A case of intuitive reasoning

Mathematical intuition is a very dangerous and unreliable compass, even more so in the case of multiprogramming. Recently we showed a very small multiprogram to our class, namely the following

```
A:     $y := \mathit{false}$                    B:     $x := \mathit{false}$
     ; if $y \rightarrow \mathit{skip}$ fi              ; if $x \rightarrow \mathit{skip}$ fi
```

These are just two straight-line programs, each consisting of just two simple statements. Hardly anything simpler can be conceived, can't it?

Now, for the sake of letting both components terminate, we granted the class the possibility to add statements "$x := \mathit{true}$" to component A , as many as they wanted and wherever they wanted. And similarly, statements "$y := \mathit{true}$" were allowed to be added to B .

The class did not hesitate very long. Because component B is "waiting" for x to become true, termination of B becomes most likely if A performs "$x := \mathit{true}$" as often as possible. And symmetrically so for "$y := \mathit{true}$" . So, here is the solution:

```
A:     $x := \mathit{true}$                     B:     $y := \mathit{true}$
     ; $y := \mathit{false}$                         ; $x := \mathit{false}$
     ; $x := \mathit{true}$                          ; $y := \mathit{true}$
     ; if $y \rightarrow \mathit{skip}$ fi              ; if $x \rightarrow \mathit{skip}$ fi
     ; $x := \mathit{true}$                          ; $y := \mathit{true}$
```

But, alas, each effort to give a genuine termination proof failed. And indeed, there is no guarantee that both components terminate. (Let A proceed to its if-statement. Then $x \wedge \neg y$ holds. Next, let B perform its first "$y := \mathit{true}$" . Then $x \wedge y$ holds. Now let A terminate. Then B gets stuck.)

The nice thing is that, if we remove the first line from each component, i.e. if we consider

```
$A$:     $y := false$                 $B$:     $x := false$
       ; $x := true$                         ; $y := true$
       ; if $y \rightarrow$ skip fi          ; if $x \rightarrow$ skip fi
       ; $x := true$                         ; $y := true$
```

then everything is okay (proof omitted here [?]).

*   *
     *

To us, the above is a very nice example to demonstrate the intricacies of multiprogramming to a novice audience, and to warn them to never lean on “intuition”, but on rigorous formal proofs instead.