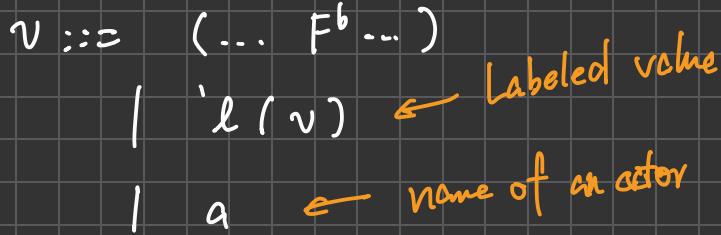
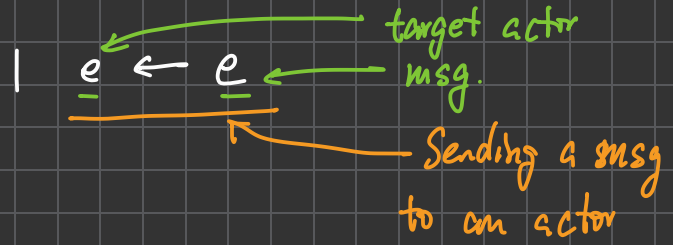


Side Effect  
Introducing



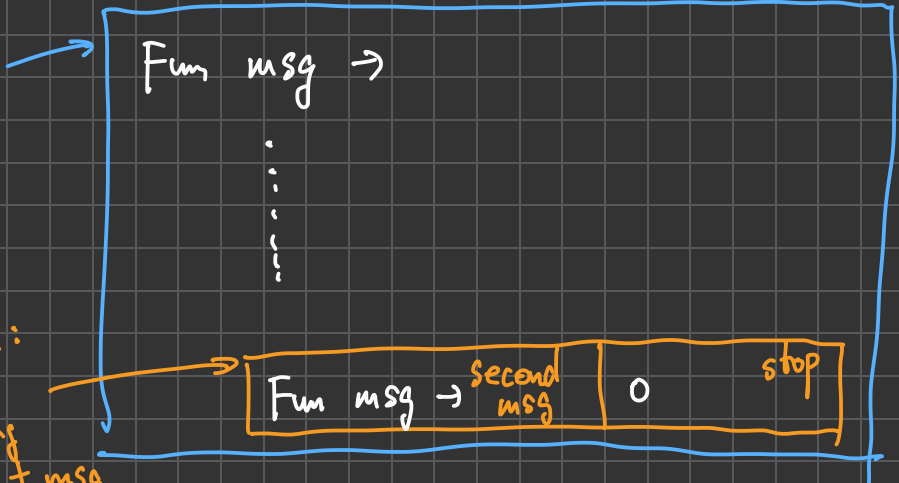
## Behavior of an Actor

{ Fun the\_name\_of\_the\_actor\_itself → my name

Starting  
an actor

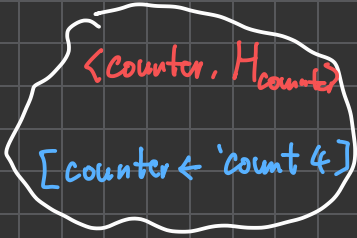
{ Fun initial\_data → data to initialize actor

Actor Msg  
Handler



Continuation:  
What to do  
after handling  
the current msg

State 1

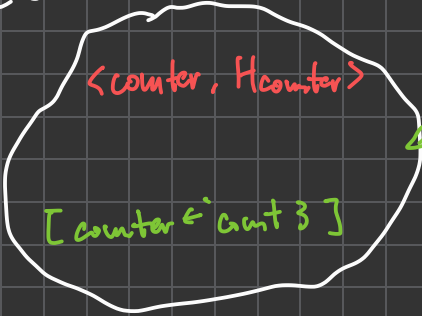


$H_{\text{counter}}(\text{'count 4'})$

↳ send new msg

counter  $\leftarrow$  'count 3'

