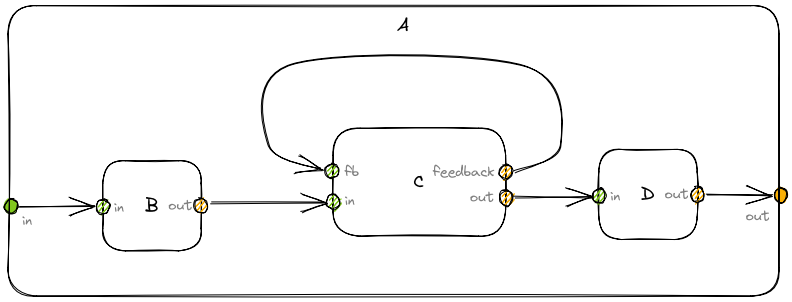


This example consists of four (4) components:

1. A
2. B
3. C
4. D

A is a Container.

B, C, and, D are Leaves.

And, there are five (5) different events, *u, v, w, x, y*.

Each component has:

- one (1) input queue
- one (1) output queue
- an input pin namespace
- an output pin namespace
- a name.

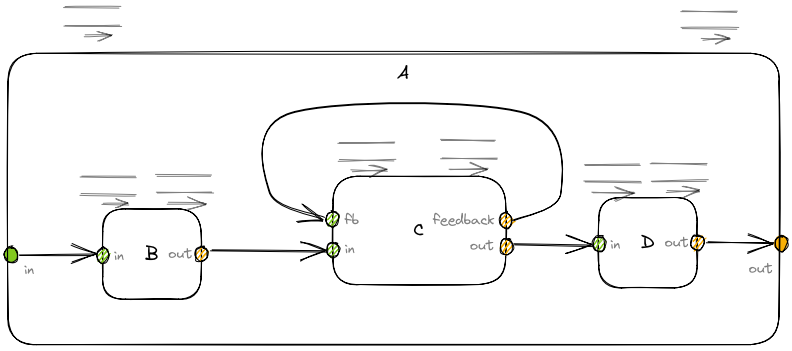
A component is like a prototype. A component can be instantiated more than once. An instance is distinguished by its (x,y) coordinate on a drawing, or, by a name in a textual representation.

Actions:

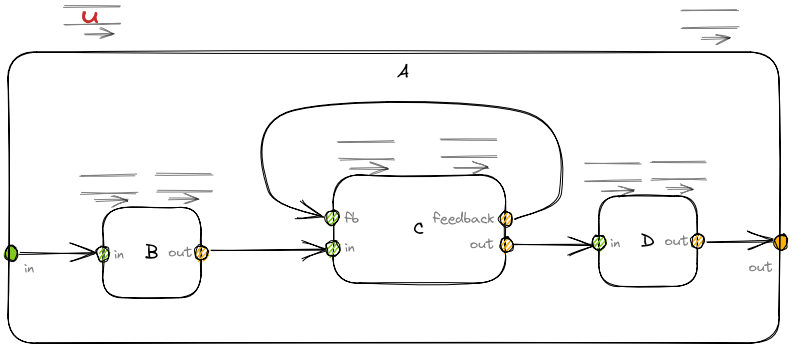
- A
  - when A gets any input event on pin "in", it sends that event *down* to its child B (see Routing Table)
  - A outputs, on pin "out", whatever its child D outputs on its pin "out" (see Routing Table)
  - *Routing Table*
    - .in -> B.in
    - B.out -> C.in
    - C.feedback -> C.fb
    - C.out -> D.in
    - D.out -> .out
- B
  - whenever B gets any input on pin "in", it sends two (2) messages, *v* and *w* to its output "out"
  - N.B. it sends message *v* first, then sends message *w* and sends both messages before finishing its reaction
- D

    - D is a pass-through component
    - when any event arrives on input pin "in", D sends the same event out on output "out"
- C
    - when C gets any event on input pin "in", it sends the event to its output pin "out", and, it sends a feedback event to its output pin "feedback"
    - when C gets any event on input pin "fb", it sends the event to its output pin "out" only, i.e. it does not send a feedback event
    - to make this example explicit, we will cause C to send a different feedback event each time it is triggered
    - the first feedback event is *y*
    - the second feedback event is *z*
    - there is no need for further feedback events in this example, but, to be explicit, all other feedback events are *?*

