Merging merges, more or less

Amos Robinson, UNSW

No buffer

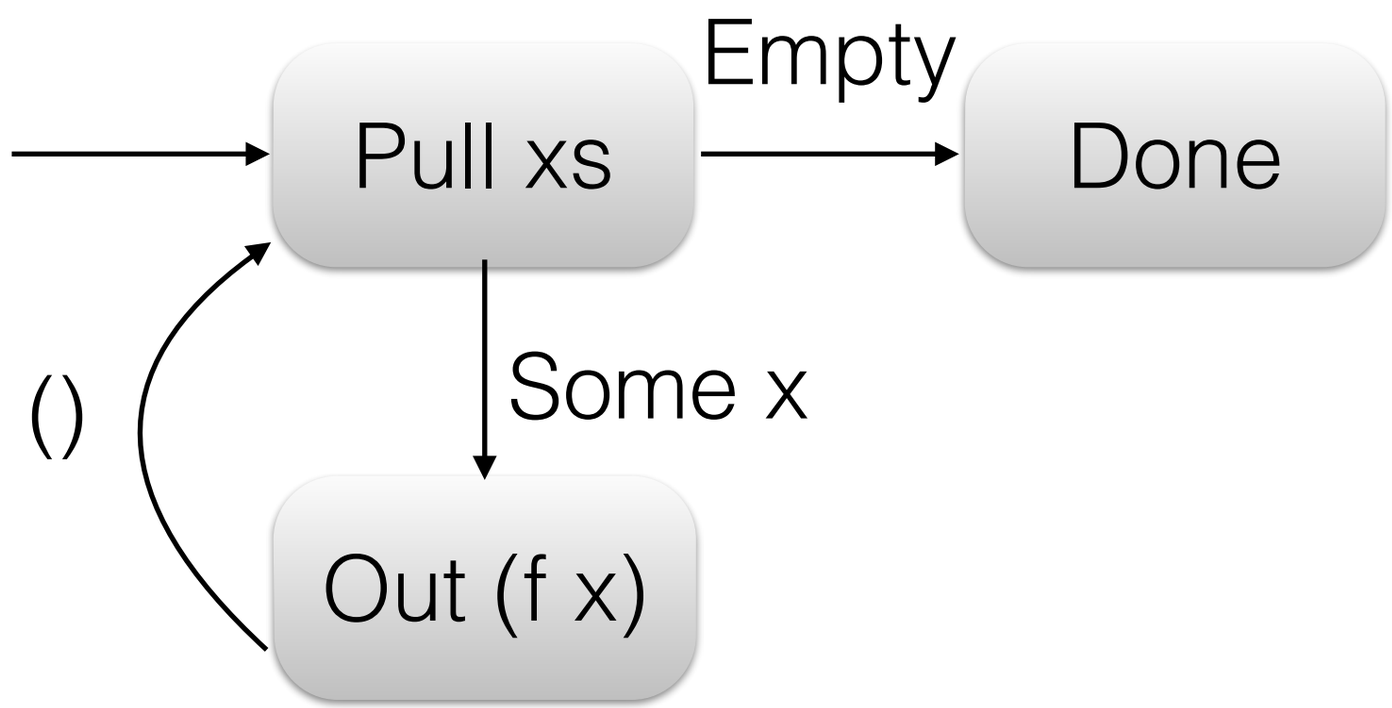
```
zipf     (xs ys : Source Int)
  = let  xs'  = filter  (>0)     xs
         ys'  = filter  (<0)     ys
    in   zip xs' ys'
```

Unbounded buffer

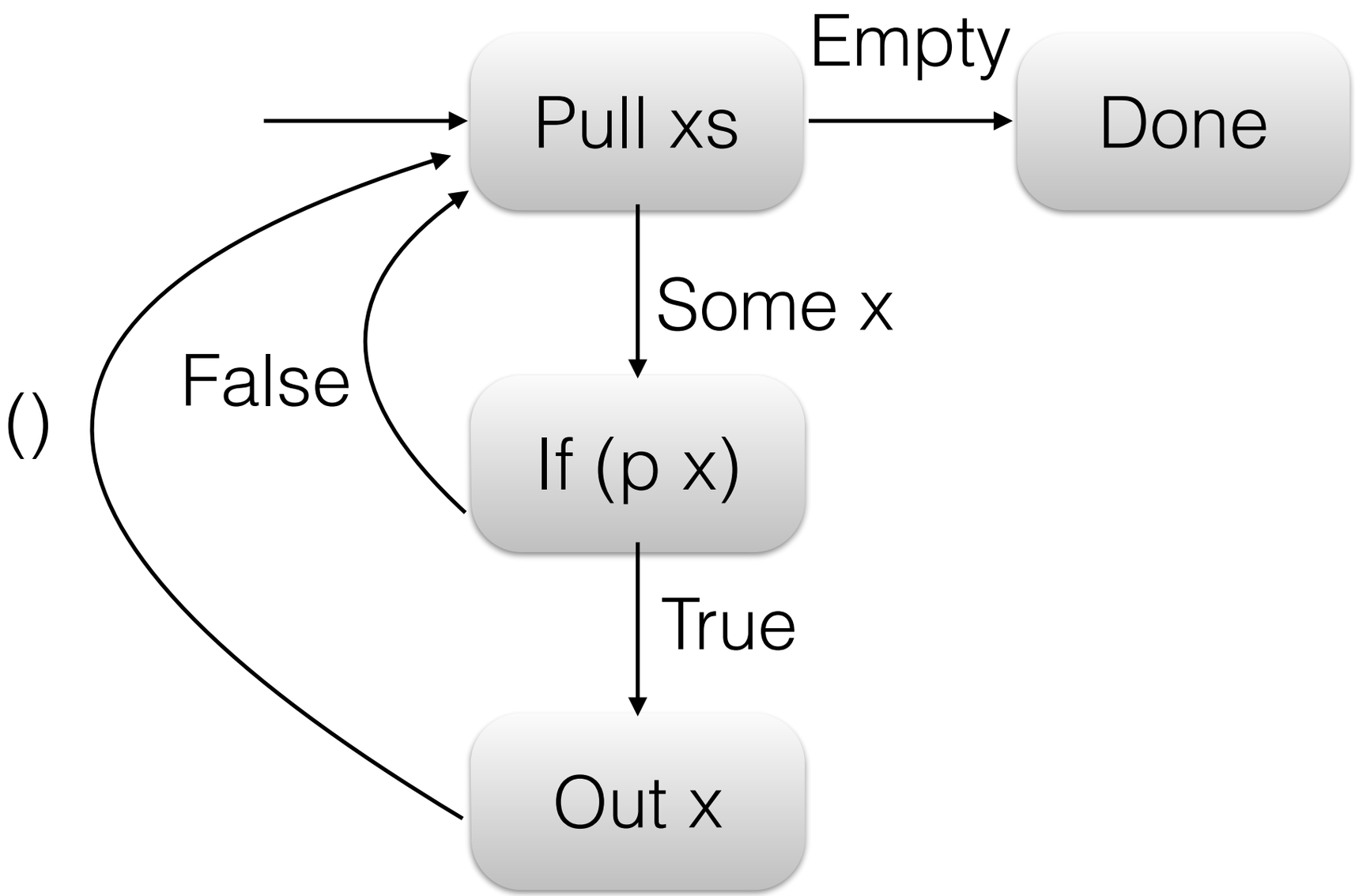
```
bad    (xs     : Source Int)
  = let  xs'  = filter  (>0)     xs
         xs'' = filter  (<0)     xs
    in   zip xs' xs''
```

Combinators

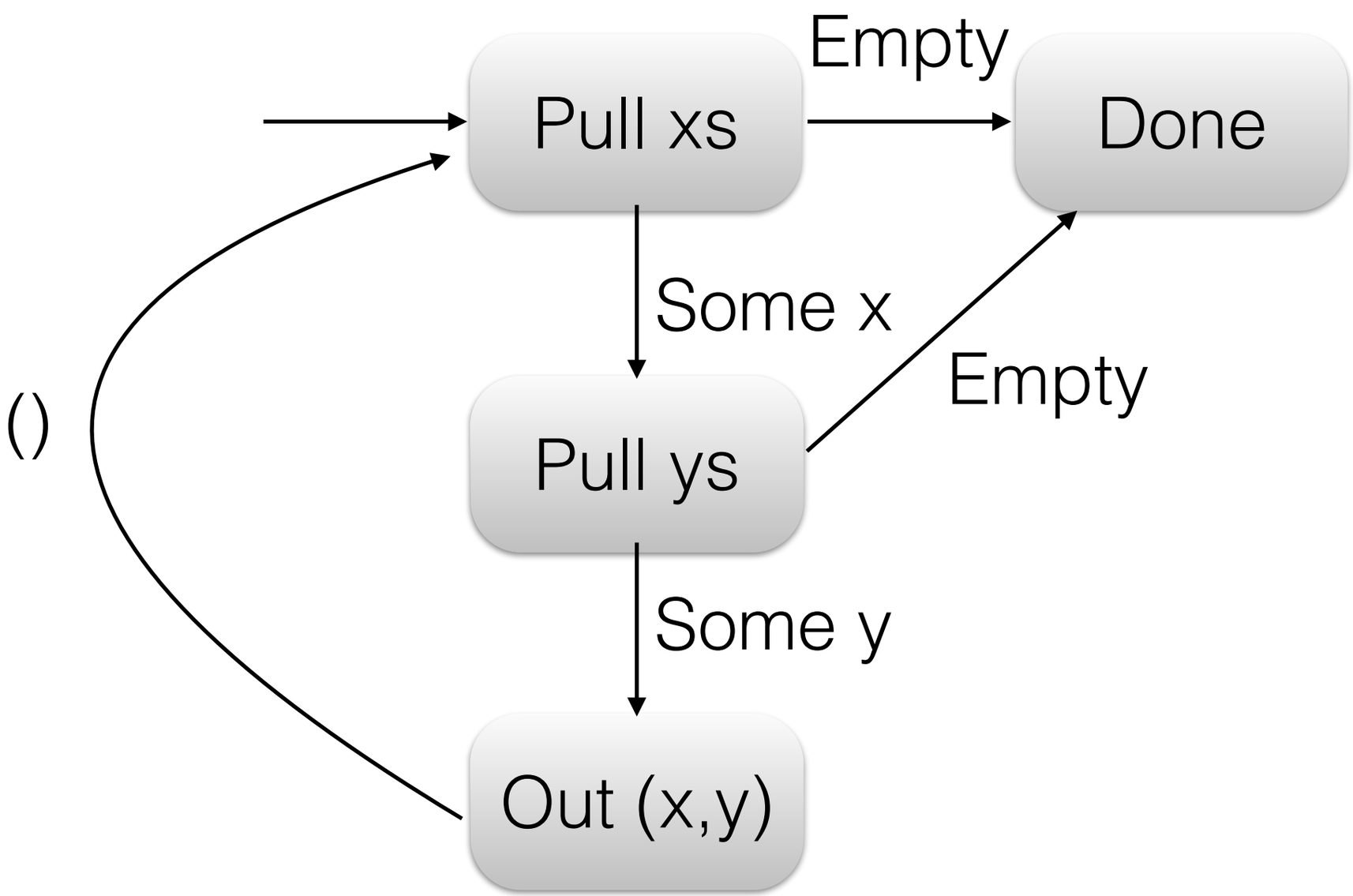
Map f xs



Filter p xs



Zip xs ys

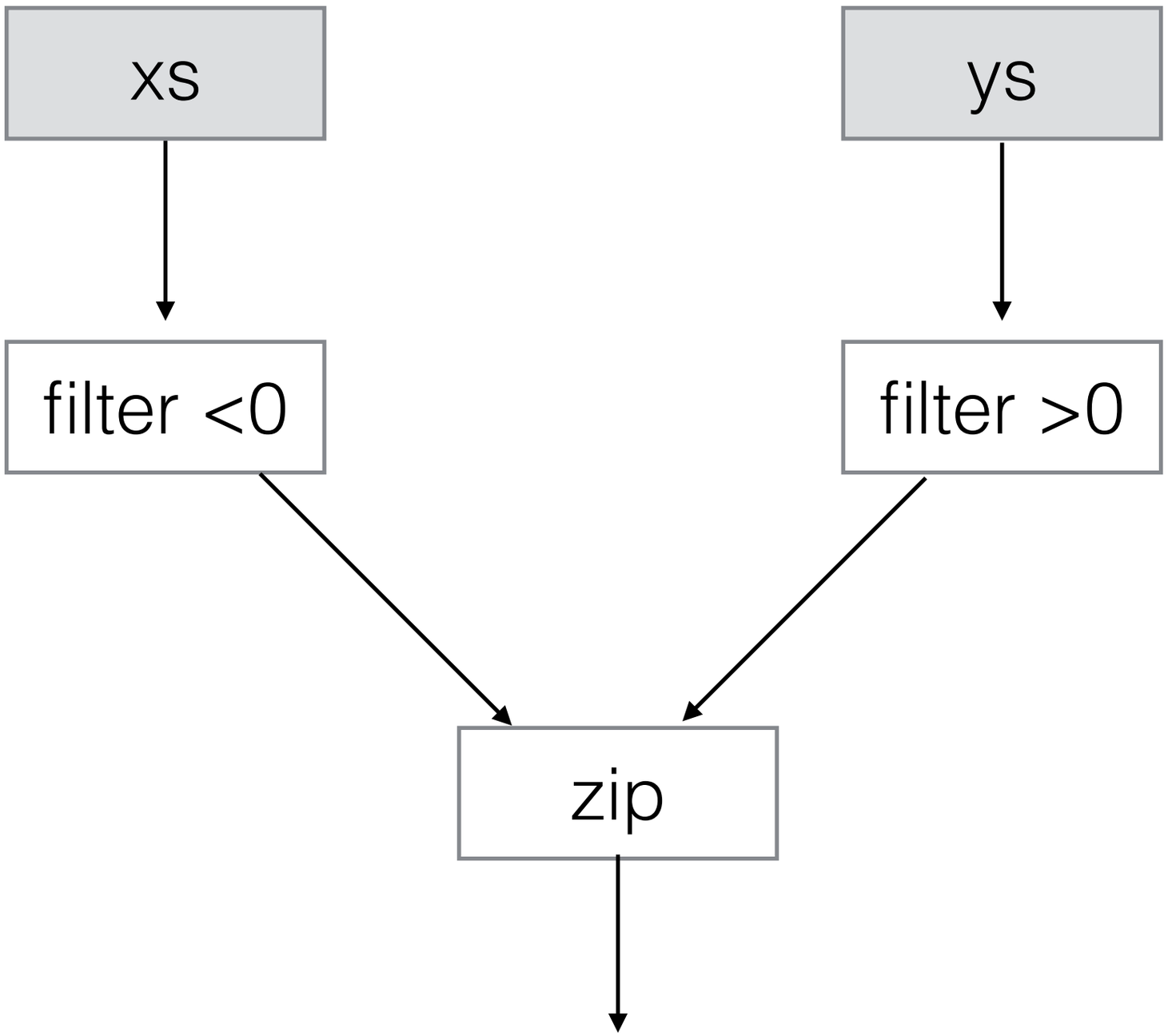


Fusion

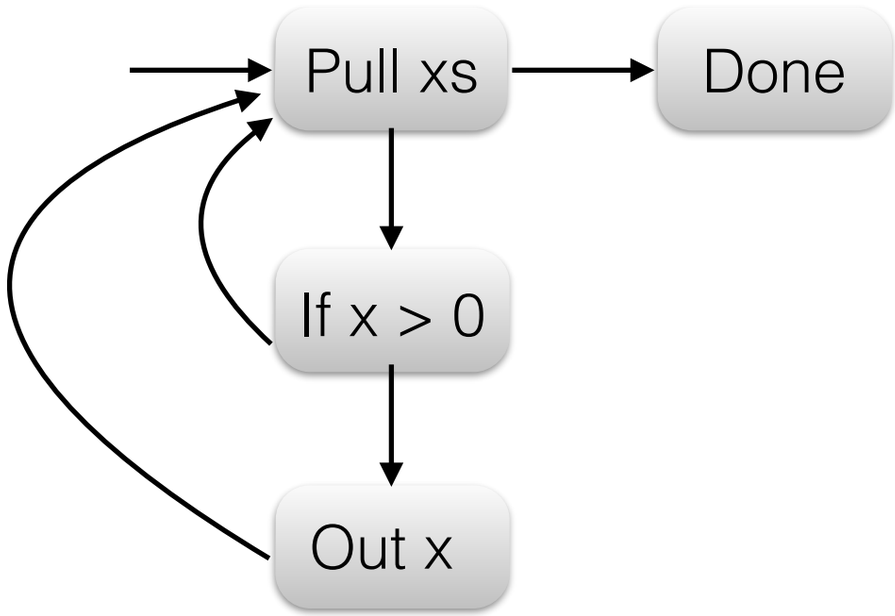
zipf again

```
zipf     (xs ys : Source Int)
  = let  xs'  = filter  (>0)     xs
         ys'  = filter  (<0)     ys
    in   zip xs' ys'
```

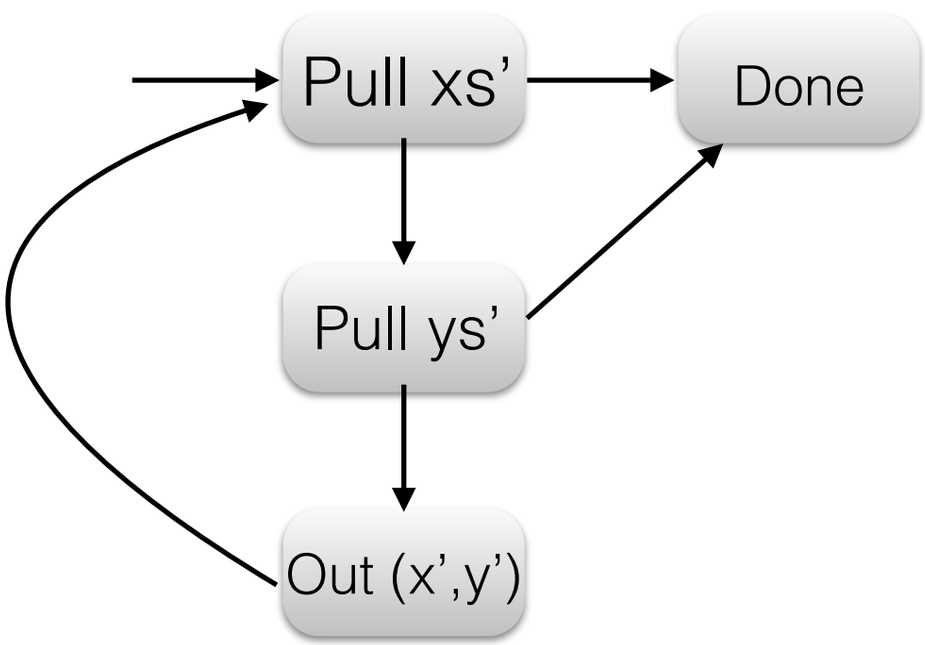
zipf again



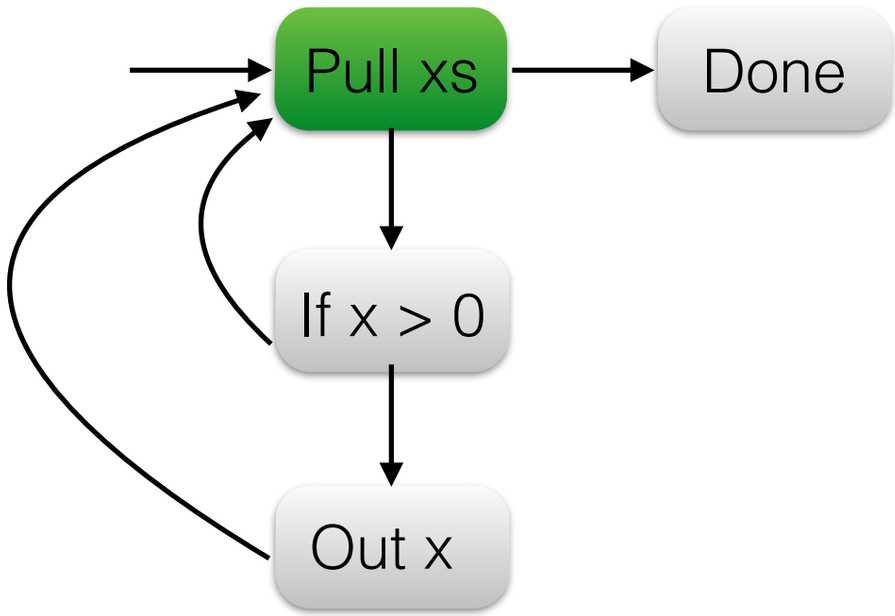
filter (>0) xs



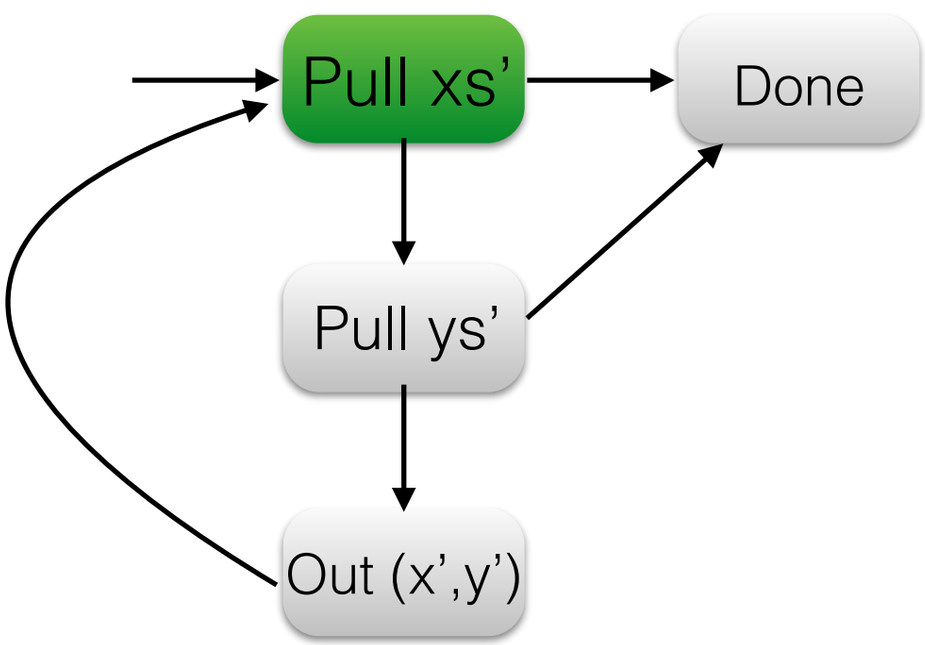
zip xs' ys'



