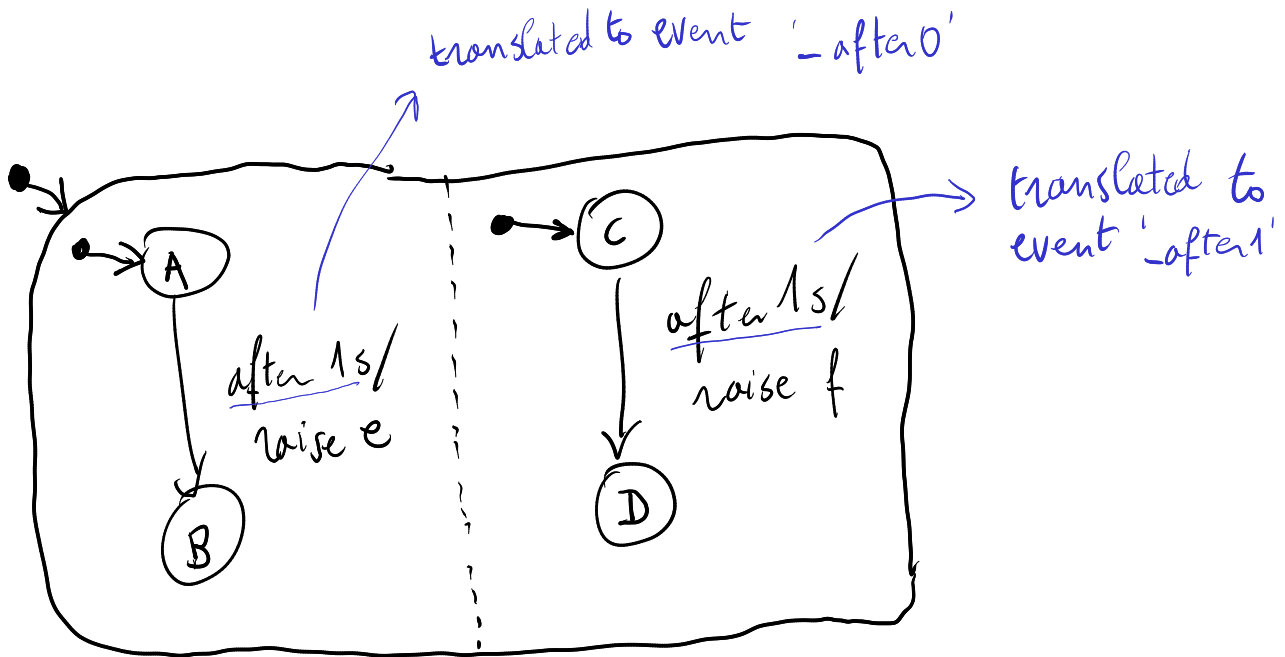


# Shortcoming of YAKINDU's unit testing framework



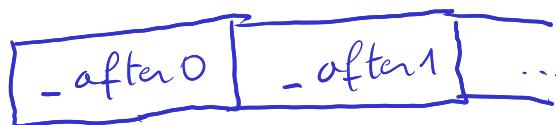
in YAKINDU unit test:

```
enter
proceed 1s
assert e
assert f
exit
```

↪ one of these will fail!

why?

After-triggers are translated to input events behind the screens. If the statechart is started at time 0.0, then at time 1.0, the input event queue could look as follows:



The execution of a statechart is a sequence of run-to-completion (RTC) steps. One RTC step is executed for every input event. So in our case, 2 RTC steps are executed:

time=1.0 [ RTC 0 : input = '\\_after0', output = 'e'  
 RTC 1 : input = '\\_after1', output = 'f'

Because both RTC steps happen at time 1.0, the unit test statement "proceed 1s" will execute both. While both 'e' and 'f' are raised (and are visible to the environment) a unit test in YAKINDU can only see the output events of the last RTC step, in this case 'f'.