Step 0 - Annotated With Queues

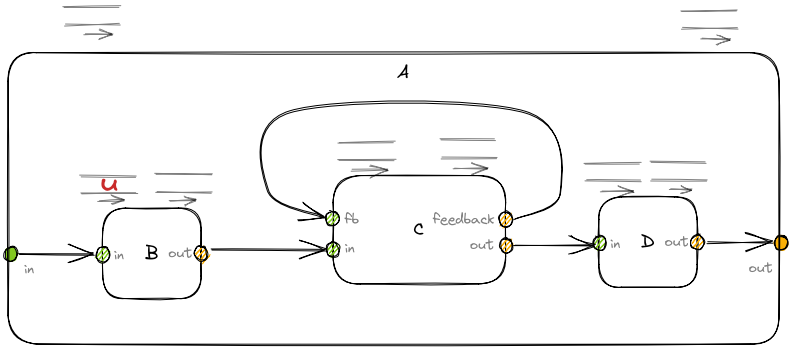


Step 1

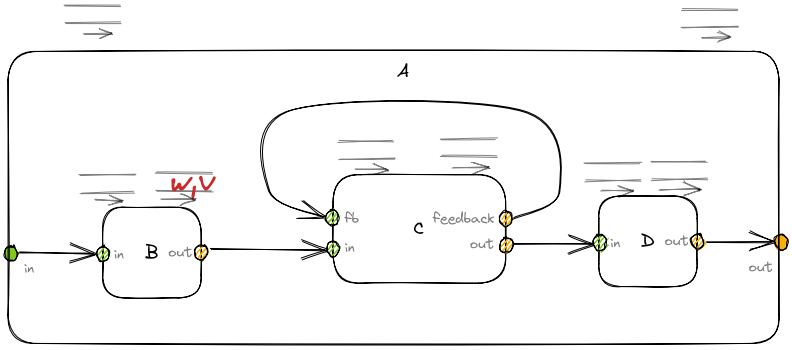


A consumes *u* and passes it *down* to child B.

Step 2

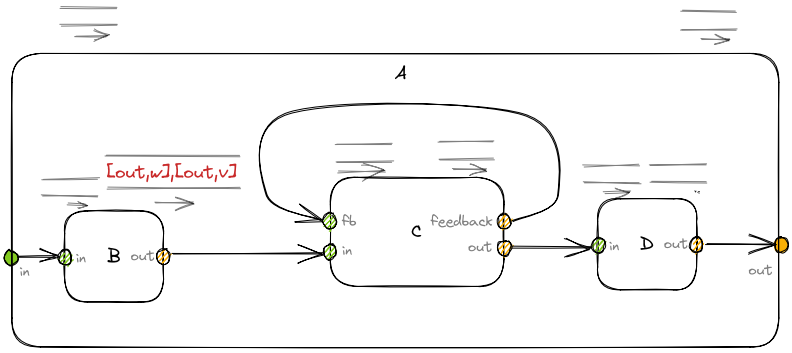


Step 3

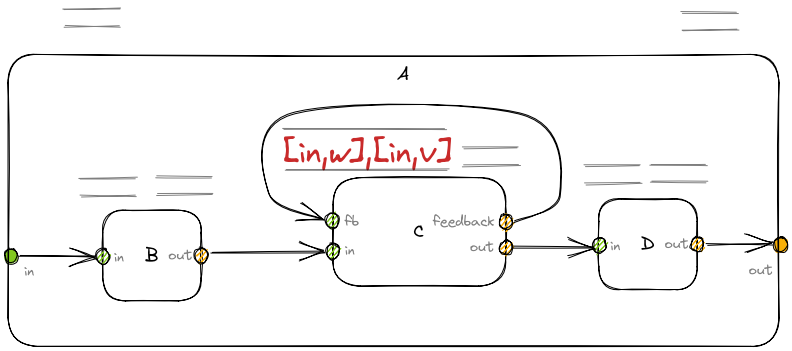


Child B consumes *u* and produces two outputs *v* and *w* (in that order).