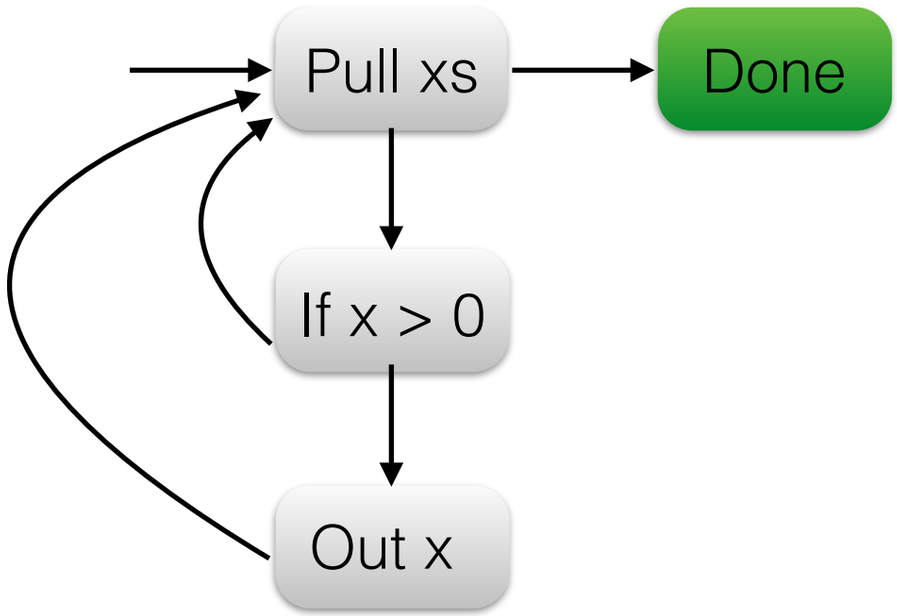
filter (>0) xs

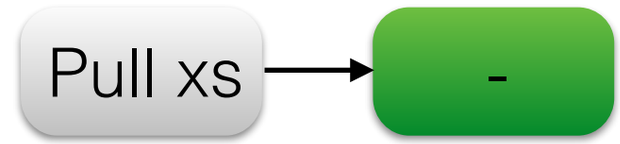


zip xs' ys'

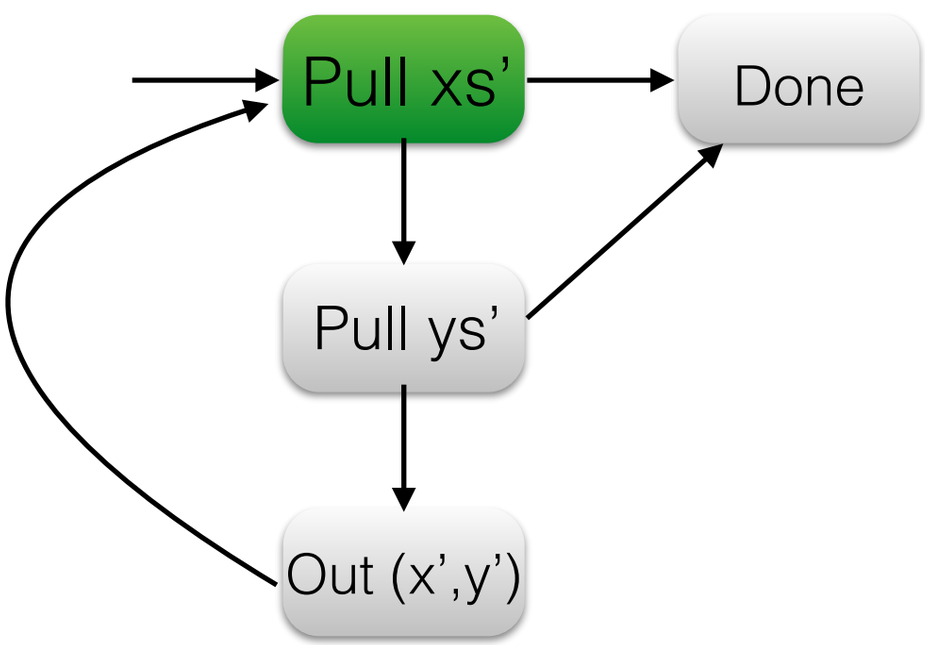


filter (>0) xs

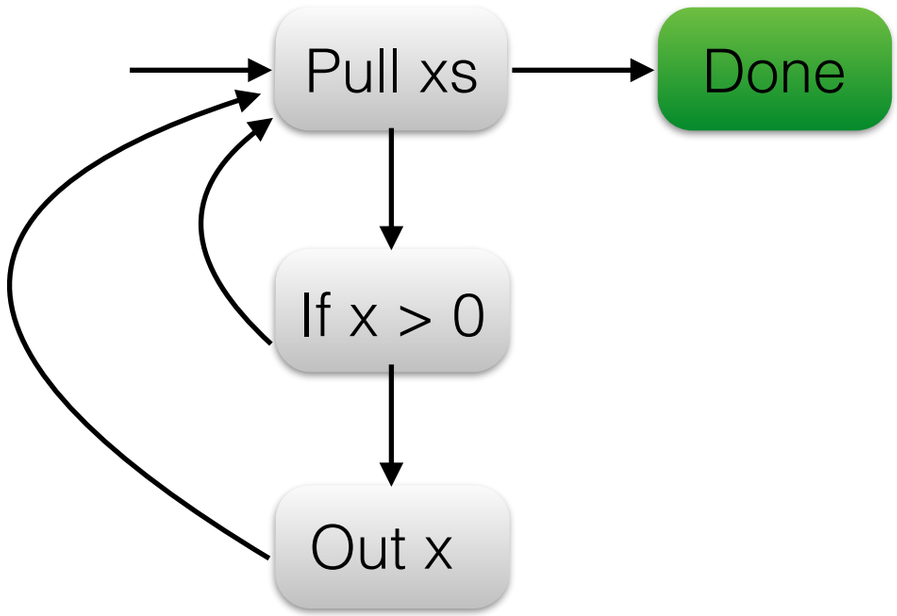




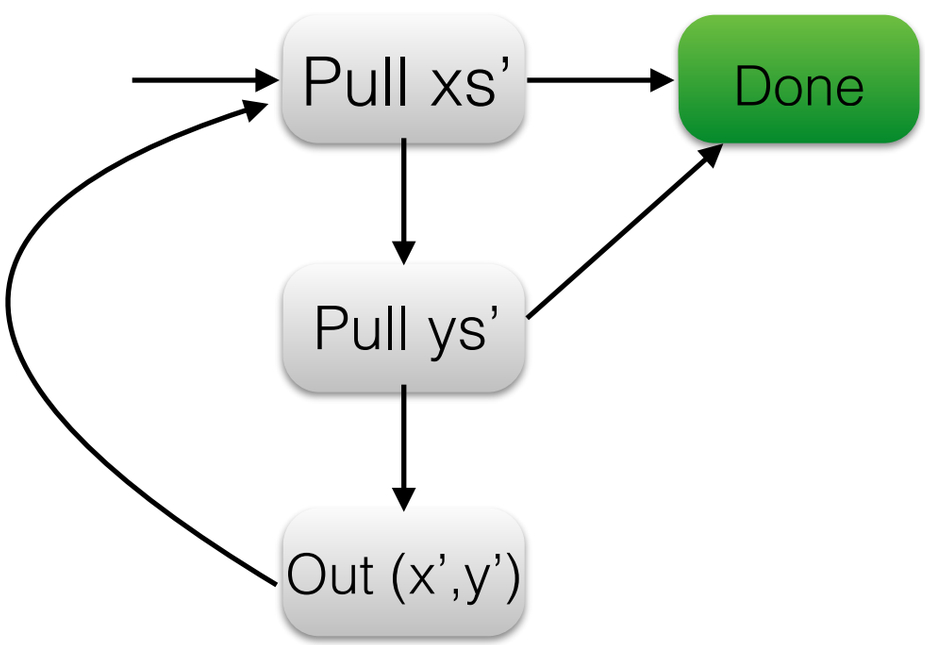
zip xs' ys'



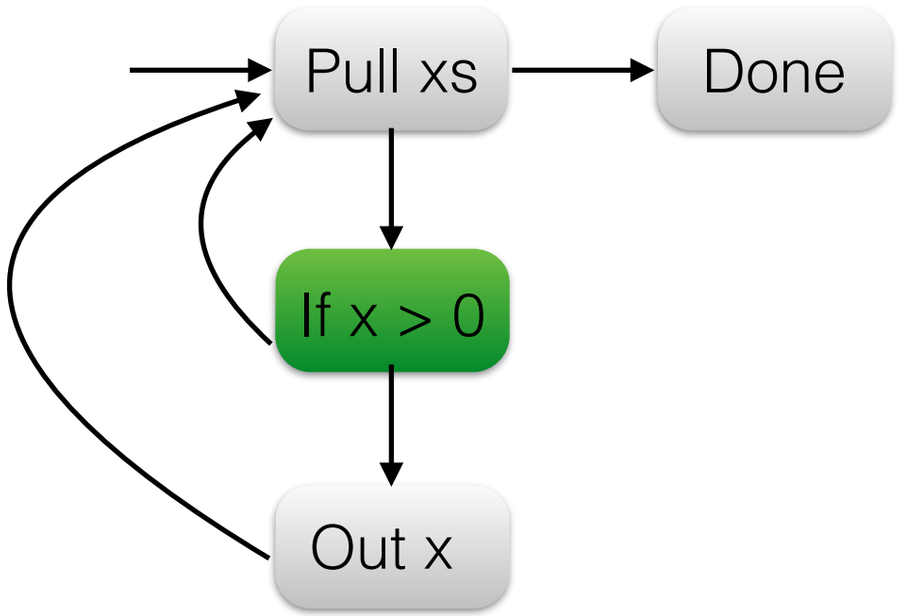
filter (>0) xs

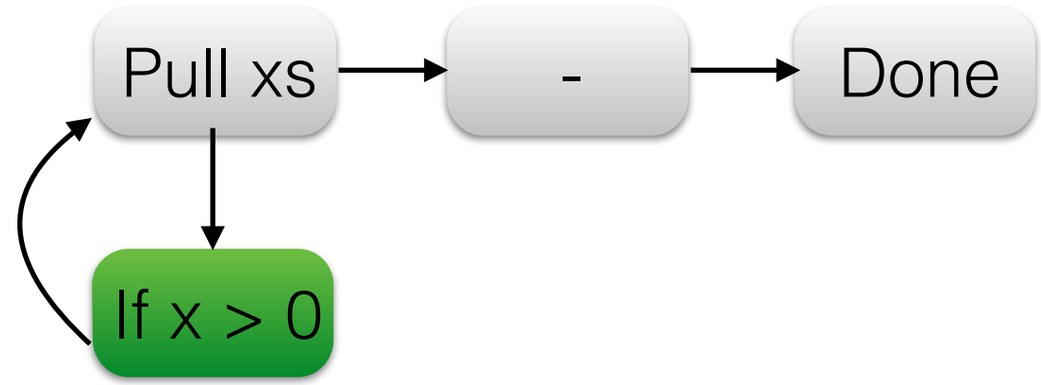


zip xs' ys'

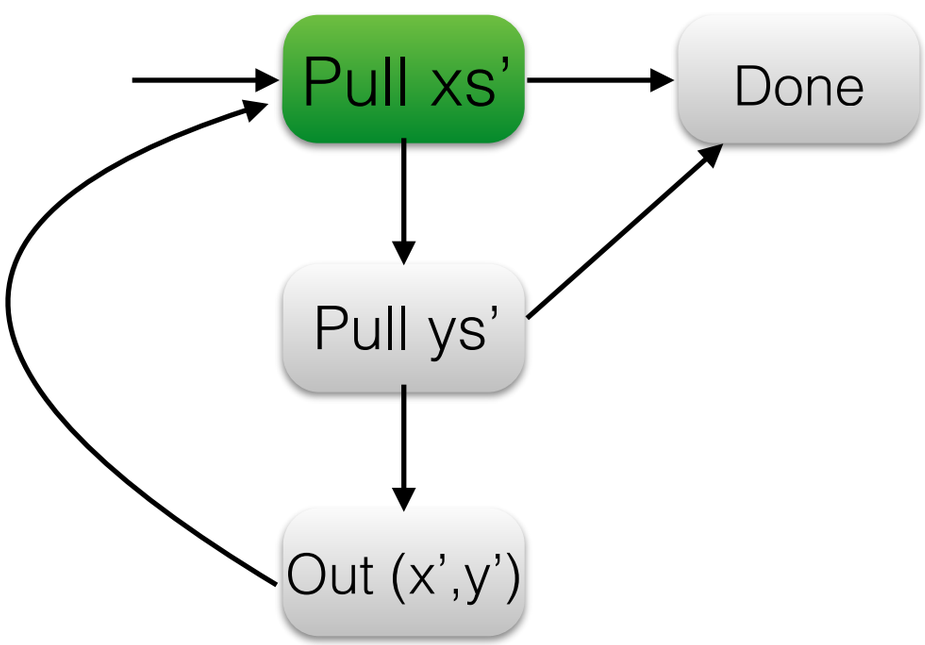


filter (>0) xs

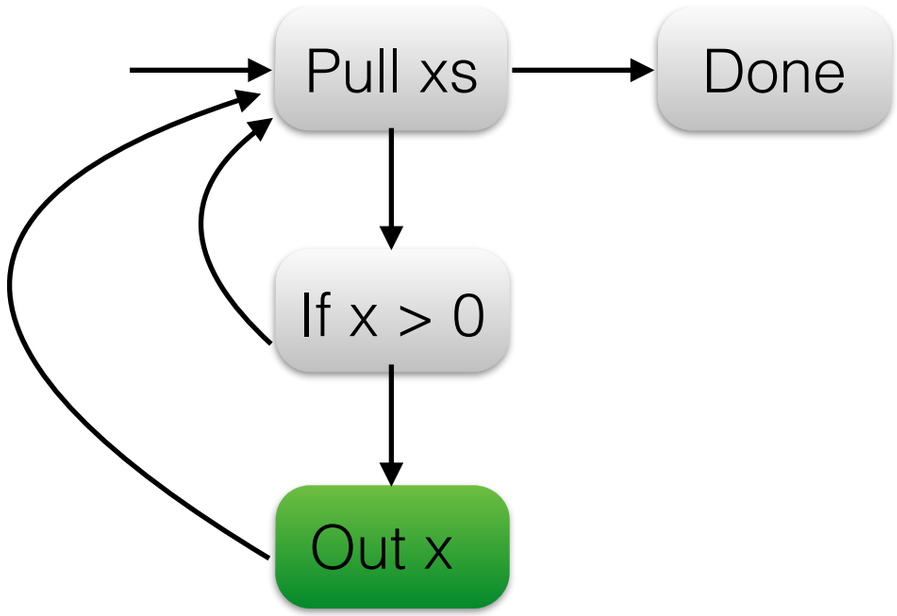


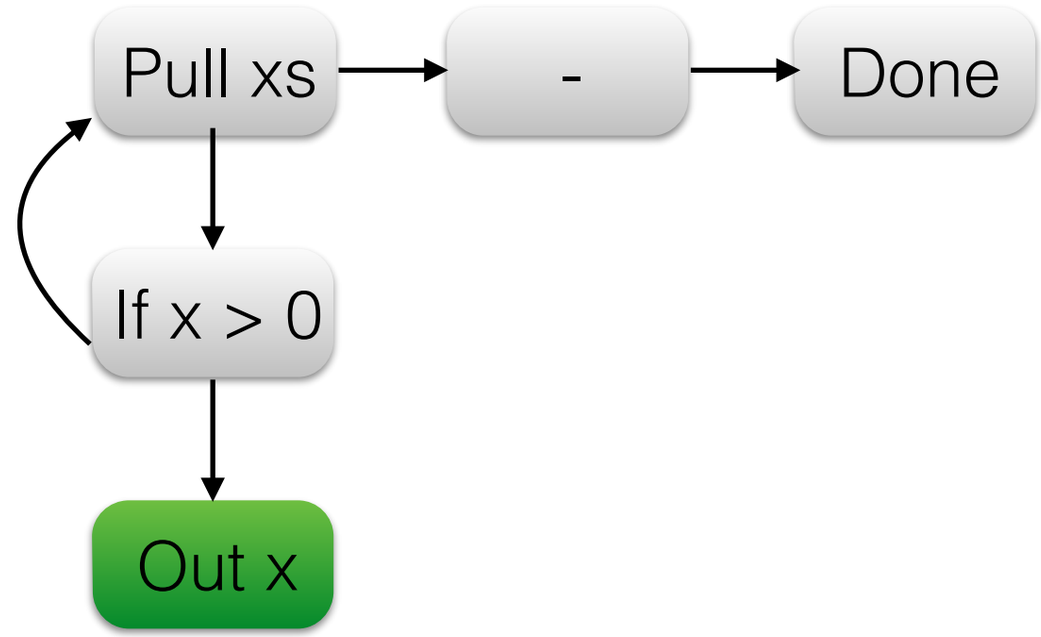


zip xs' ys'

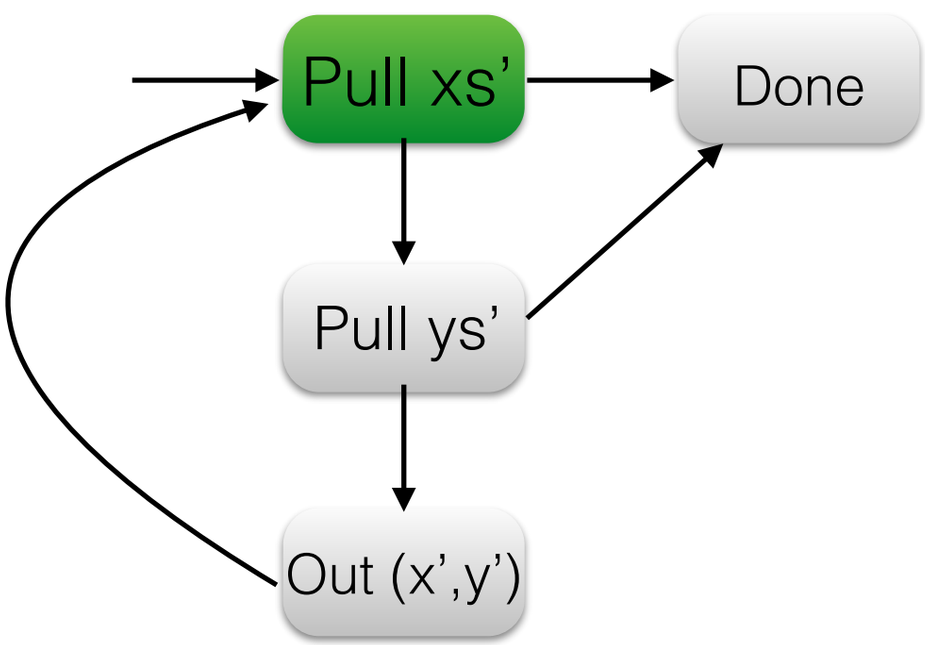


filter (>0) xs

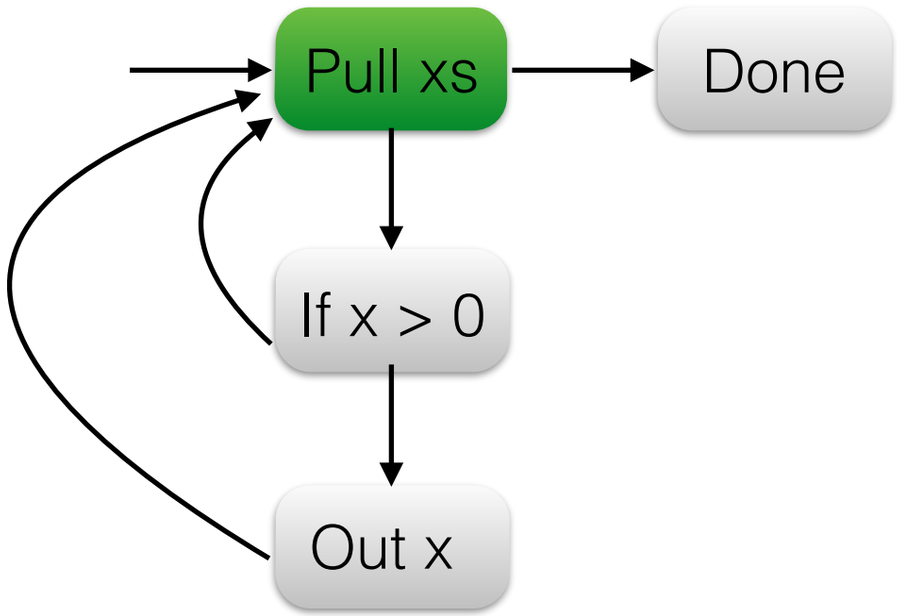




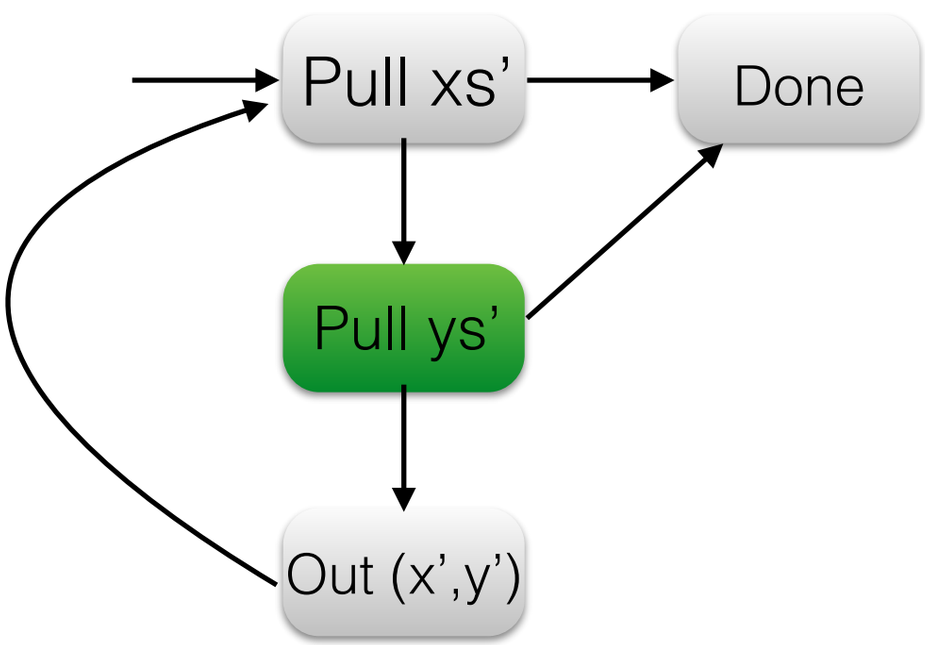
zip xs' ys'

