{33} 0 R q_{34}
q_{34} 1 R q_{34}
q_{34} 0 R q_{35}
q_{35} 1 1 q_{34}
q_{35} 0 L q_{36}
q_{36} 0 L q_{17}- Back to exponent, enters state q_{17} to verify if it was reduced to number 1

(IV) MACHINE (D) – *The Multiplier of Multiple Factors*

q_{40} 1 L q_{40} – Checks if there is a string to multiply.
q_{40} 0 L q_{41}
q_{41} 0 0 q_{65}
q_{65} 0 R q_{65}
q_{65} 1 1 q_{66} – Machine halts

q_{41} 1 1 q_{42} – There is a string to multiply, the head will now find the leftmost factor, and writes a symbol "1" just left to that factor, separated by a single "0", where we leave the product
q_{42} 1 L q_{42}
q_{42} 0 L q_{43}
q_{43} 1 1 q_{42}
q_{43} 0 1 q_{44}
q_{44} 1 R q_{44}
q_{44} 0 R q_{45}
q_{45} 1 1 q_{44}
q_{45} 0 L q_{46}
q_{46} 0 L q_{46}

q_{46} 1 L q_{47} – Now it will test if the multiplier has reached number zero. Notice that, differently from Machine (C), which checks if exponent has been reduced to "…11…", this machine checks for a zero, this happens because each symbol of the multiplier makes a copy of the factor, and leaves the whole product in the leftmost string, thus erasing both the multiplier and the factor that is being multiplied.

→ q_{47} 0 R q_{61} – Multiplier string has reached number zero

q_{61} 1 0 q_{62}
q_{62} 0 L q_{62}
q_{62} 1 0 q_{63} – Starts to erase the factor that has been multiplied
q_{63} 0 L q_{64}
q_{64} 1 1 q_{62}
q_{64} 0 L q_{40} – Back to state q_{40}, to check if there is a string to multiply, or if we have reached the desired result

→ q_{47} 1 R q_{48} - Multiplier string has not reached "...1...", so it starts (or continues) to multiply the factor in its left
q_{48} 1 0 q_{48} – Erases a symbol from the Multiplier
q_{48} 0 L q_{49}
q_{49} 1 L q_{49}
q_{49} 0 L q_{50}

q_{50} 1 0 q_{50} – Erases "1" from the factor and writes a symbol "1" in the product string
q_{50} 0 L q_{51}
q_{51} 1 L q_{51} ←
q_{51} 0 L q_{52}
→ q_{52} 1 1 q_{51}
q_{52} 0 R q_{67} – Found where the product string is

q_{67} 0 1 q_{53}
q_{53} 1 R q_{53} – Now the head moves right, until it reaches the factor that is being multiplied.
q_{53} 0 R q_{54}
q_{54} 1 1 q_{53}
q_{54} 0 L q_{55}
q_{55} 0 L q_{55}

q_{55} 1 L q_{56} – Tests if multiplied factor has been reduced to "...1..."

q_{56} 1 R q_{50} – Back to state q_{50}

q_{56} 0 R q_{57} - Factor has been reduced to "...1...", starts rewriting the symbols it has erased.

q_{57} 1 R q_{58}
q_{58} 0 1 q_{57}
q_{58} 1 L q_{59}
q_{59} 1 0 q_{59}
q_{59} 0 R q_{60}
q_{60} 1 R q_{60}
q_{60} 0 L q_{46} Back to state q_{46}, and test if Multiplier has been reduced to "...1..."

(V) Instructions to implement this TM on the internet.

The quadruples of this machine were adapted to test it, using an emulator. You first need to access the following link: http://ironphoenix.org/tril/tm/ . Then in "Load new program", select "Subtracter" and click on "Load new Program"; then click on "Clear Program" and erase the input in the box "Initial characters on tape". Now put your entry in "Initial characters on tape" like this: type x+1 symbols "1", followed by "_", and y+1 symbols "1" to represent x^y; for example, for 2^3, we type "111_1111" (without quotation marks). Then set "Initial tape position" in maximum position, that is, "29950". Now you only have to copy instructions below, paste it into "Programming" box, click on "Install Program" and "Start" the machine.

```
1 1 1 >
1 _ 2 >
2 1 3 >
3 _ 4 1
4 1 4 <
4 _ 5 <
5 1 5 _
5 _ 6 <
6 1 5 1
6 _ 7 _
7 _ 7 >
```

7 1 66 1
3 1 8 1
8 1 8 <
8 _ 9 <
9 1 10 <
10 _ 11 >
11 1 11 >
11 _ 12 >
12 1 12 _
12 _ 13 >
13 1 12 1
13 _ 14 _
14 _ 14 <
14 1 66 1
10 1 15 1
15 1 15 >
15 _ 16 >
16 1 16 >
16 _ 17 <
17 1 18 <
18 1 19 <
19 _ 37 >
37 1 37 _
37 _ 38 >
38 1 39 _
39 _ 39 <
39 1 40 1
19 1 20 >
20 1 21 >
21 1 21 _
21 _ 22 <
22 1 22 <
22 _ 23 <
23 1 22 1
23 _ 24 1
24 1 24 >
24 _ 25 >
25 1 26 >
26 1 27 1
27 1 27 >
27 _ 28 <
28 1 28 _
28 _ 29 <
29 1 29 <
29 _ 30 <
30 1 30 <
30 _ 23 _
26 _ 31 1
31 1 32 >
32 _ 26 _
32 1 33 <
33 1 33 _
33 _ 34 >
34 1 34 >
34 _ 35 >

35 1 34 1
35 _ 36 <
36 _ 17 <
40 1 40 <
40 _ 41 <
41 _ 65 _
65 _ 65 >
65 1 66 1
41 1 42 1
42 1 42 <
42 _ 43 <
43 1 42 1
43 _ 44 1
44 1 44 >
44 _ 45 >
45 1 44 1
45 _ 46 <
46 _ 46 <
46 1 47 <
47 _ 61 >
61 1 62 _
62 _ 62 <
62 1 63 _
63 _ 64 <
64 1 62 1
64 _ 40 <
47 1 48 >
48 1 48 _
48 _ 49 <
49 1 49 <
49 _ 50 <
50 1 50 _
50 _ 51 <
51 1 51 <
51 _ 52 <
52 1 51 1
52 _ 67 >
67 _ 53 1
53 1 53 >
53 _ 54 >
54 1 53 1
54 _ 55 <
55 _ 55 <
55 1 56 <
56 1 50 >
56 _ 57 >
57 1 58 >
58 _ 57 1
58 1 59 <
59 1 59 _
59 _ 60 >
60 1 60 >
60 _ 46 <