Step 3 - Gory Detail



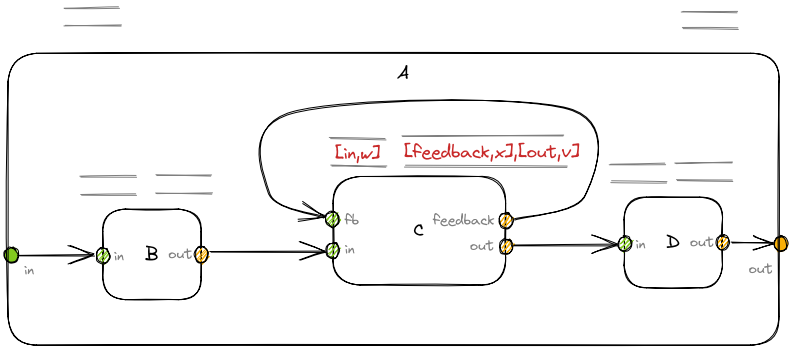
Step 4



A receives B's outputs and transfers them to C, using its routing table.

In the process, A remaps B's output pin names to C's input pin names.

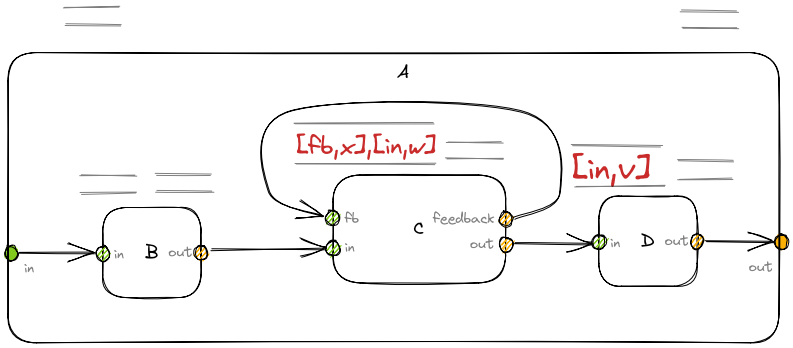
Step 5



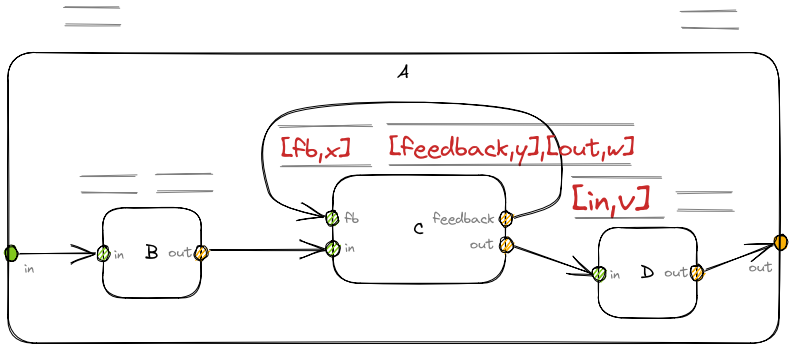
C consumes the first event, *v*, and executes the code associated with its "in" pin.

In this case, C produces two (2) outputs *v* and *x*. The outputs are sent to different output pins.