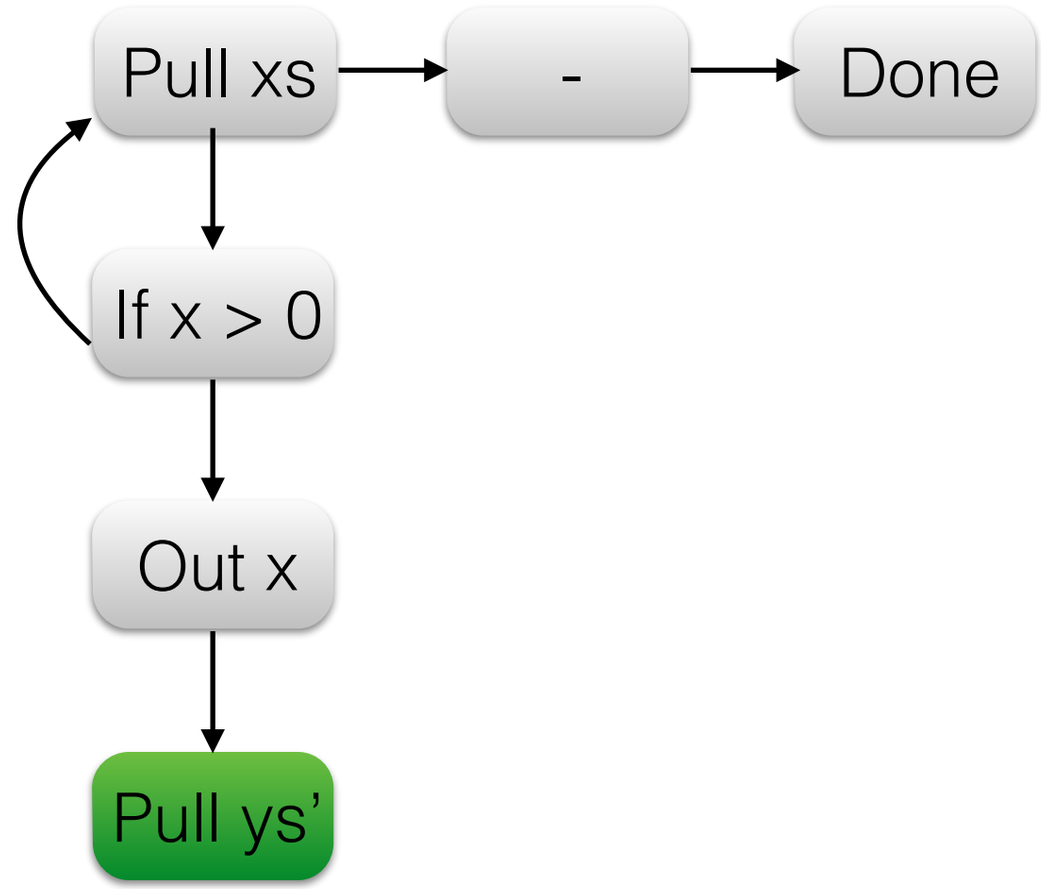


filter (>0) xs

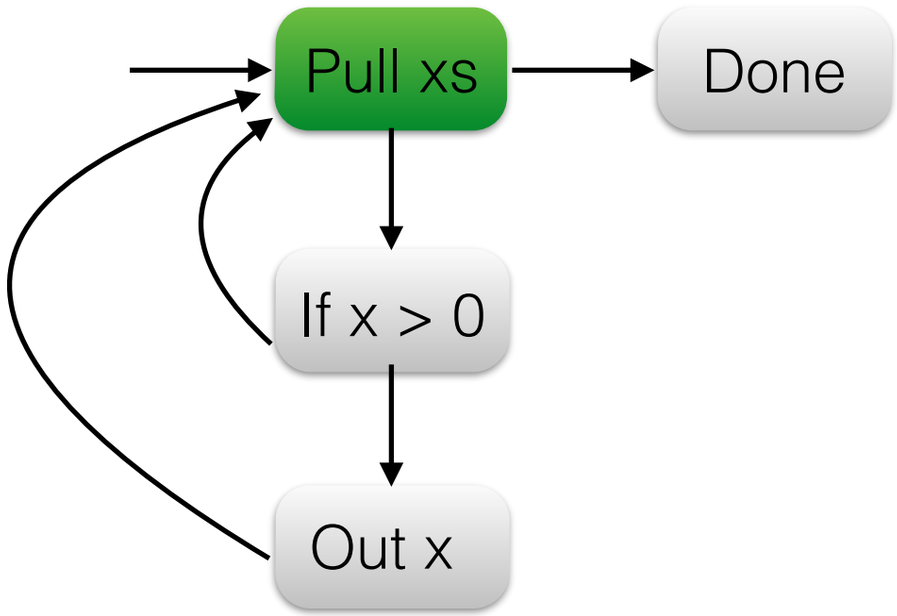


zip xs' ys'

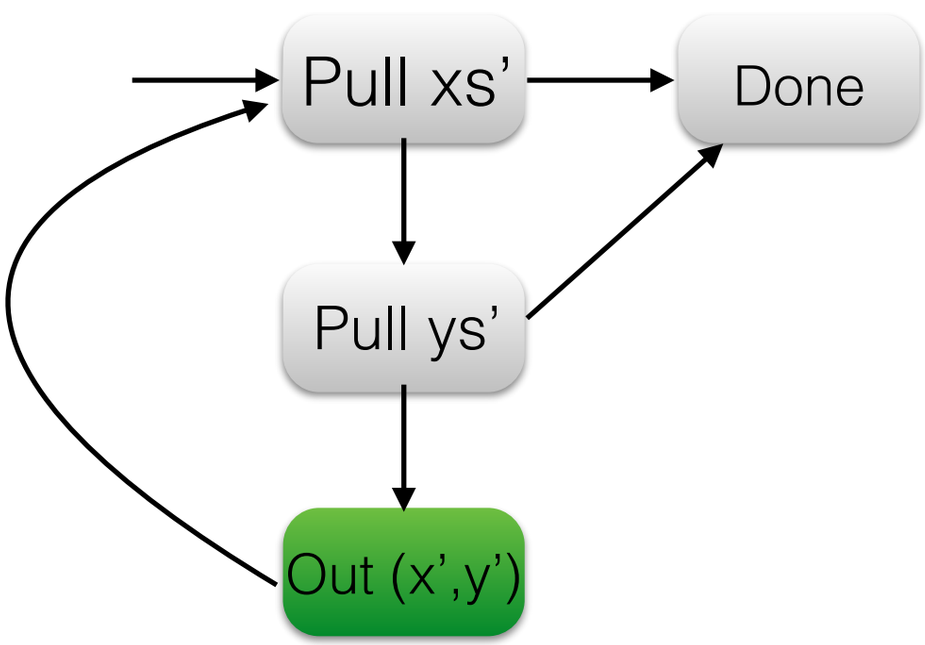


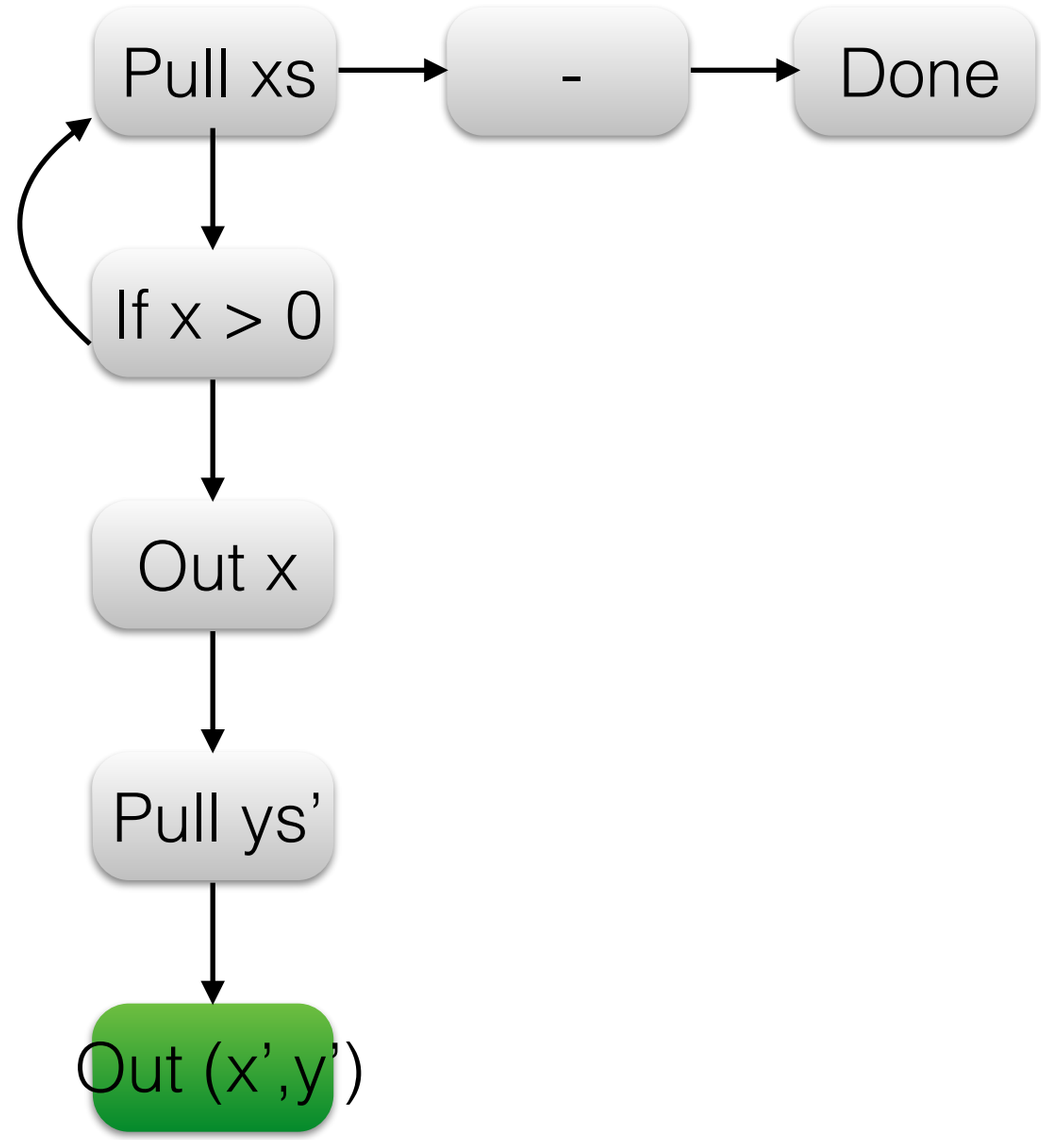


filter (>0) xs

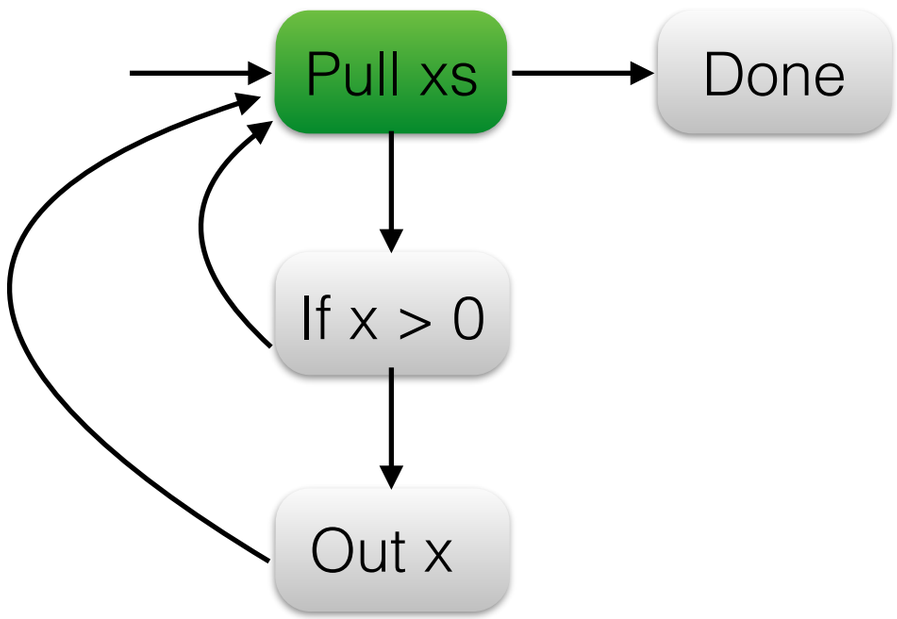


zip xs' ys'

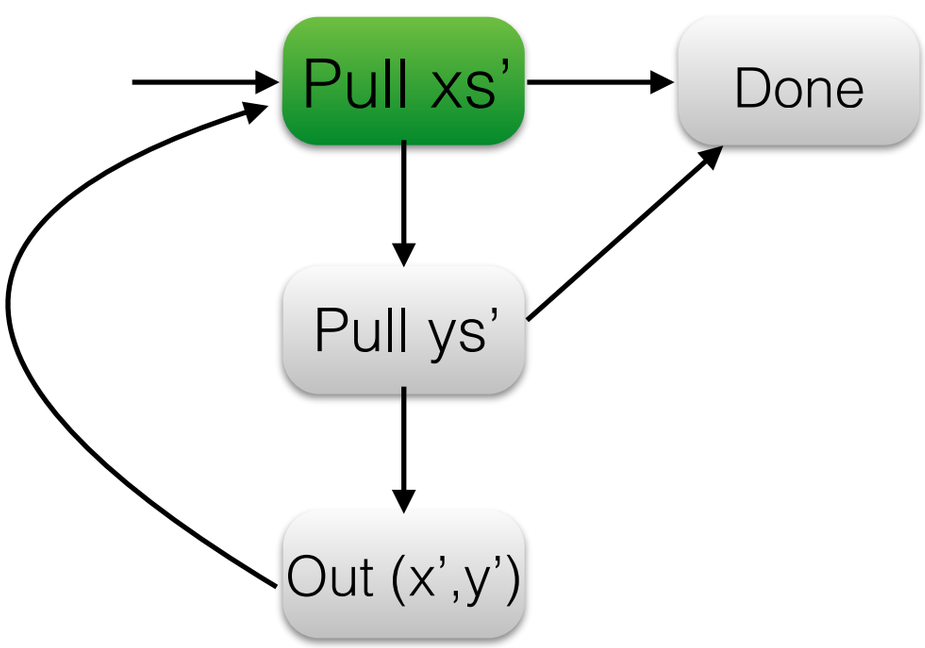


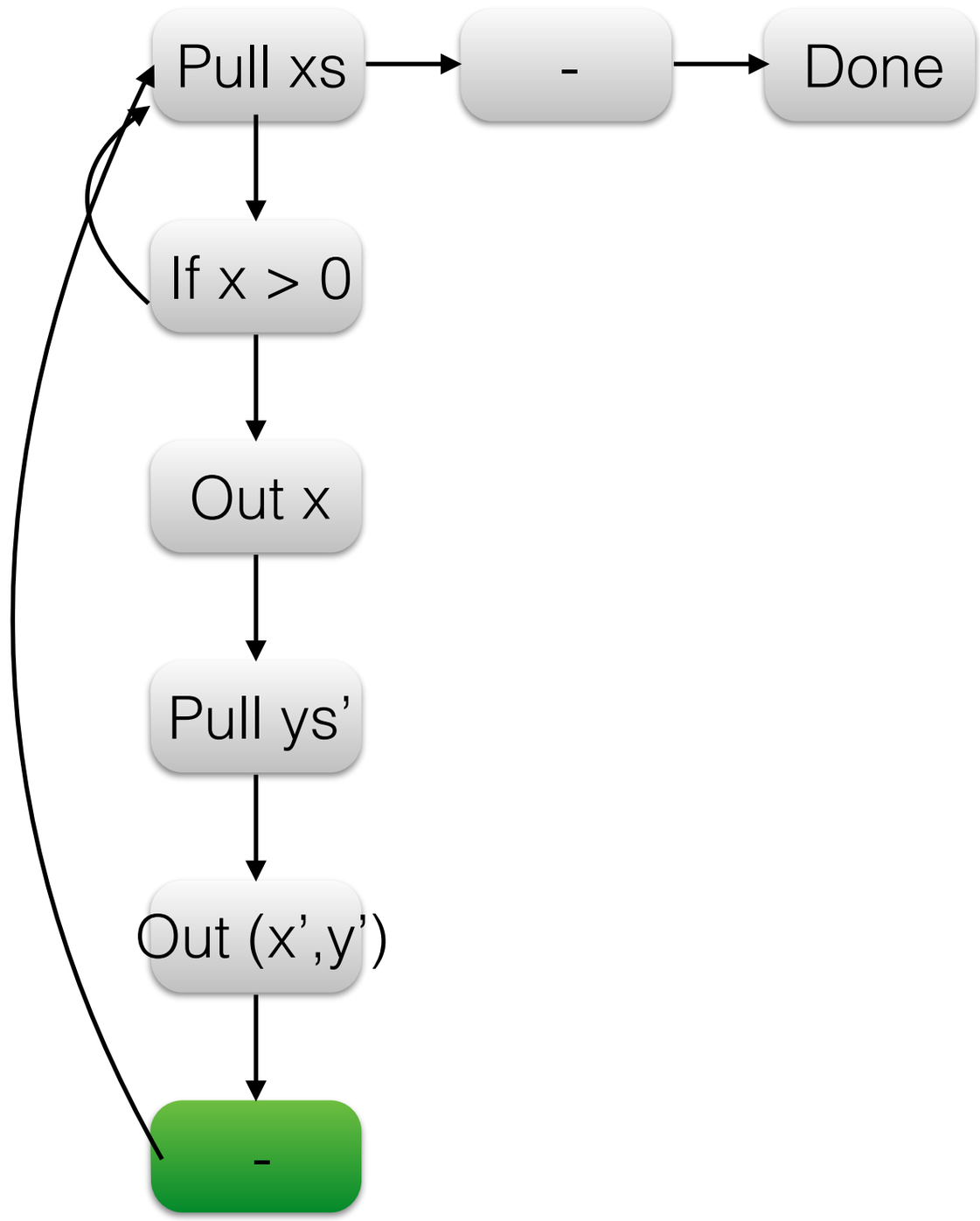


filter (>0) xs

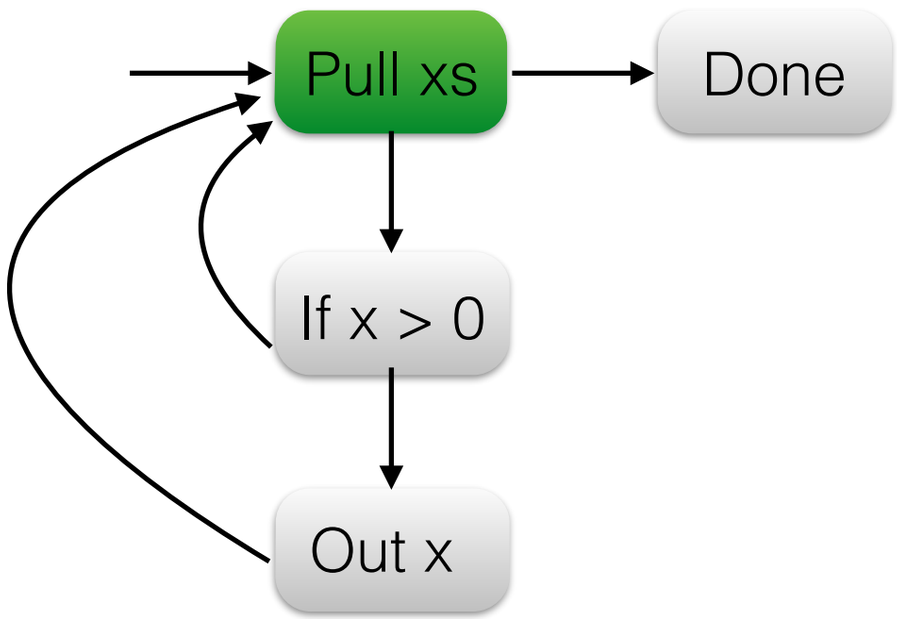


zip xs' ys'

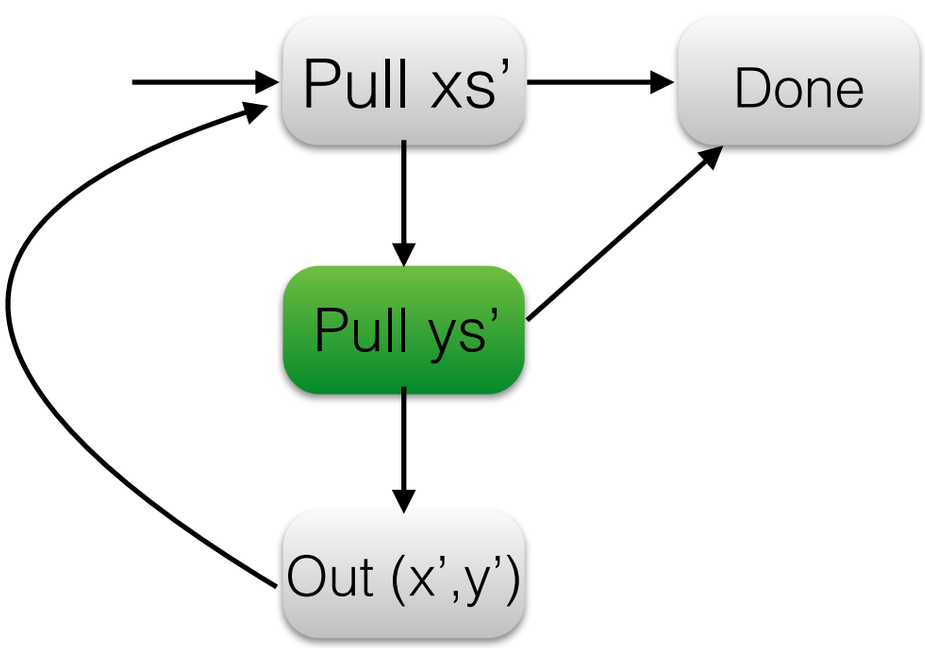


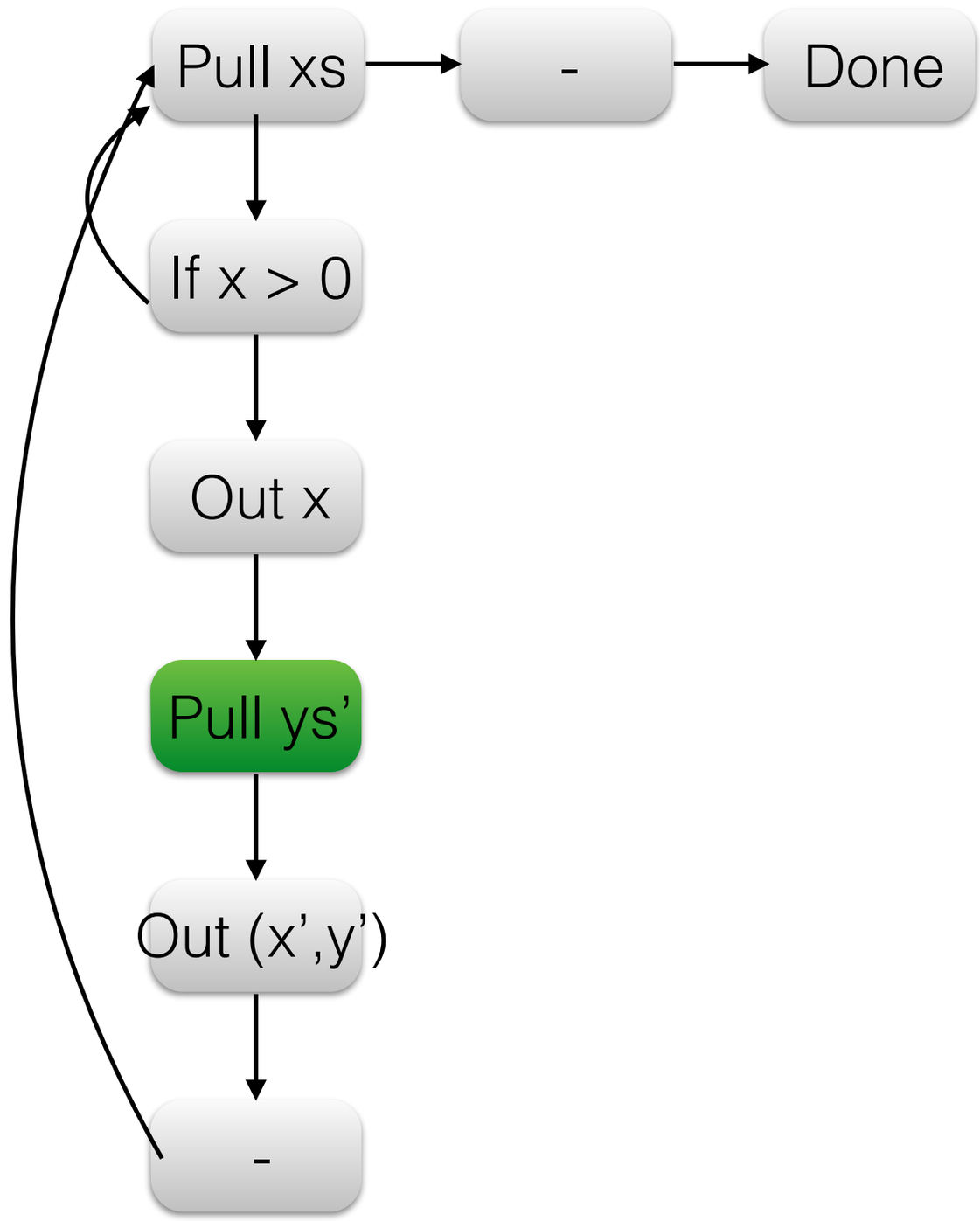


filter (>0) xs

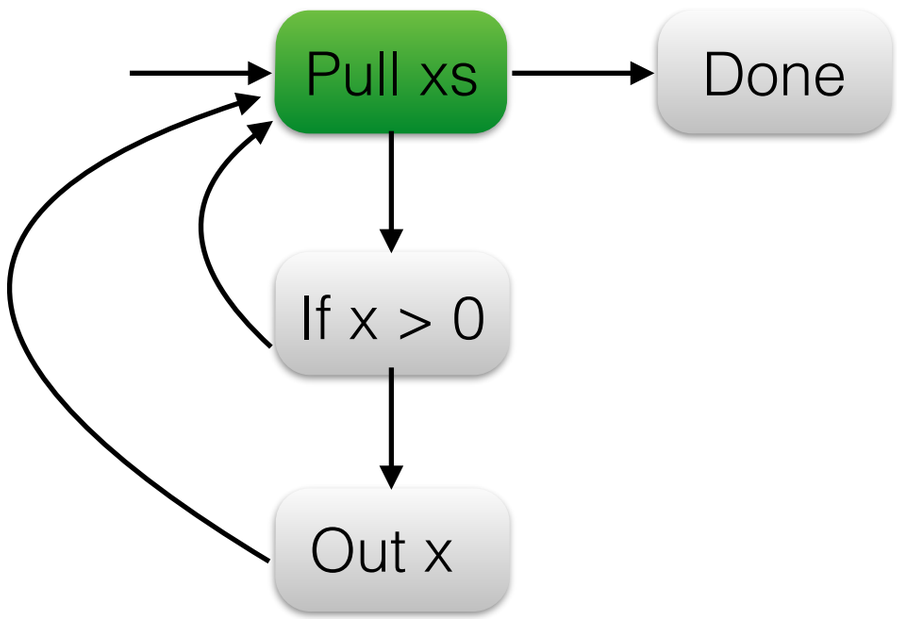


zip xs' ys'

