How even the simplest multiprogram-transformations need not work

We consider the 2-component multiprogram

Pre : $\neg x$

A: $|[\ x := \mathrm{true}\ ]|$ B: $|[\ \mathbf{if}\ x \rightarrow x := \mathrm{false}\ ]|$

Now we can prove that a postcondition of this program is $\neg x$.

Next, we slightly alter this multiprogram by replacing in component A statement $x := \mathrm{true}$ with $\langle x := \mathrm{true} \rangle \ ; \ \langle x := \mathrm{true} \rangle$. Now we can no longer prove postcondition $\neg x$.

* * *

The above observation was made in a discussion with my student Toon Wijnands. I recorded it to remind us of the fact that in the context of multiprograms there is no such thing as "intuition": each - even the simplest conjecture - has to be rigorously proven. I also recorded it because the above example can serve as a simple demonstration to our students of the intricacies of parallel programming.

WHJ Feijen,
Eindhoven,
9 December 1993