Why one should reset the noninput variables of M if it is used more than once in a program.

Consider the URM $M^{\mathbf{z}}_{\mathbf{x}}$ —that is, URM with input/output choice as shown— below:

1 : $\mathbf{x} \leftarrow \mathbf{x} + 1$
2 : $\mathbf{x} \leftarrow \mathbf{x} + 1$
3 : $\mathbf{z} \leftarrow 0$
4 : **stop**

Here $M^{\mathbf{z}}_{\mathbf{x}} = \lambda z.2$.

What happens if we stack two copies of M on top of each other? *Let's choose the same I/O variables*.

We obtain, $N^{\mathbf{z}}_{\mathbf{x}}$ below:

1 : $\mathbf{x} \leftarrow \mathbf{x} + 1$
2 : $\mathbf{x} \leftarrow \mathbf{x} + 1$
3 : $\mathbf{z} \leftarrow 0$
4 : $\mathbf{x} \leftarrow \mathbf{x} + 1$
5 : $\mathbf{x} \leftarrow \mathbf{x} + 1$
6 : $\mathbf{z} \leftarrow 0$
7 : **stop**

Note that $N^{\mathbf{z}}_{\mathbf{x}} = \lambda z.4$. *This is odd given that N is*

M

M

and *each* M *separately* computes $\lambda z.2$.

What happened?

The bottom M *does not work as expected!* WHY?

Because the noninput variable ($\mathbf{x}$) of the second M was NOT set to 0. The second M starts off with *initial value* for $\mathbf{x}$ equal to 2.