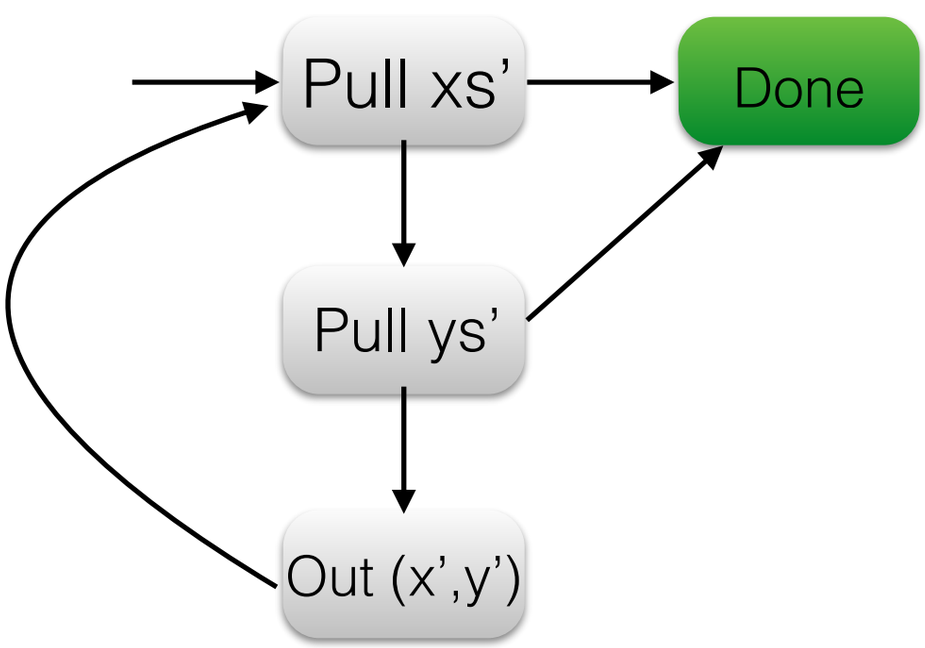


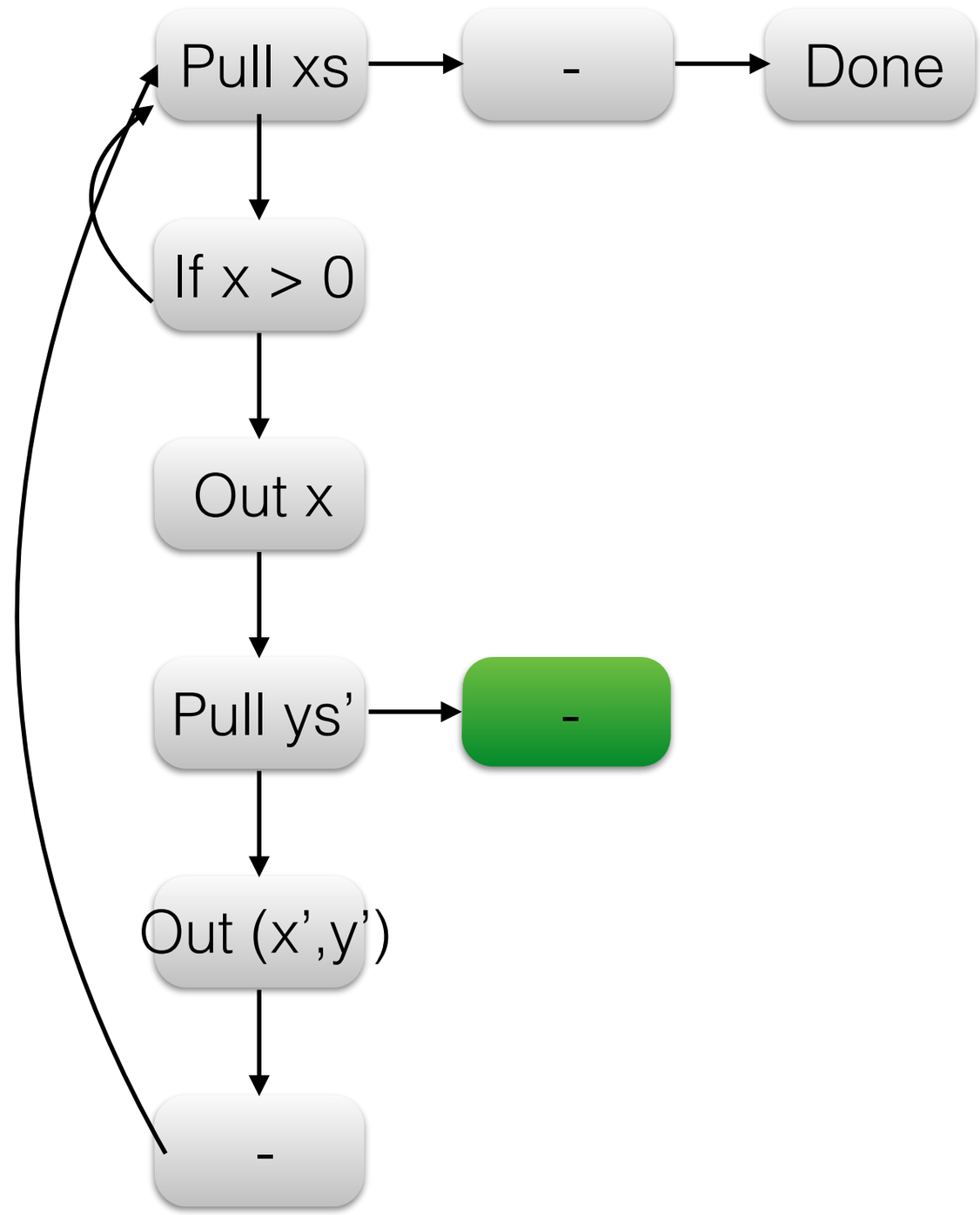


filter (>0) xs

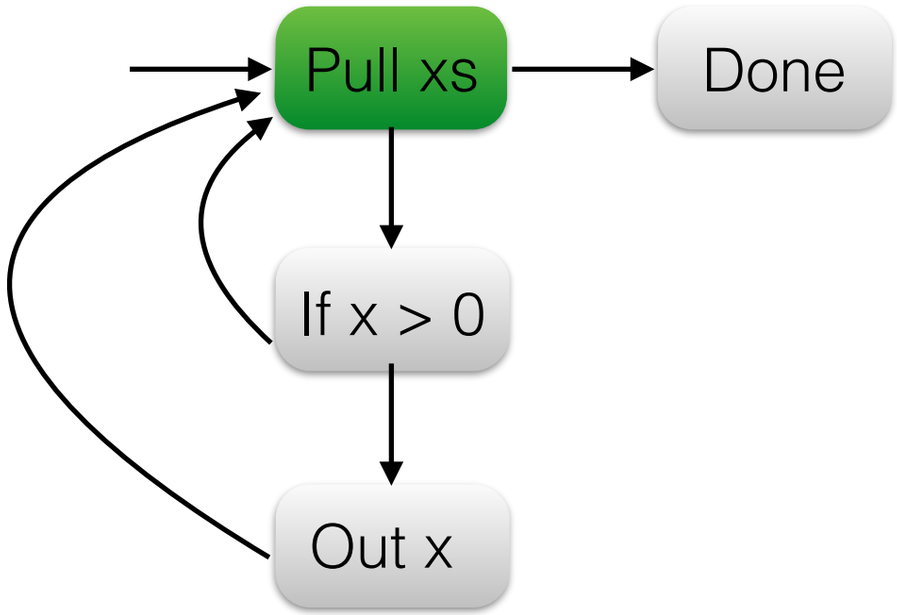


zip xs' ys'

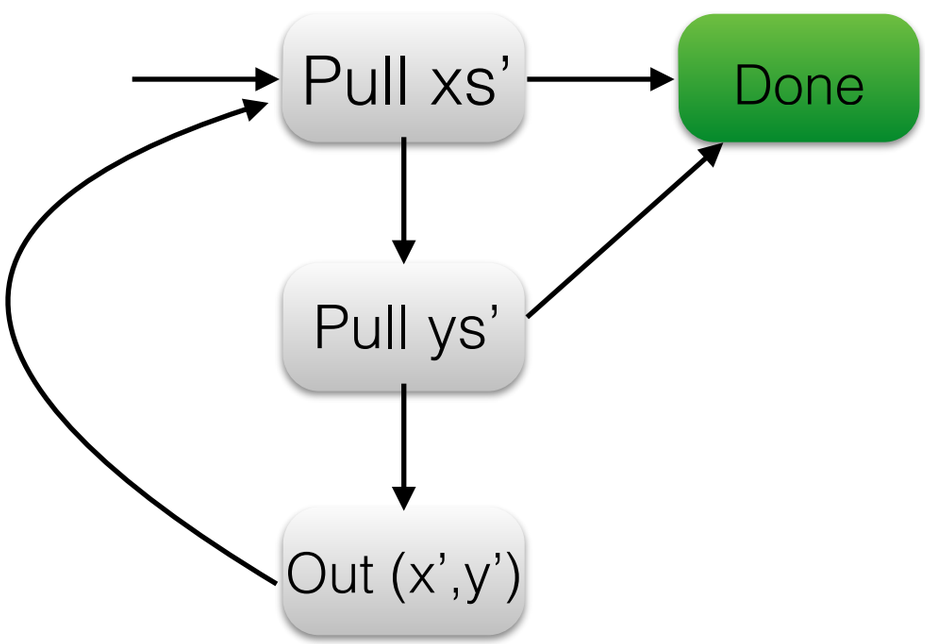


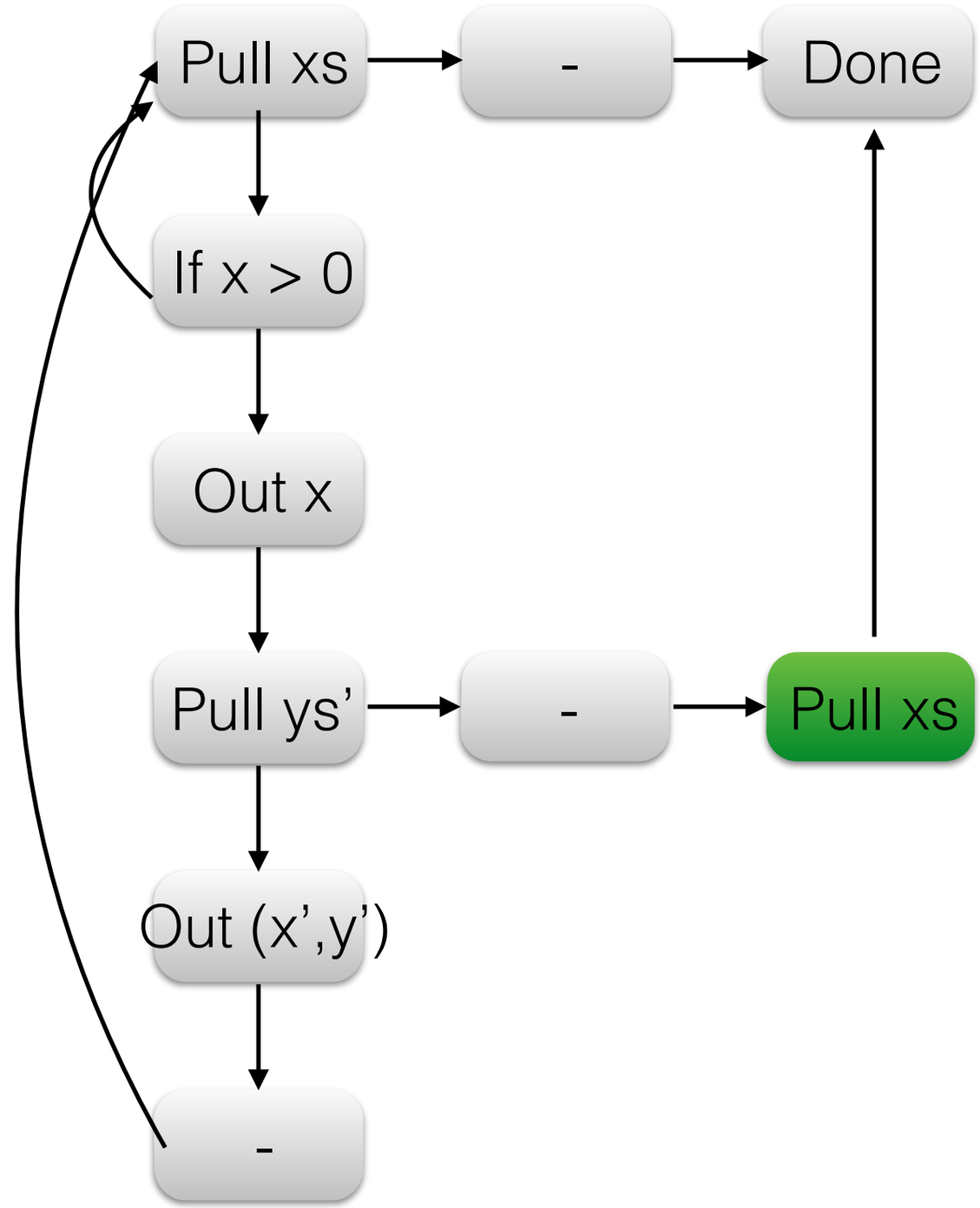


filter (>0) xs

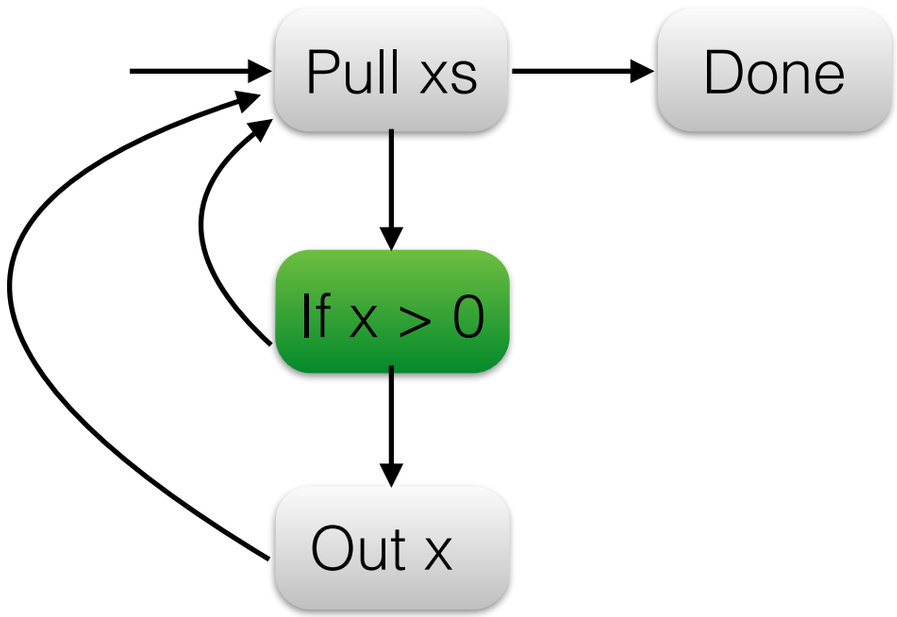


zip xs' ys'

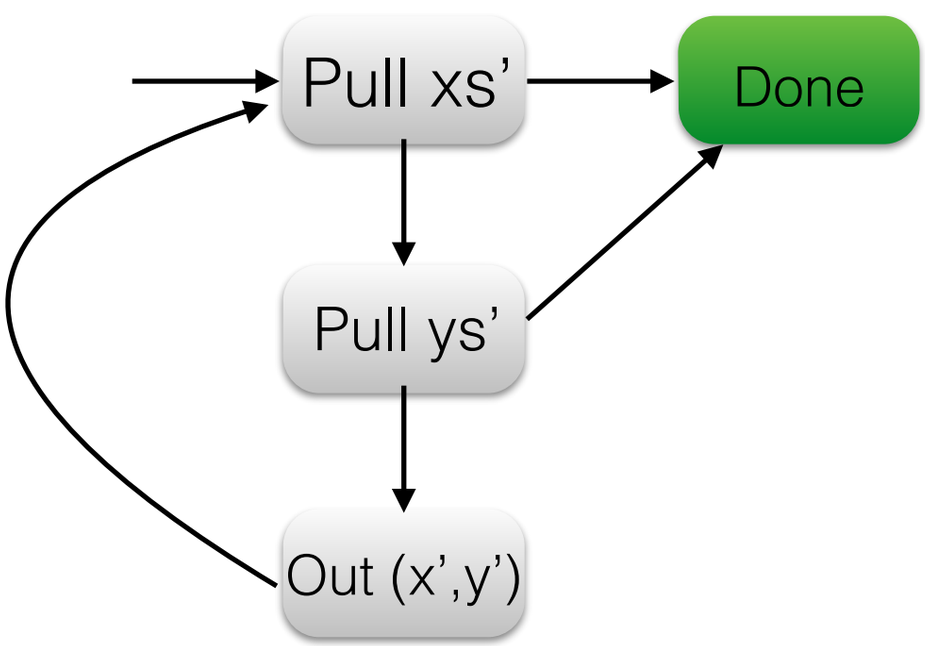


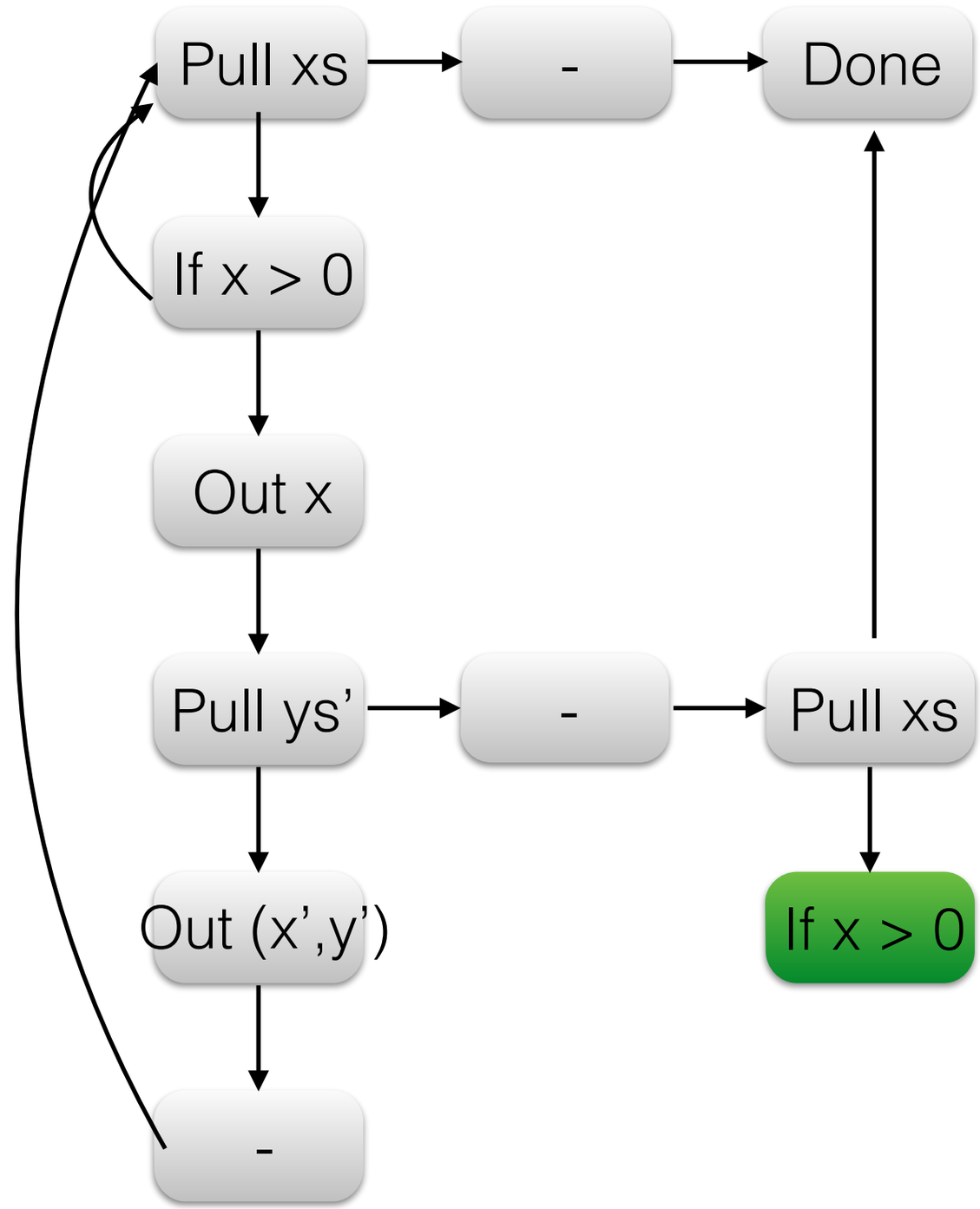


filter (>0) xs

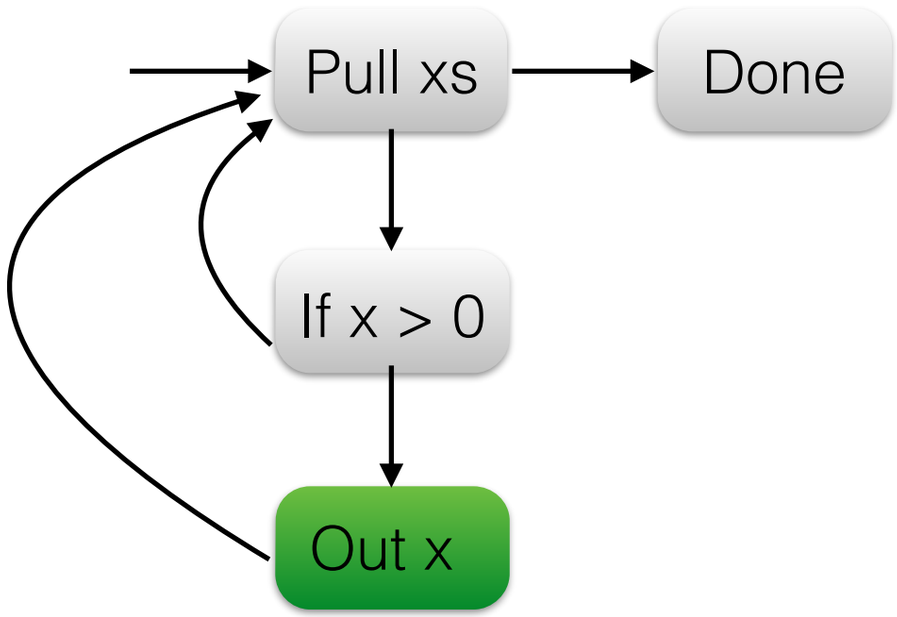


zip xs' ys'