Step 6



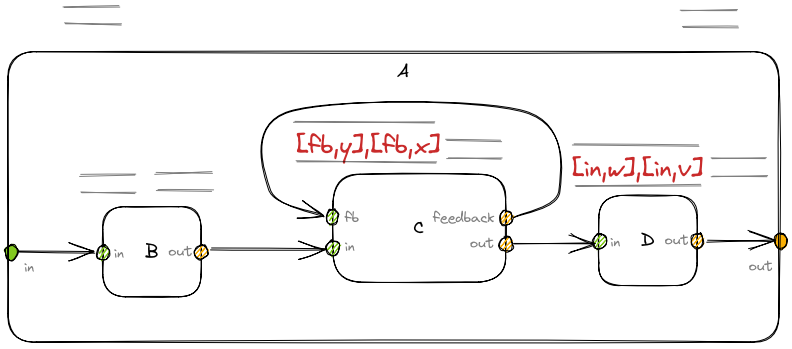
Step 7



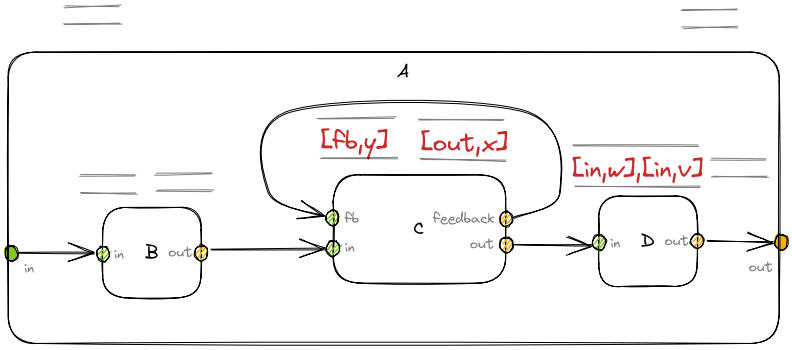
C consumes the first event, *w*, and executes the code associated with its "in" pin.

To make this example explicit, the feedback event is *y* this time.