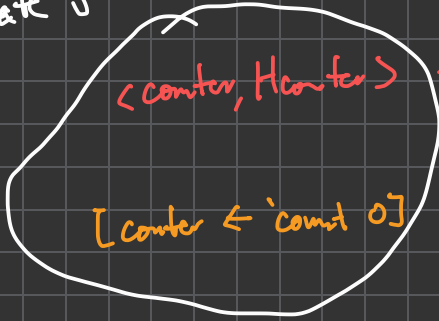
State 2



$H_{\text{counter}}(\text{'count 3'})$

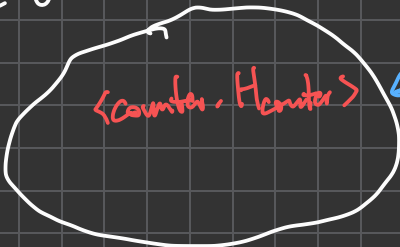
↳ 'count 2'

State 5



$H_{\text{counter}}(\text{'count 0'})$

State 6



No more msgs

# Using Y-combinator

Let actor =

AFbV

initialization

Fun my\_name =>

Stateless

Fun init\_data ->

Event Handling Loop

(Y (Fun this ->

One Event Handler

Fun msg ->

handling msgs

Continue

{

(this)

))

Let actor =

AFbV

State FUL

initialization { Fun my\_name =>

~~Fun init\_data =>~~

Event Handling Loop

( { Fun this => Fun curr\_state =>

One Event Handler

{ Fun msg =>

handling msgs  
compute next state

Continue {

(this next state)

))

Web servers

JS

app.listen(8000, (msg) => {  
... handling msgs ...  
})

C++

while (true) {  
msg = recv();  
handle(msg);  
}

# Formal Model. Semantics

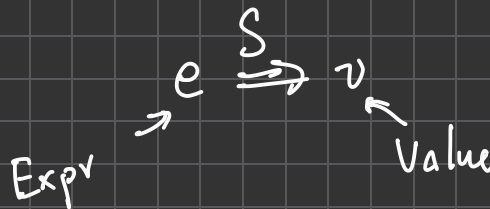
## Two Layered Design

Global



Small Step  
 $\rightarrow$

Local



Big-step  
 $e \xRightarrow{S} v$

$S_G$

Soup

$\langle a, H_a \rangle$

$\vdots \langle b, H_b \rangle$

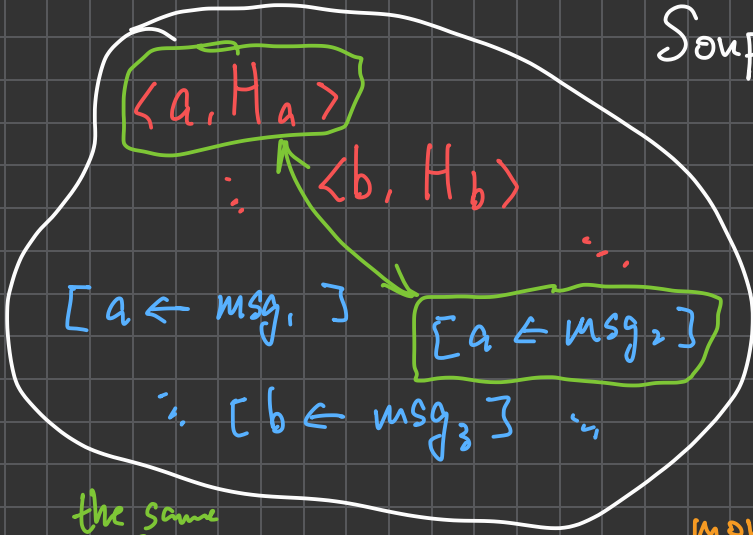
$\dots$

$[a \leftarrow msg_1]$

$[a \leftarrow msg_2]$

$\vdots [b \leftarrow msg_3] \dots$

Soup

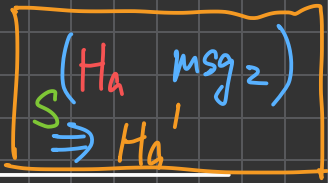


the same actor

$\langle a, H_a \rangle \in S_G$

$[a \leftarrow msg_2] \in S_G$

monotonic



$$S_G \rightarrow S_G \setminus \{ \langle a, H_a \rangle, [a \leftarrow msg_2] \}$$


---


$$\cup S \cup \langle a, H_a' \rangle$$

State Sem  $\langle e, \sigma_1 \rangle \Rightarrow \langle v, \sigma_2 \rangle$

Can be  
non monotonic  
e.g. can make changes

Big Step

Small-step

Big step w/  
Side effects

$e \xrightarrow{S} v$

Side effects

Strictly monotonic  
Can only introduce new side-effects