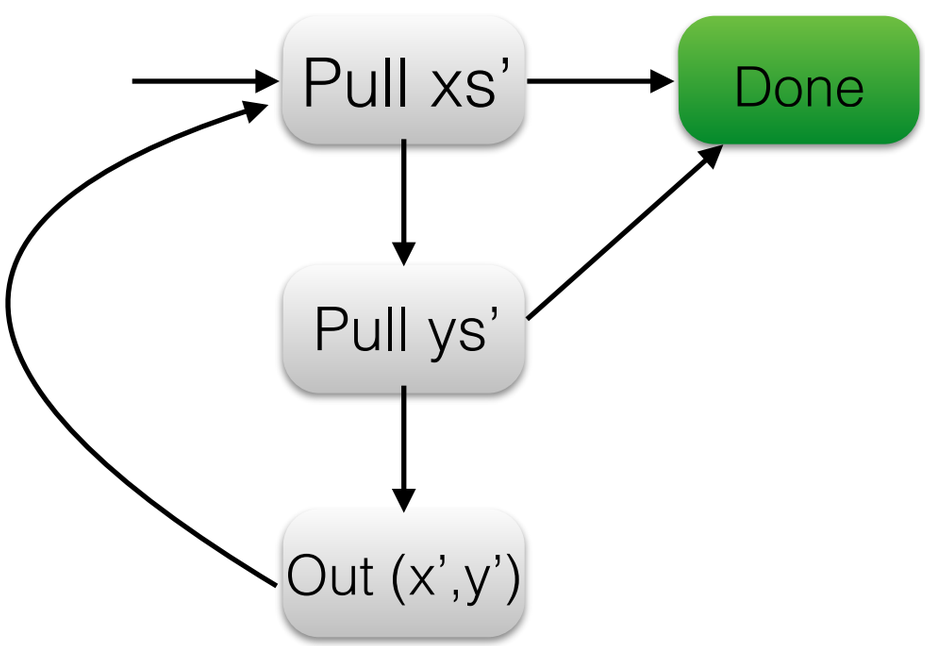


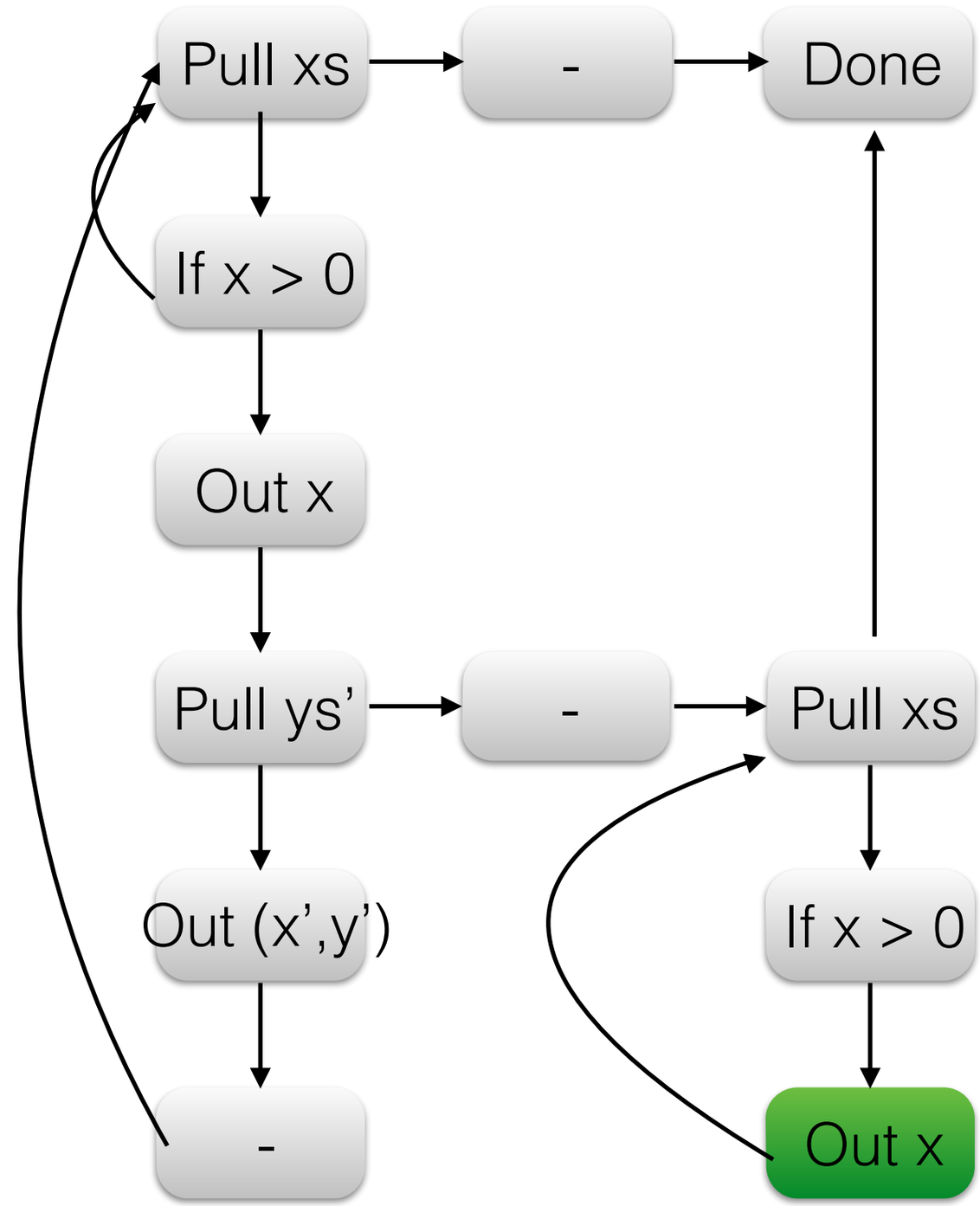


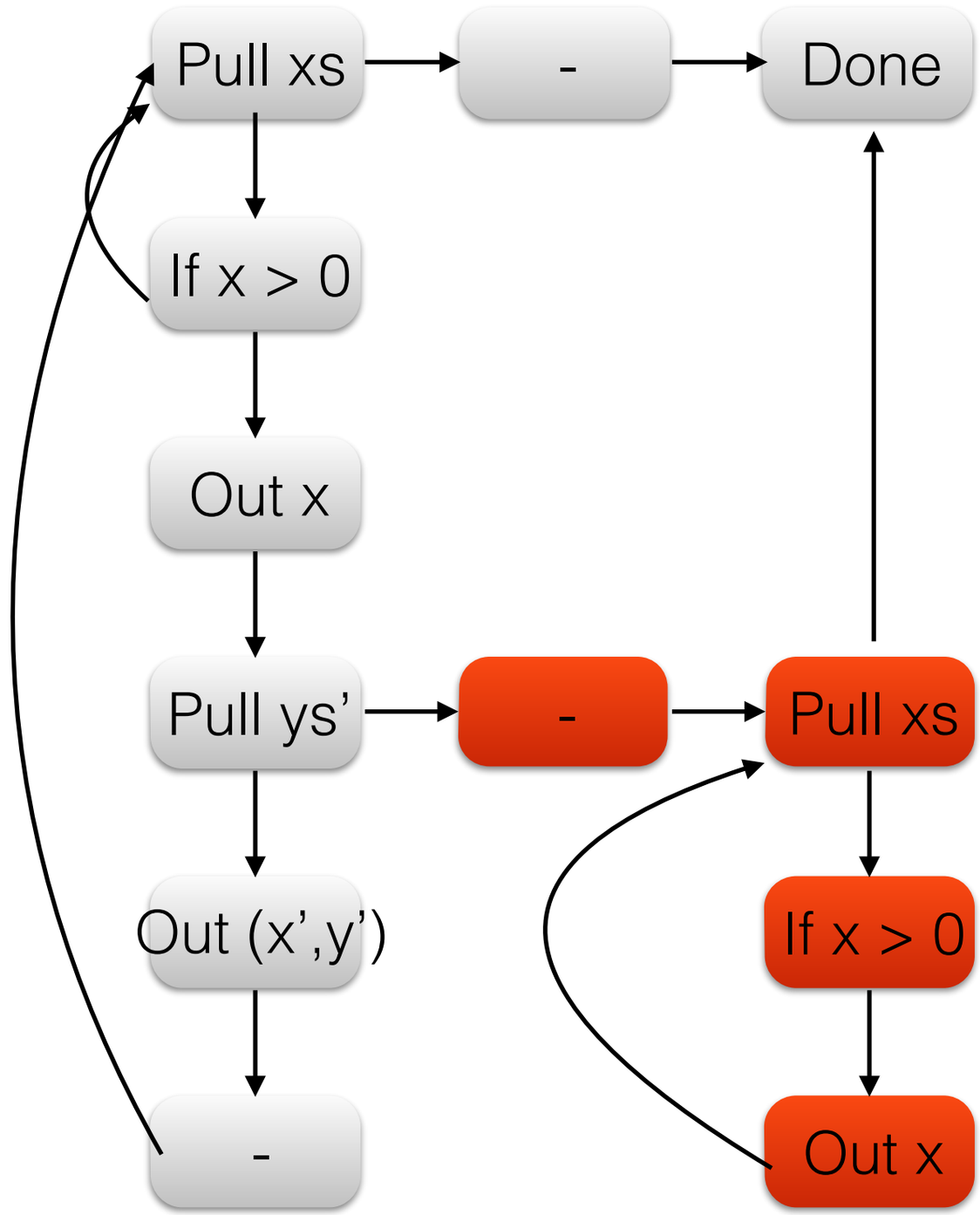
filter (>0) xs



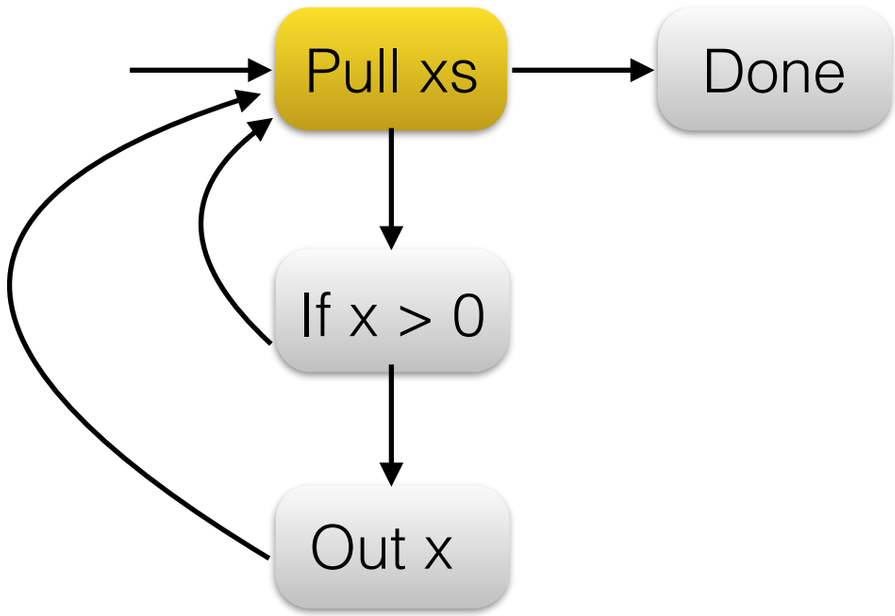
zip xs' ys'



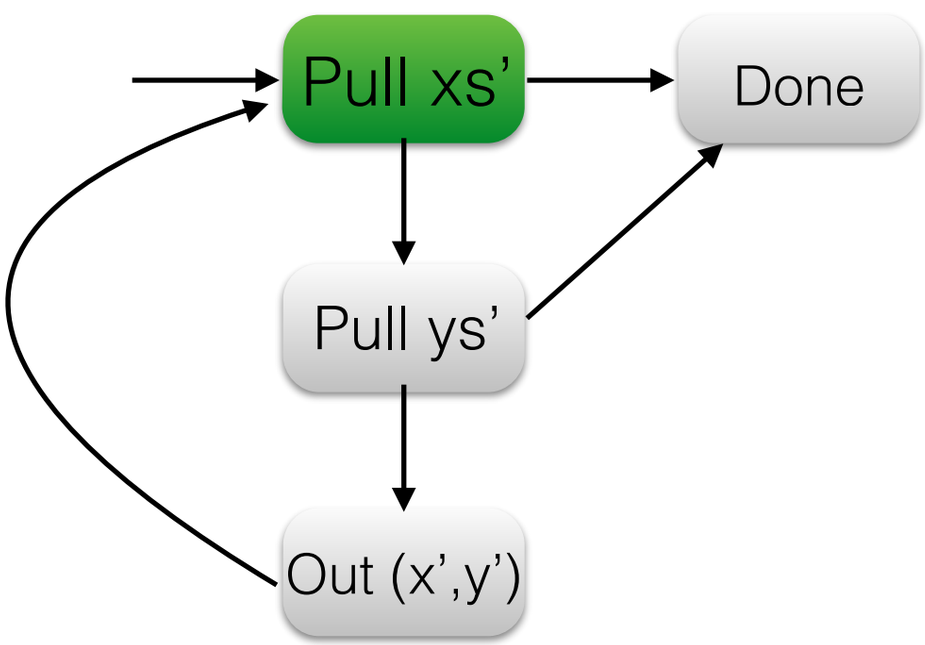



Pull xs
-
Done
If x > 0
Out x
Pull ys'
-
Pull xs
Out (x',y')
If x > 0
-
Out x

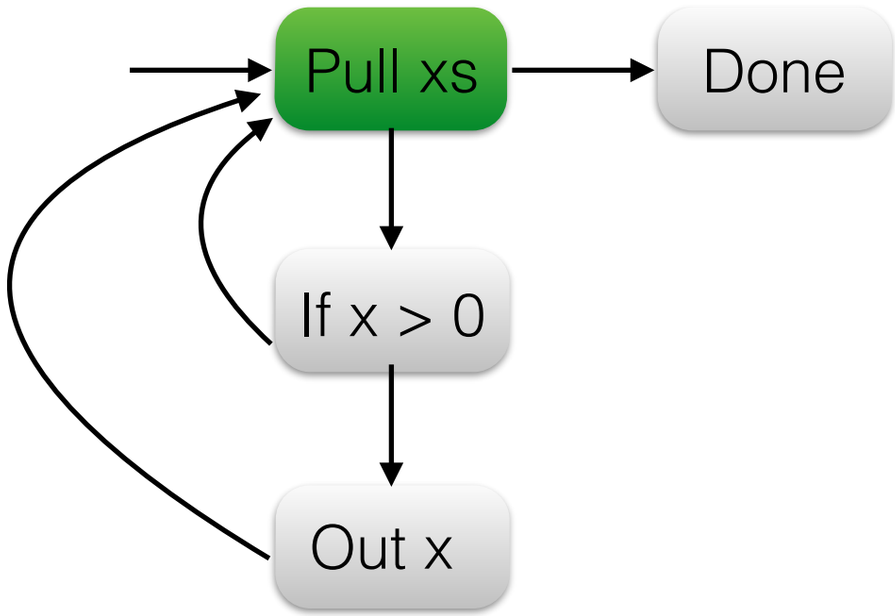
filter (>0) xs



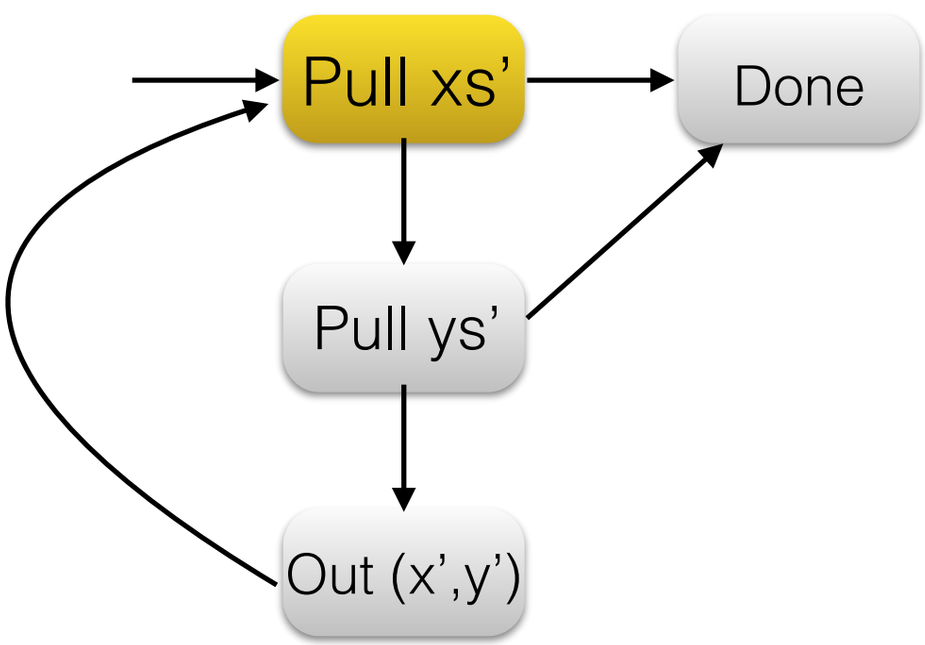
zip xs' ys'



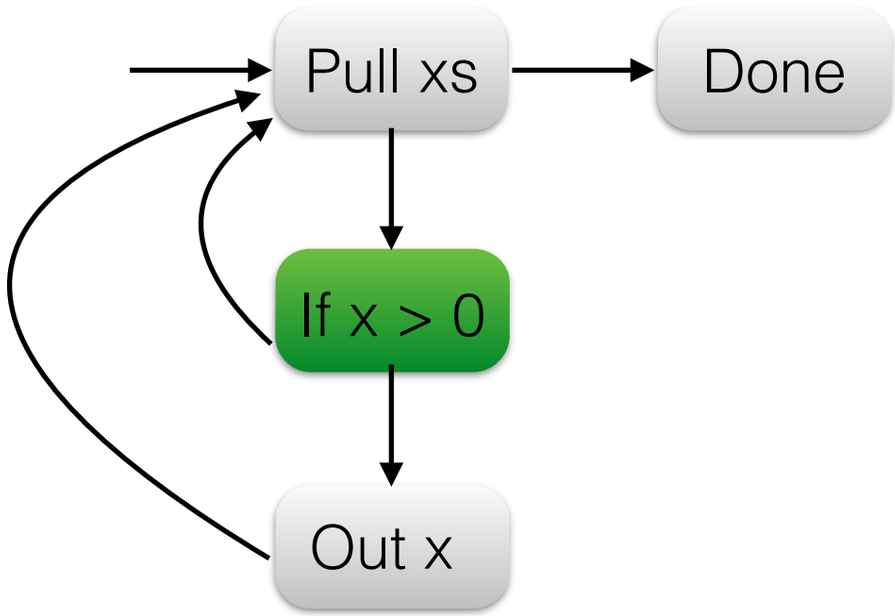
filter (>0) xs

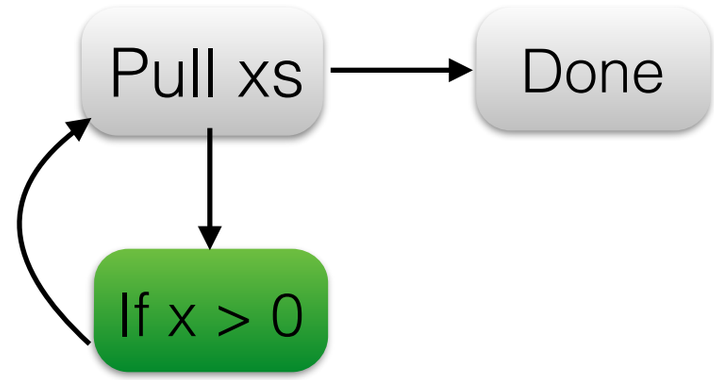


zip xs' ys'

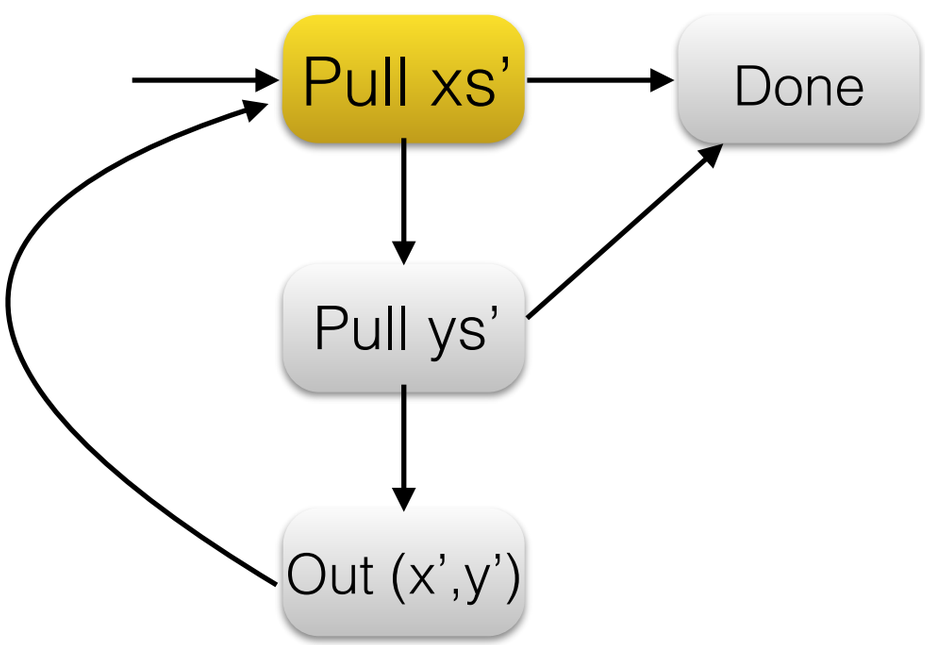


filter (>0) xs

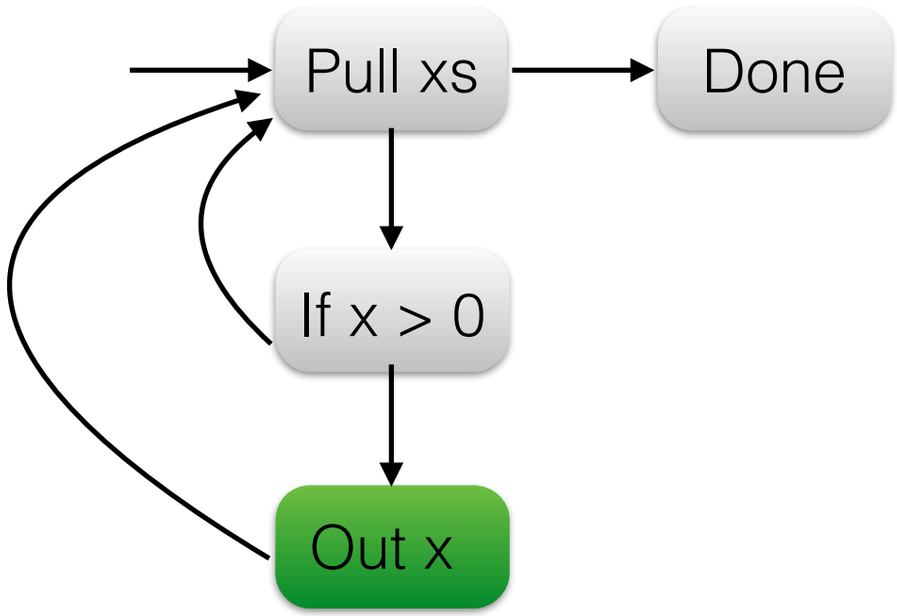


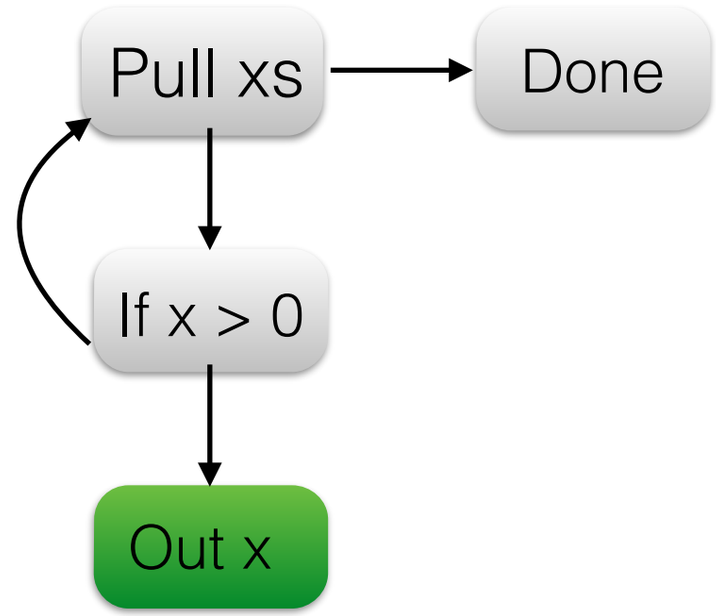


zip xs' ys'