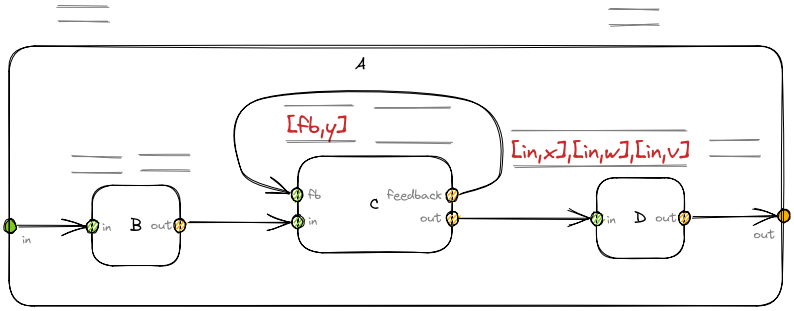
Step 8



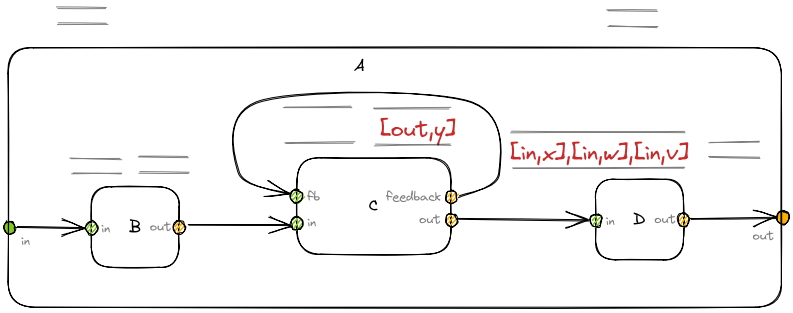
Step 9



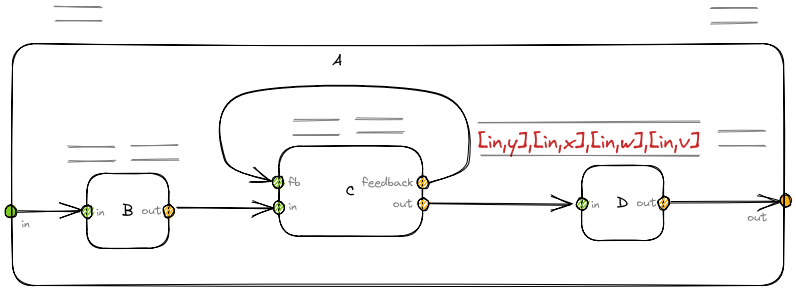
Step 10



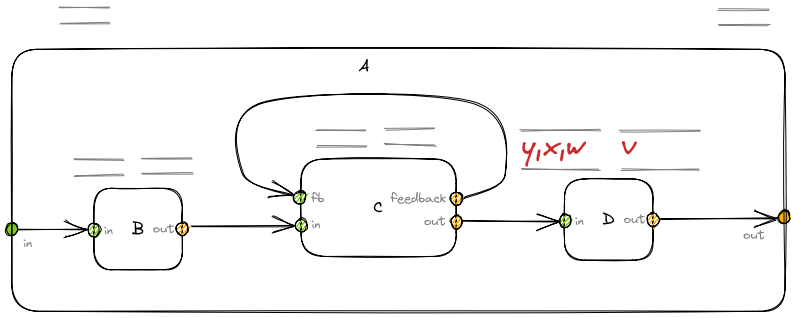
Step 11



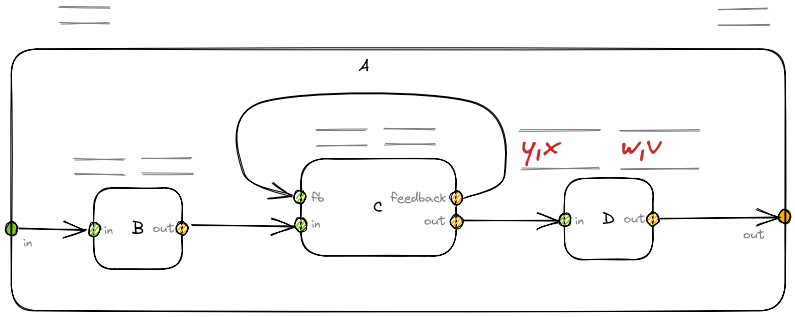
Step 12



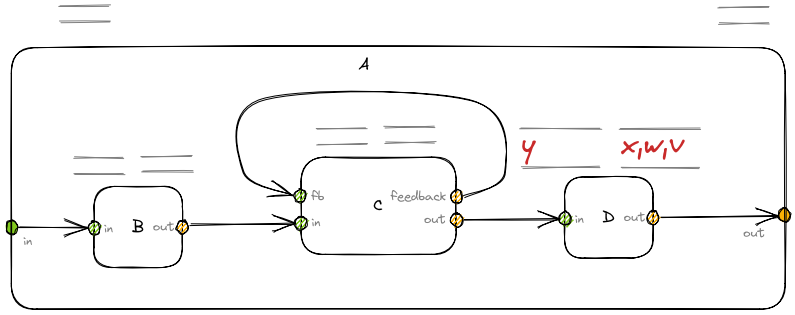
Step 13



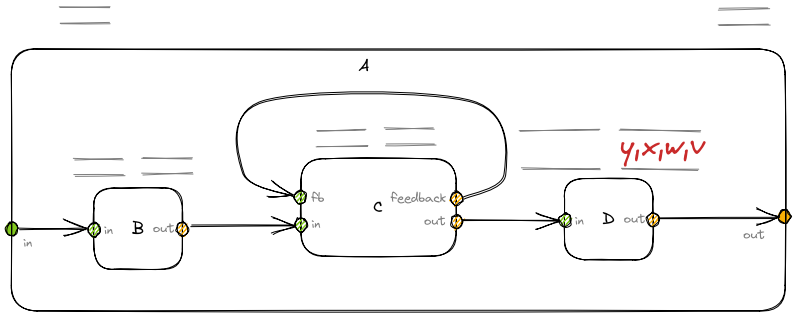
Step 14



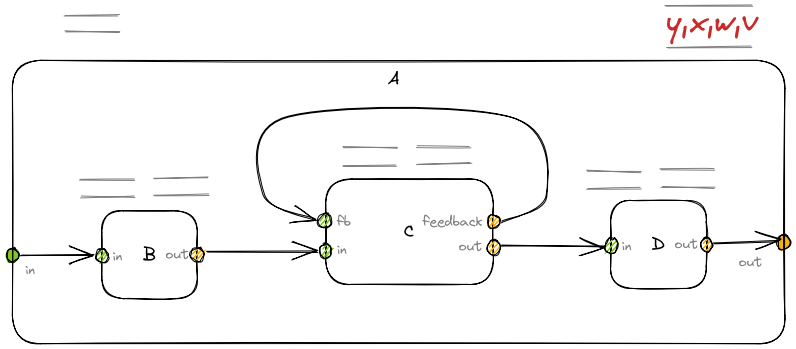
Step 15



Step 16



Step 17



Python Code

https://github.com/guitarvydas/py0d/tree/dev