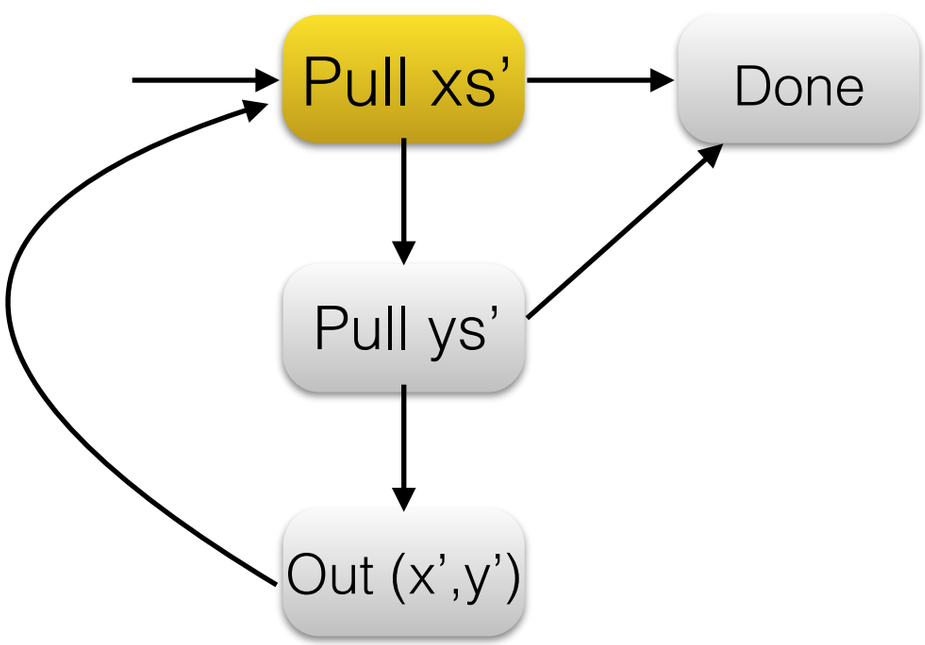


filter (>0) xs

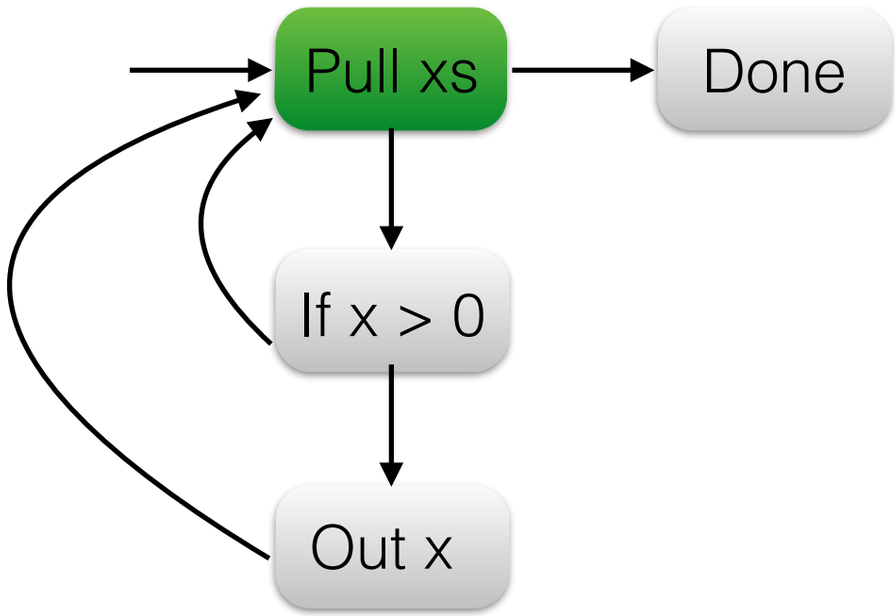


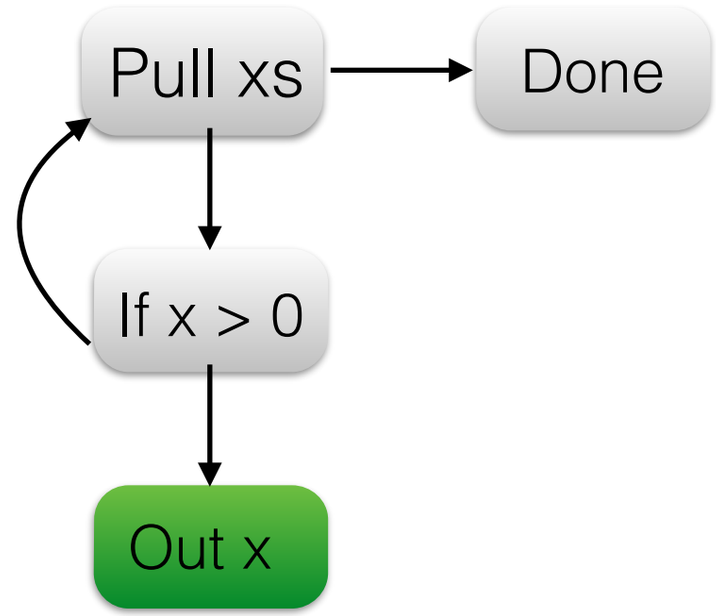


zip xs' ys'

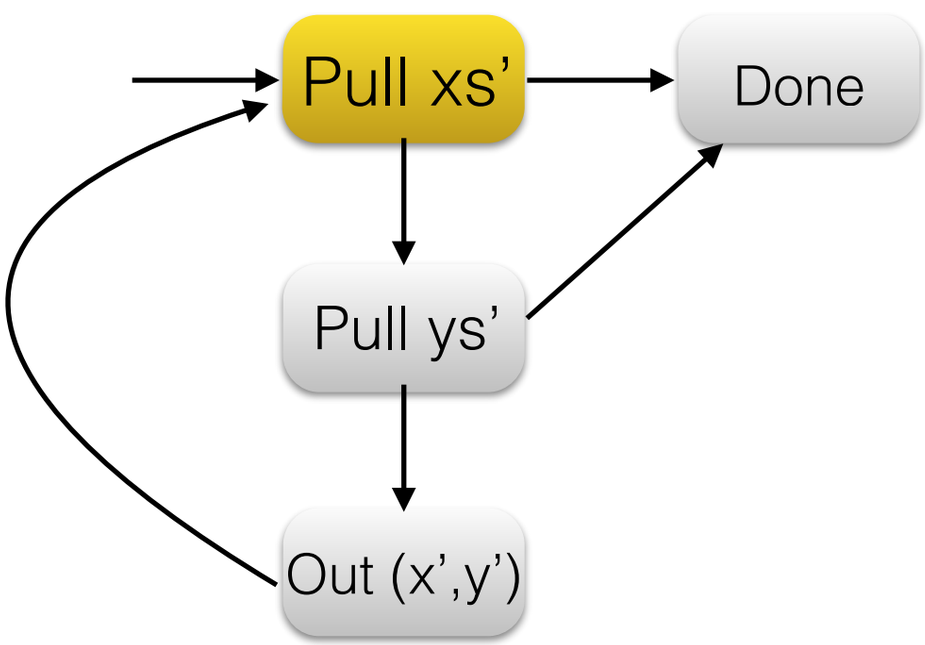


filter (>0) xs

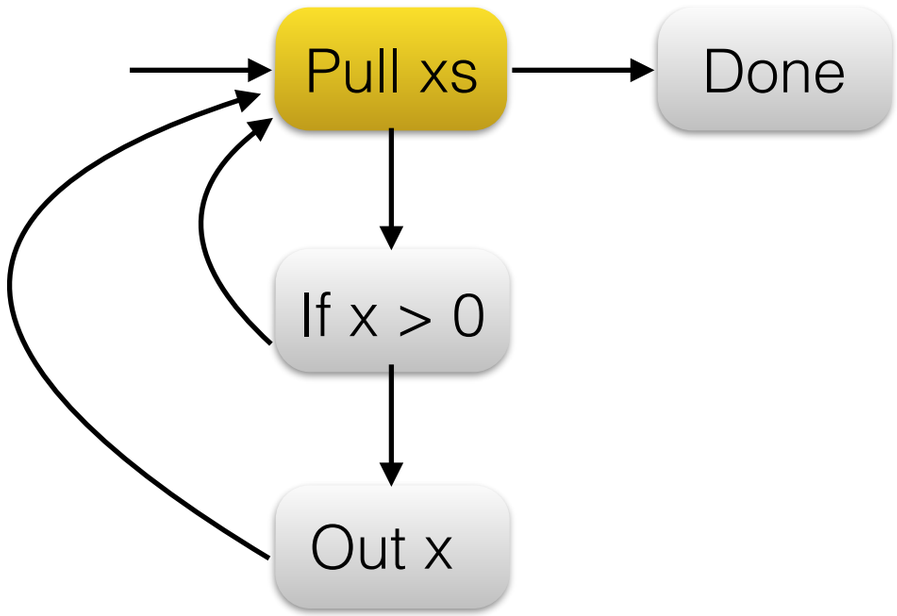


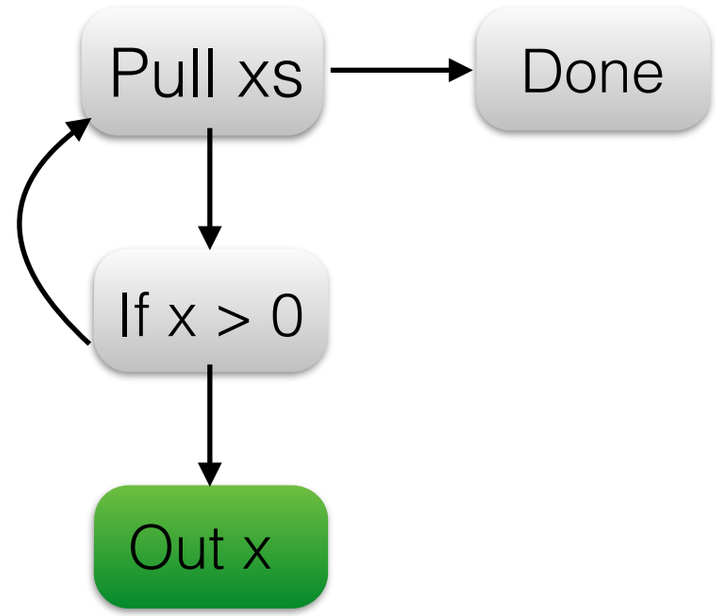


zip xs' ys'

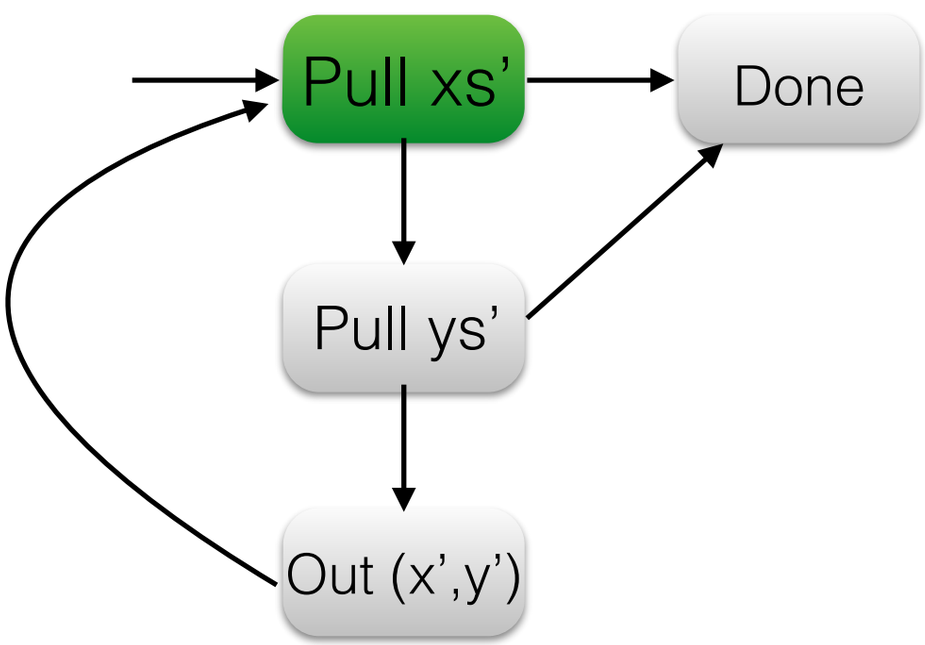


filter (>0) xs

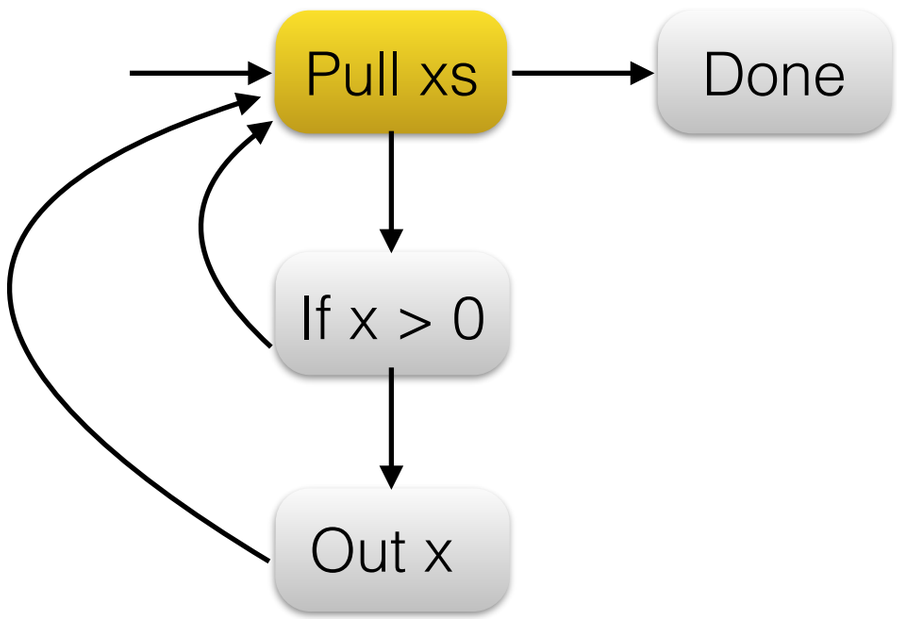


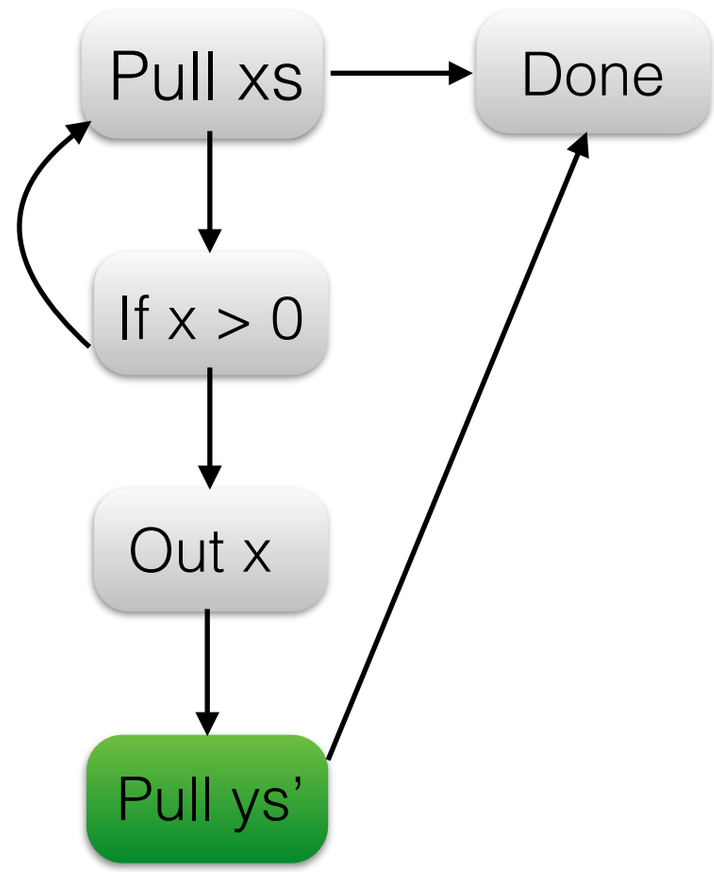


zip xs' ys'

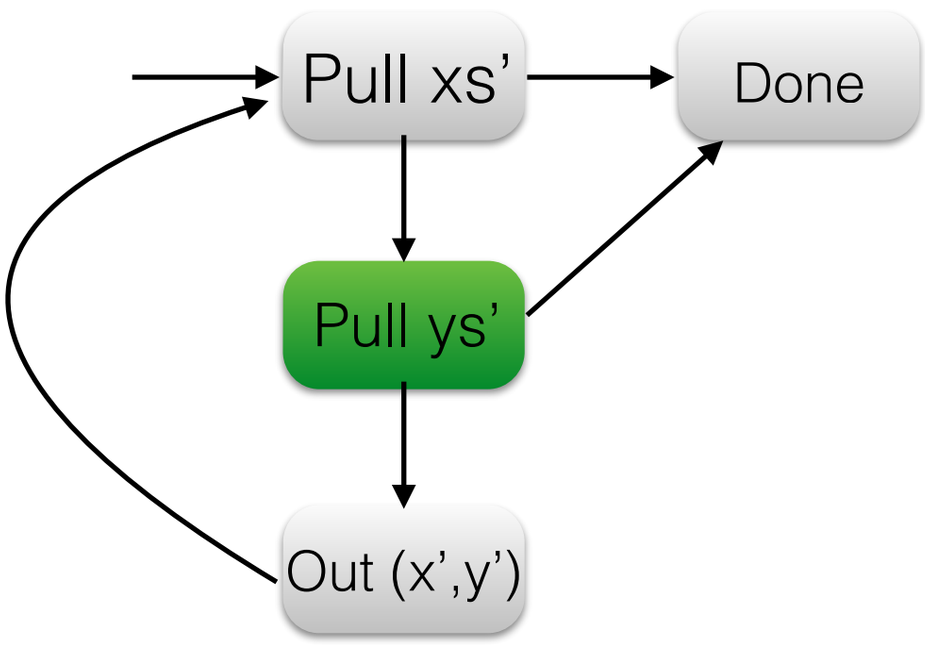


filter (>0) xs

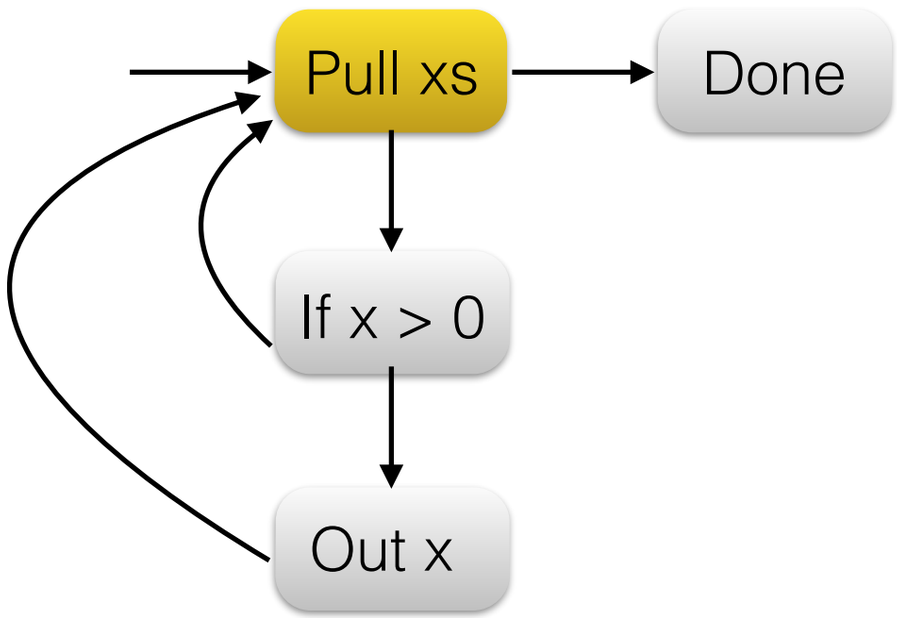




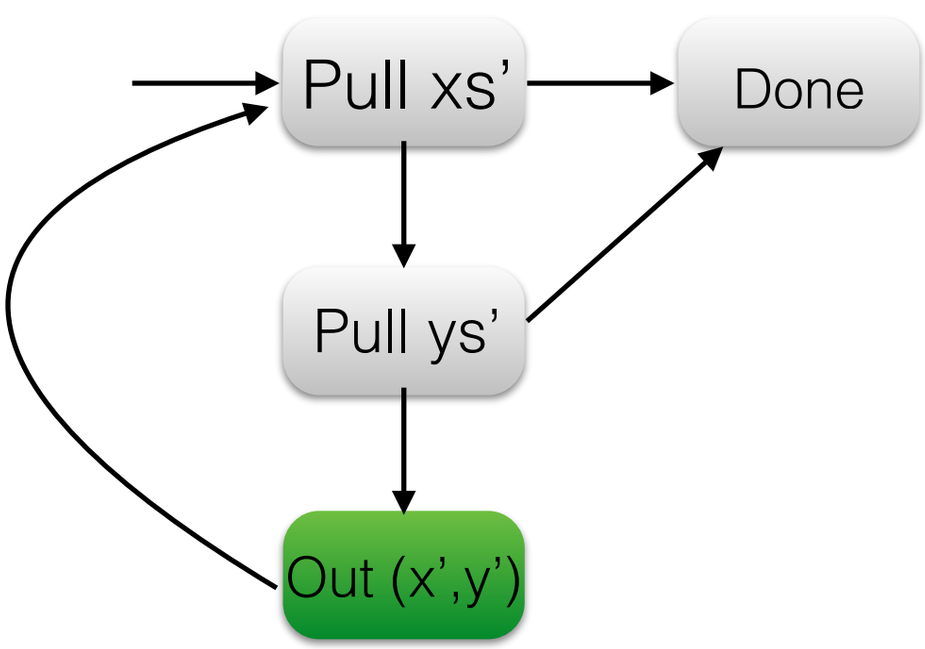
zip xs' ys'

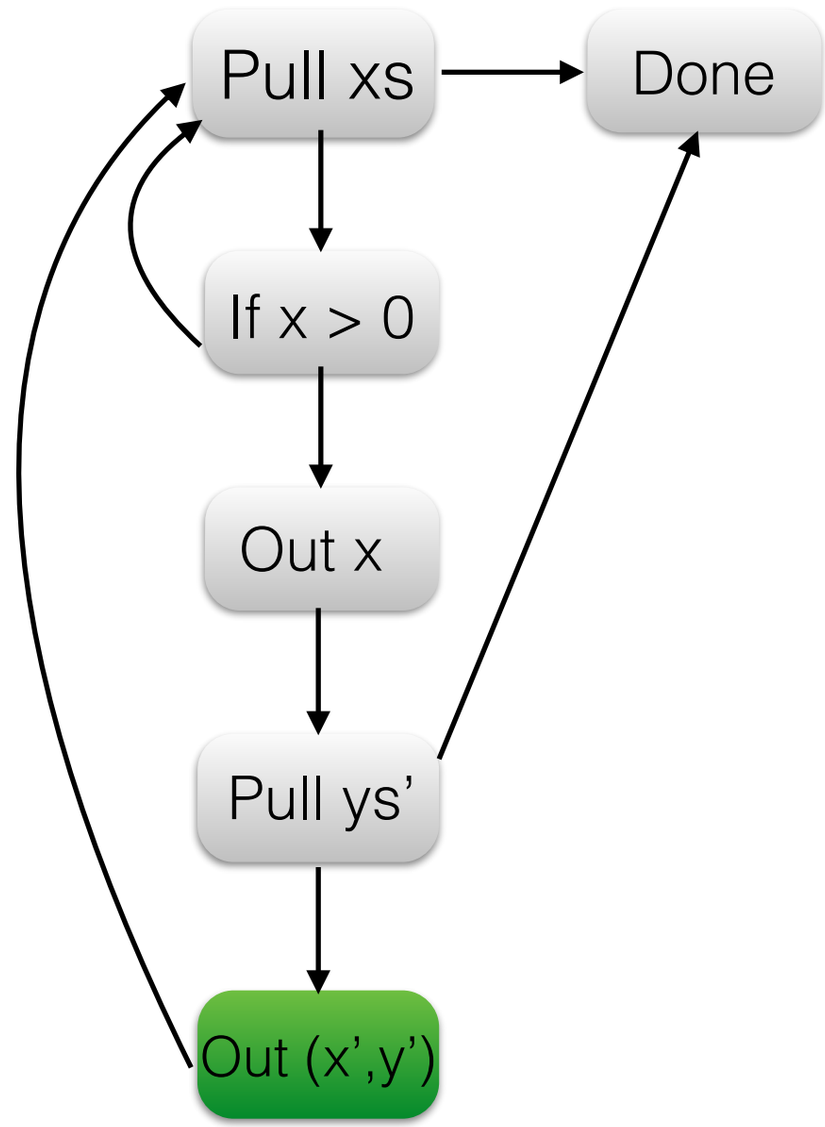


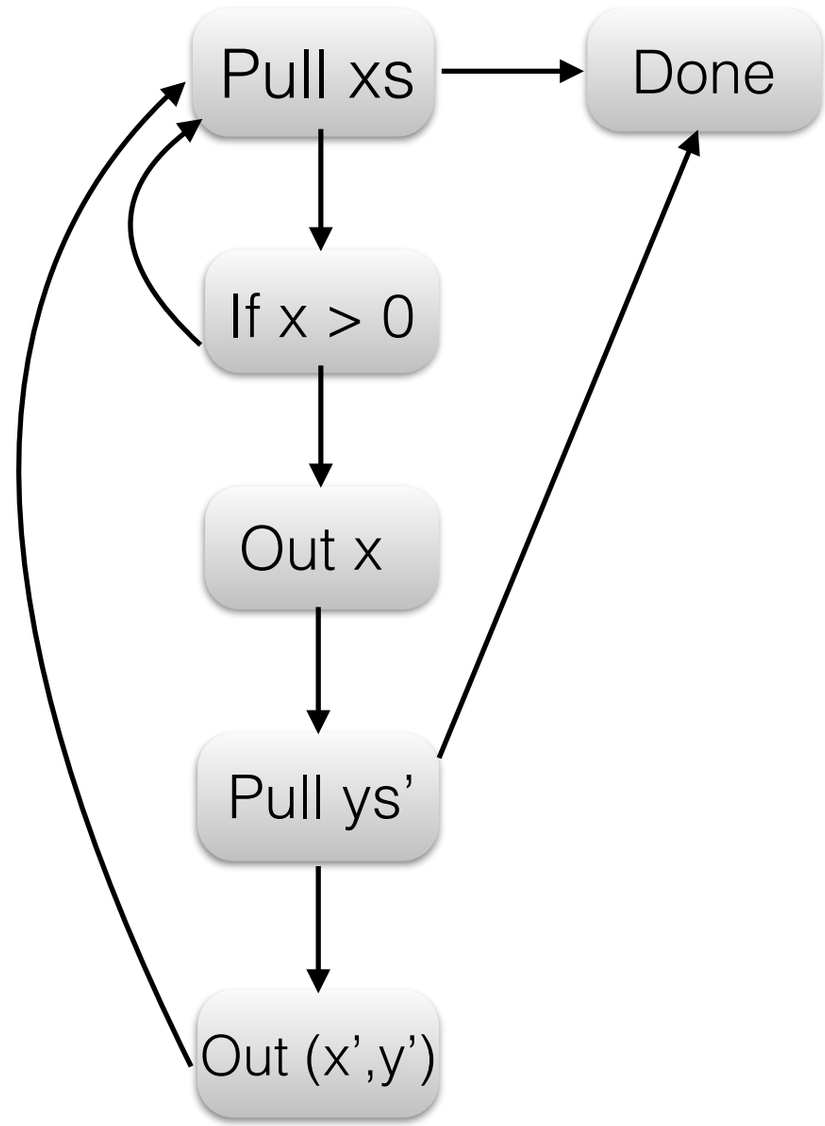
filter (>0) xs



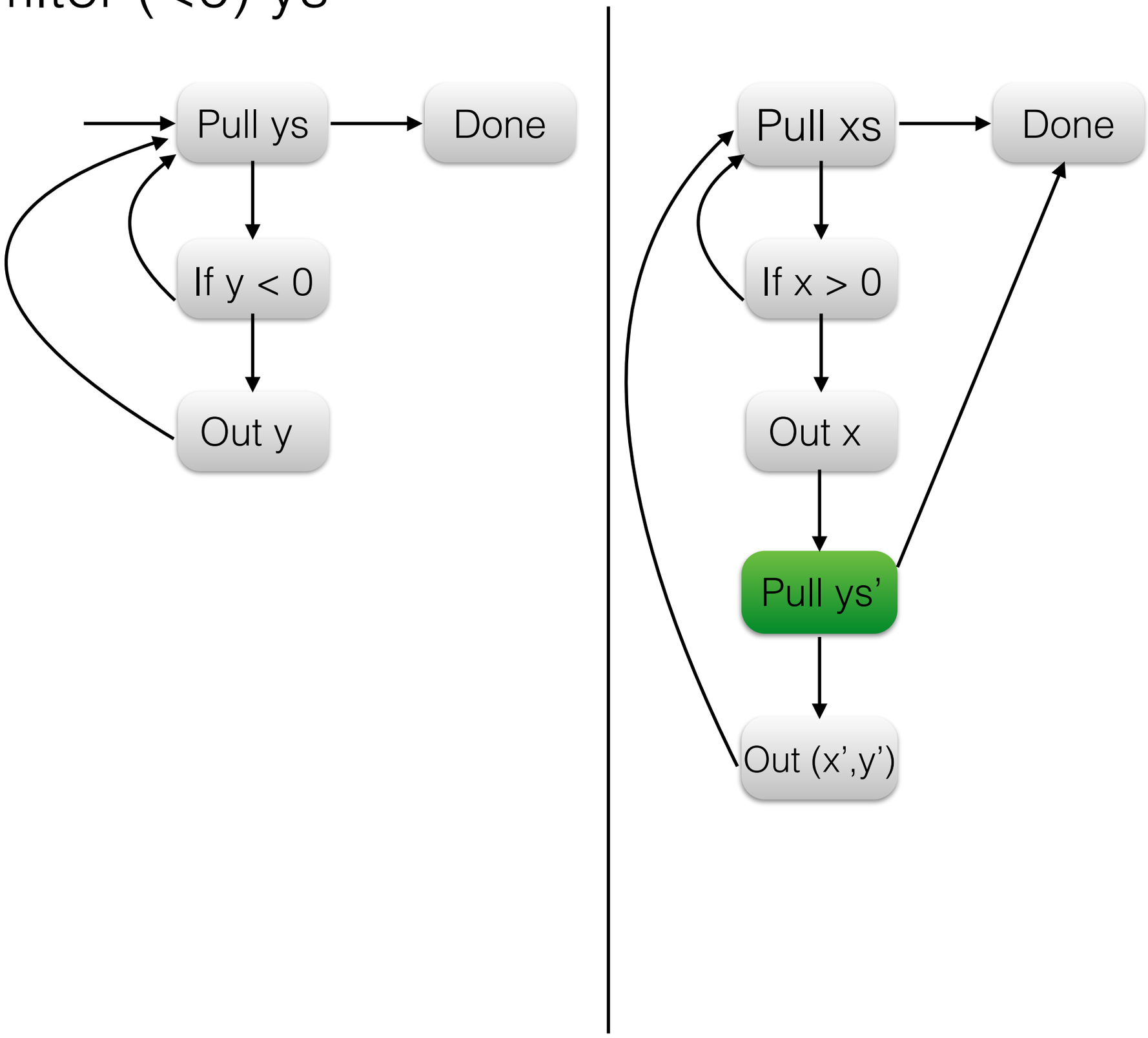
zip xs' ys'



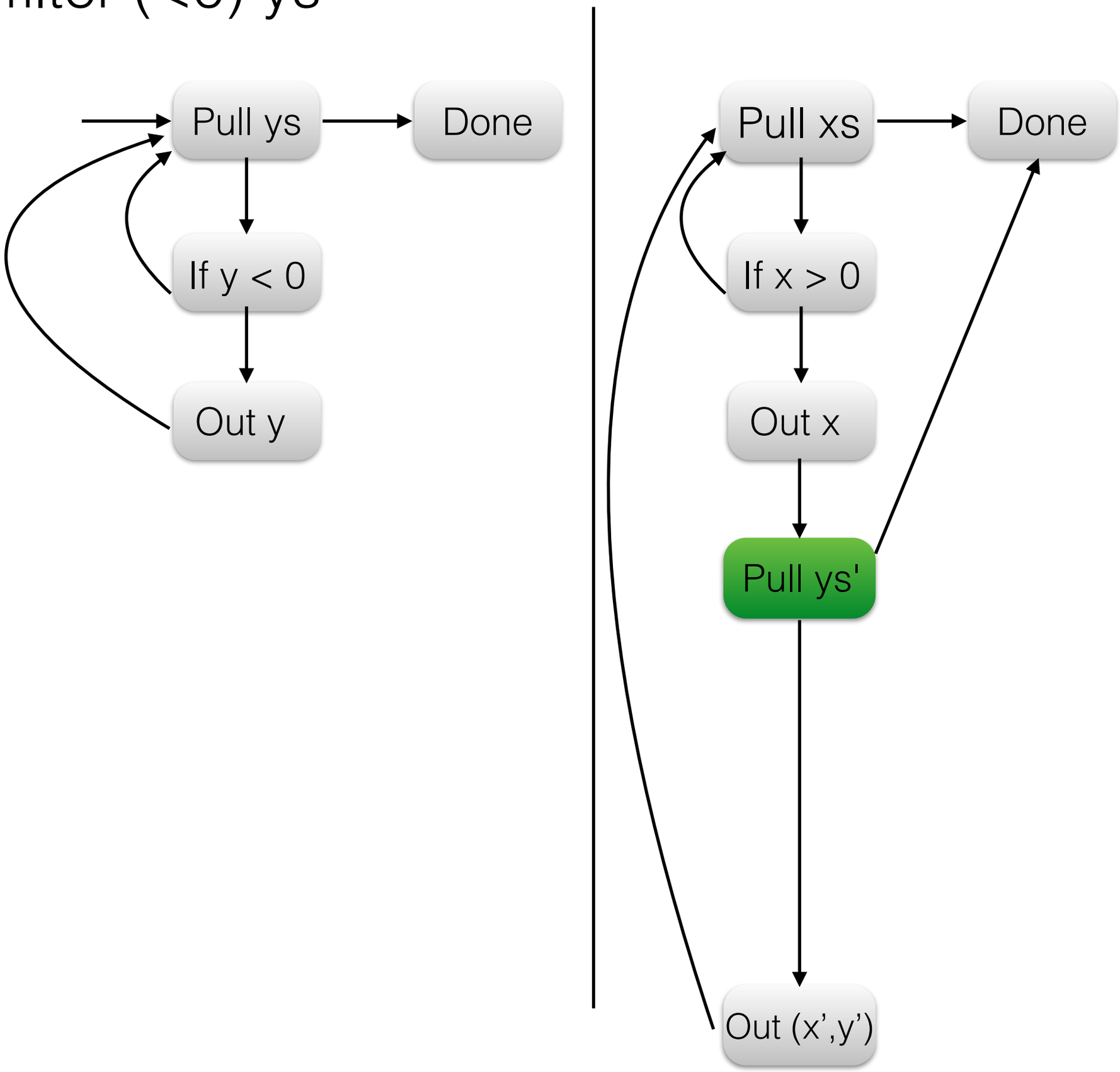



Pull xs
Done
If x > 0
Out x
Pull ys'
Out (x',y')

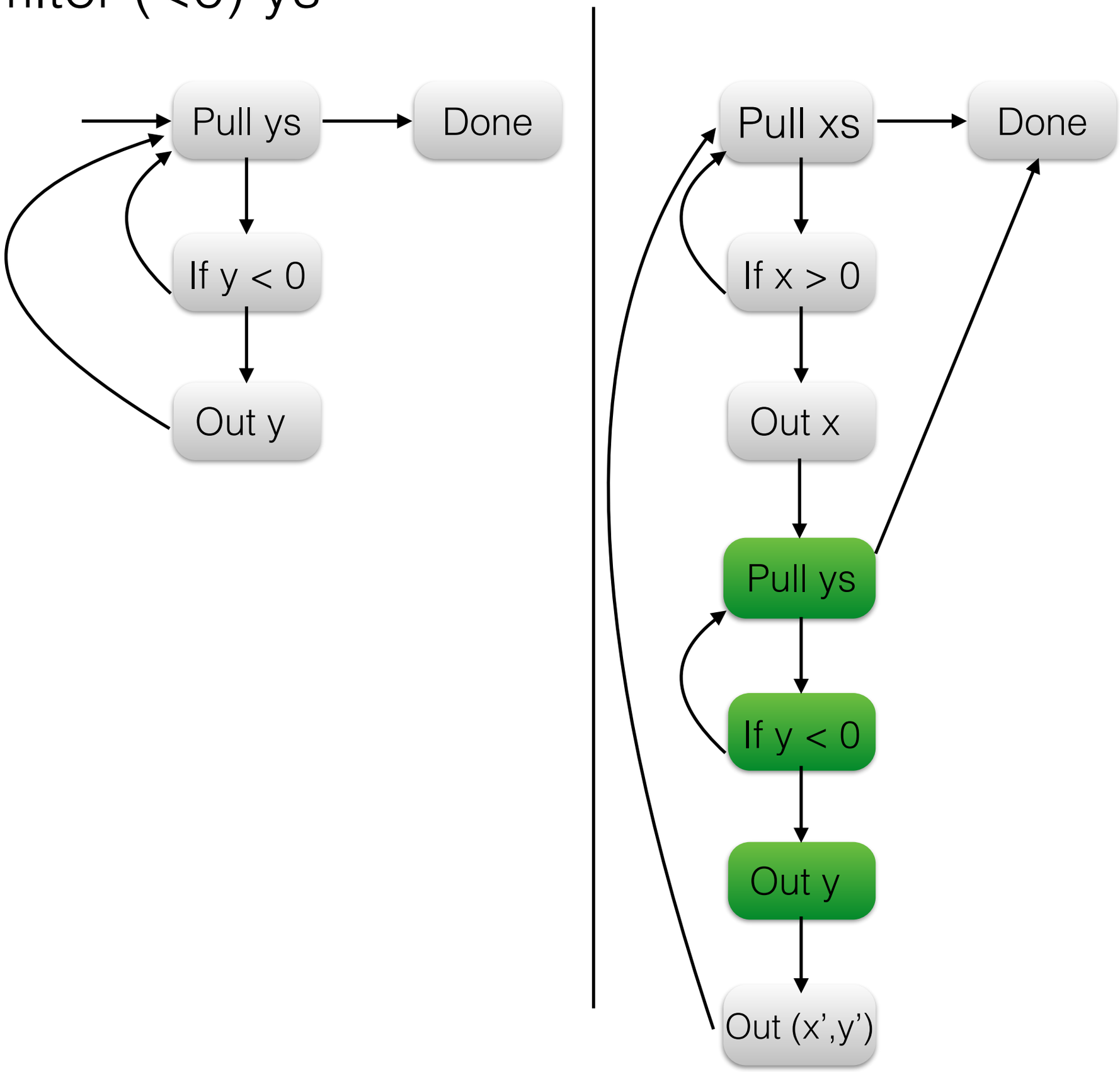
filter (<0) ys

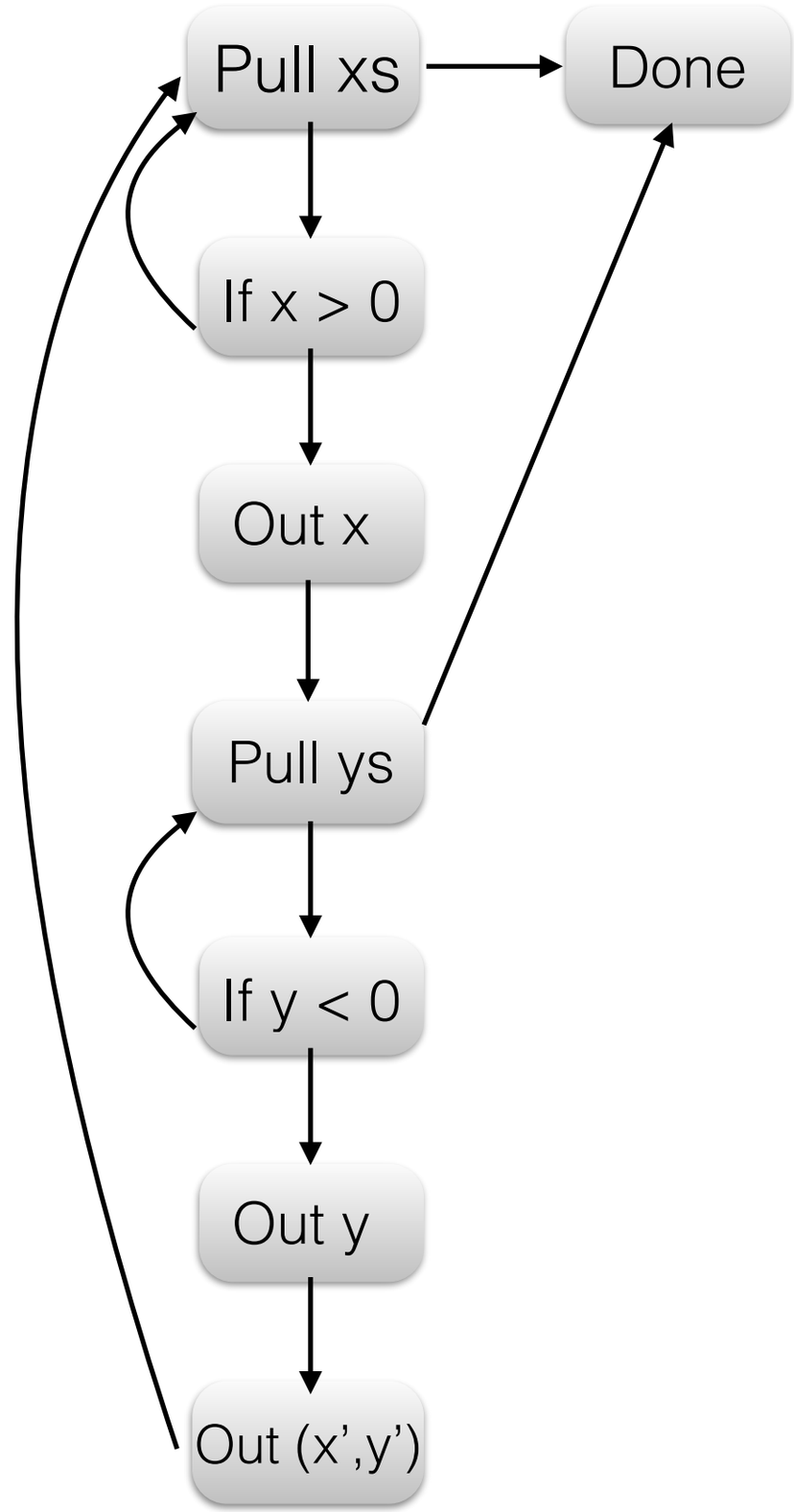


filter (<0) ys

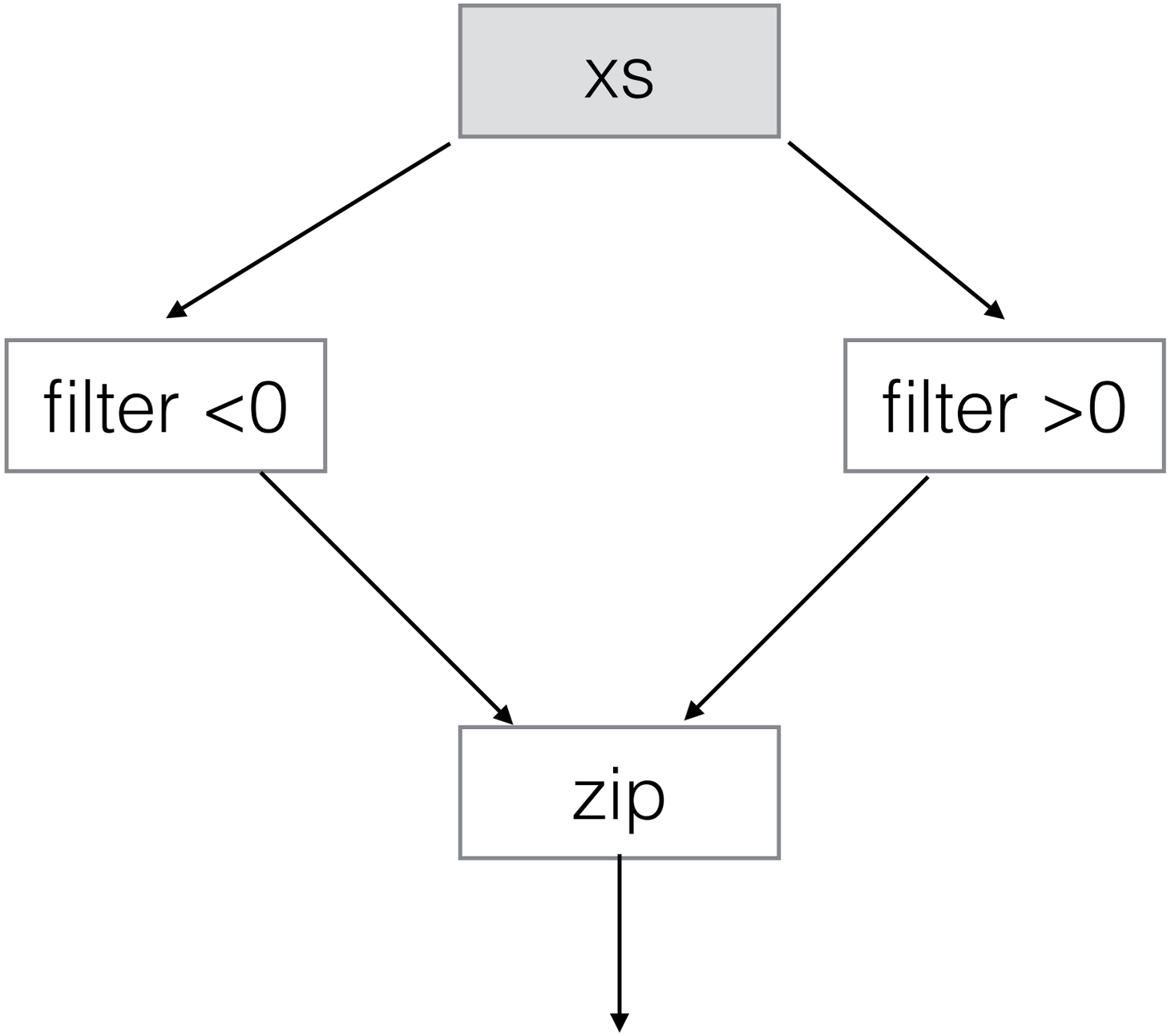


filter (<0) ys

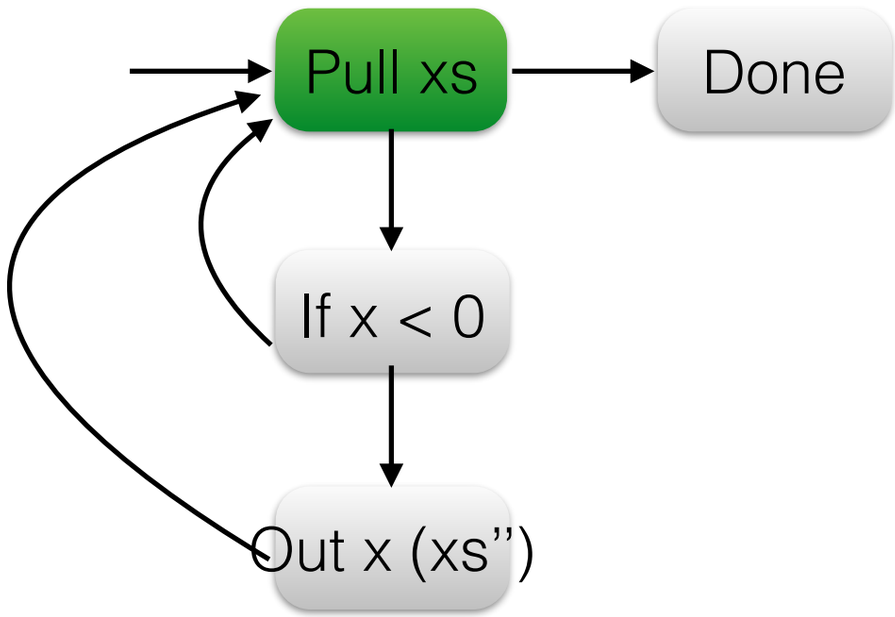

Pull ys
Done
If y < 0
Out y
Pull xs
Done
If x > 0
Out x
Pull ys
If y < 0
Out y
Out (x',y')


Pull xs
Done
If x > 0
Out x
Pull ys
If y < 0
Out y
Out (x',y')

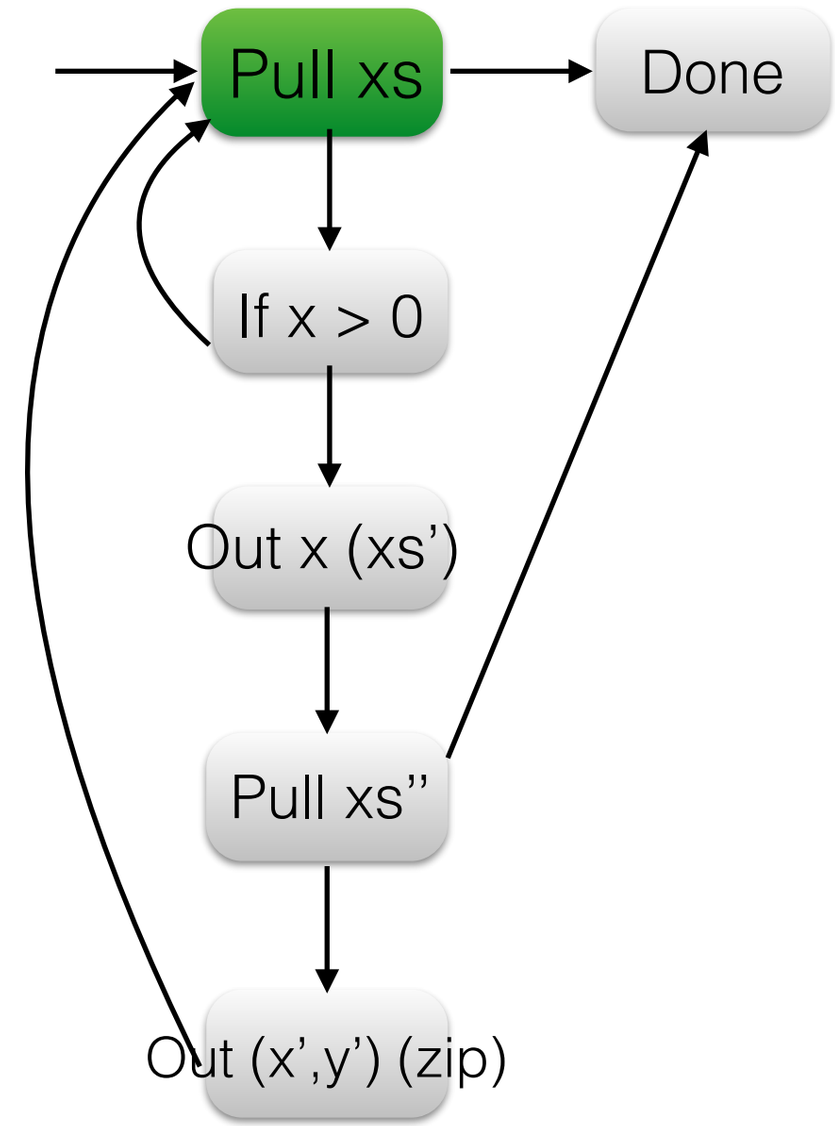
bad again



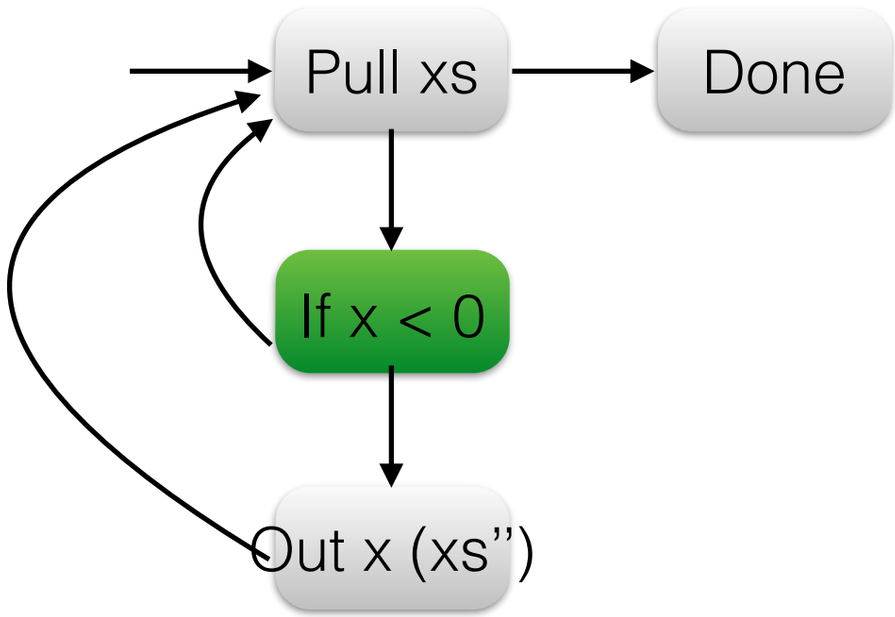
filter (<0) xs



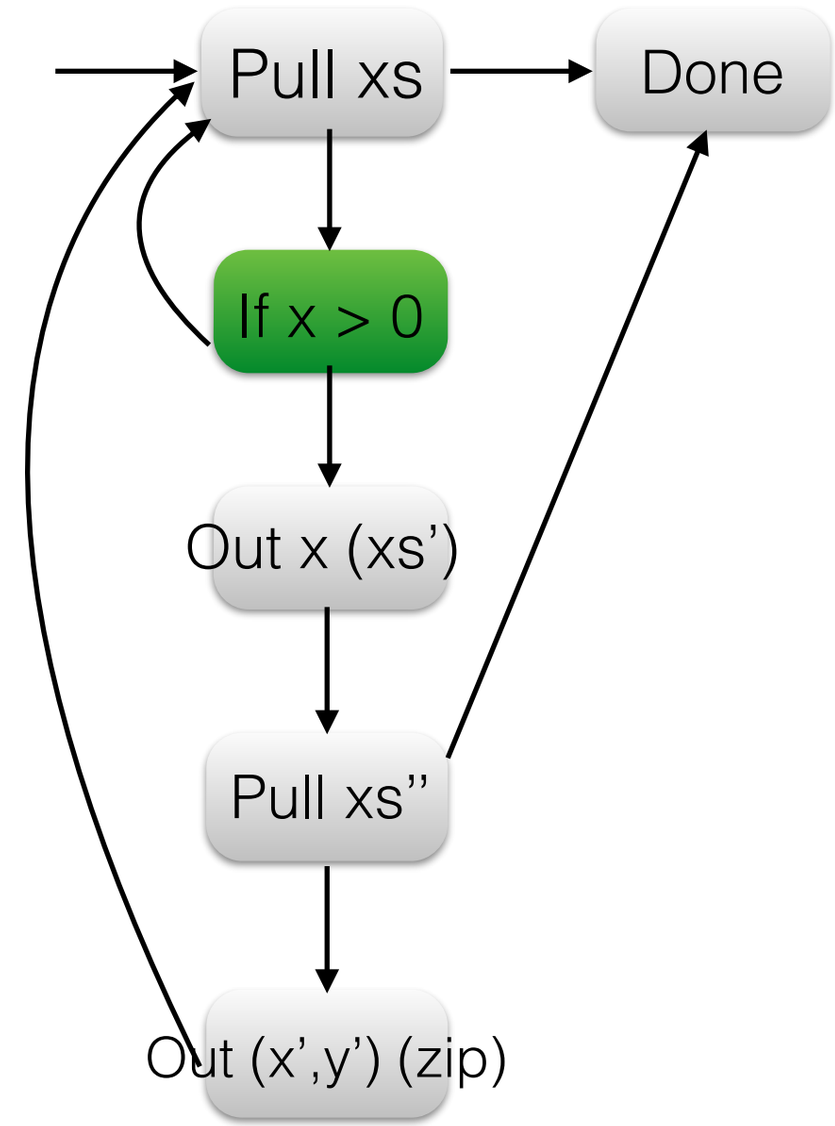
zip (filter (>0) xs) xs''



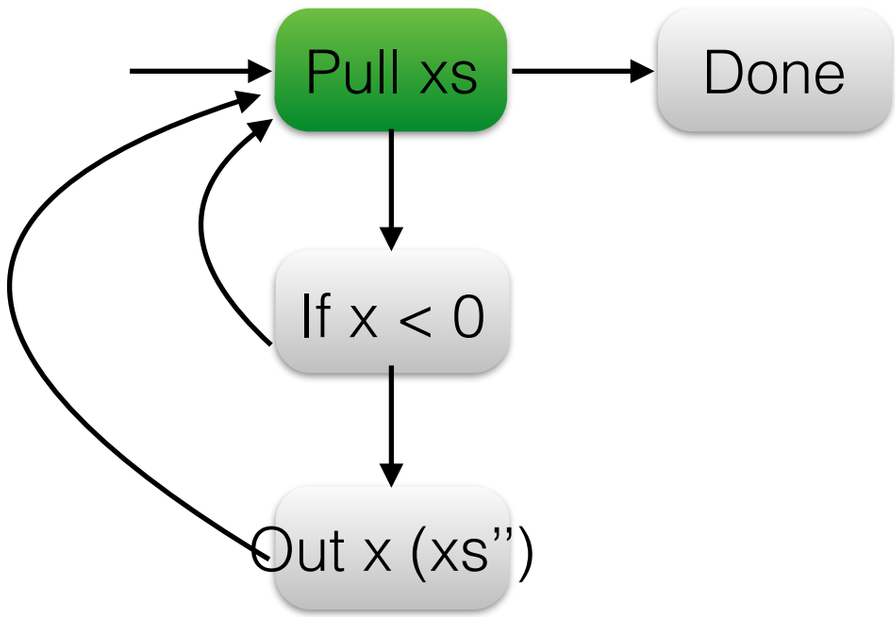
filter (<0) xs



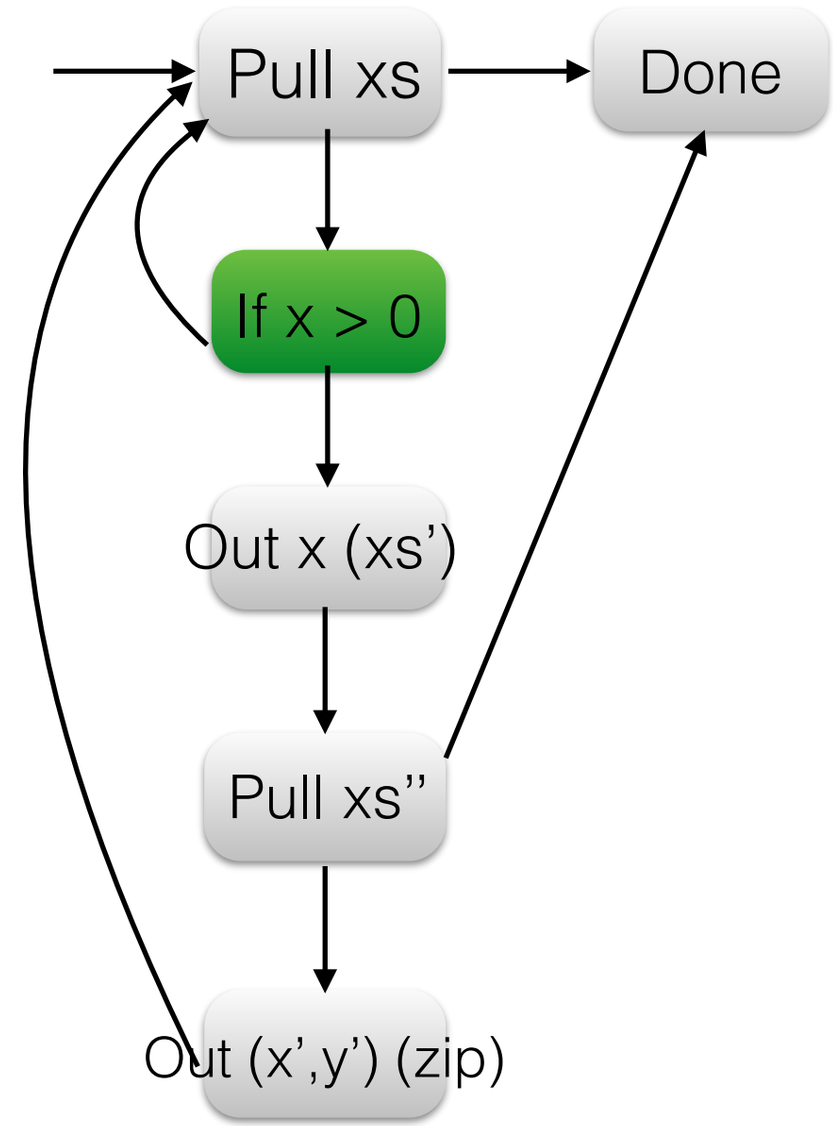
zip (filter (>0) xs) xs''



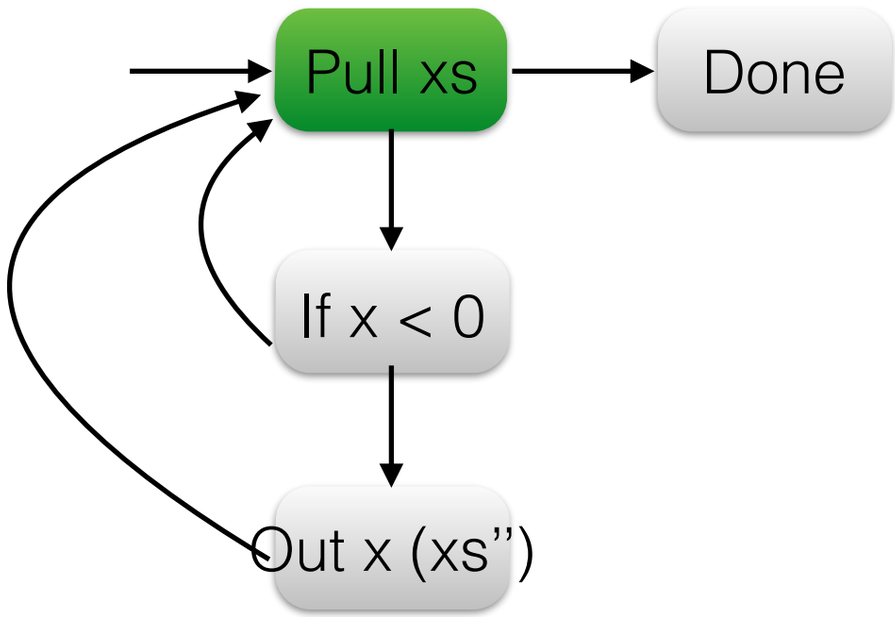
filter (<0) xs



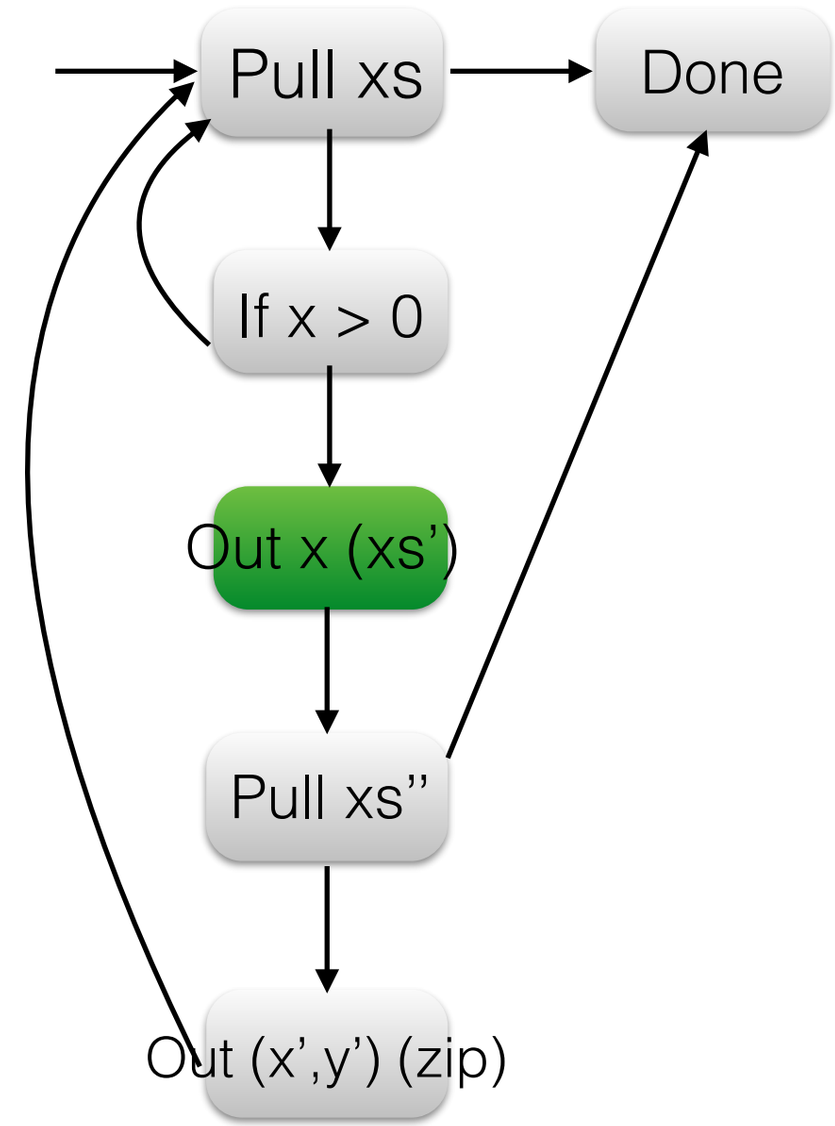
zip (filter (>0) xs) xs''



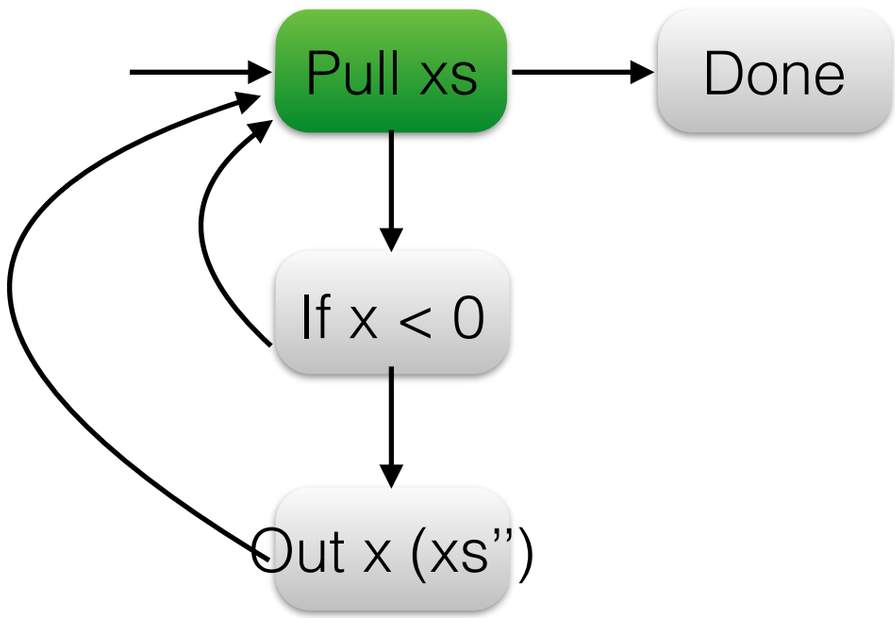
filter (<0) xs



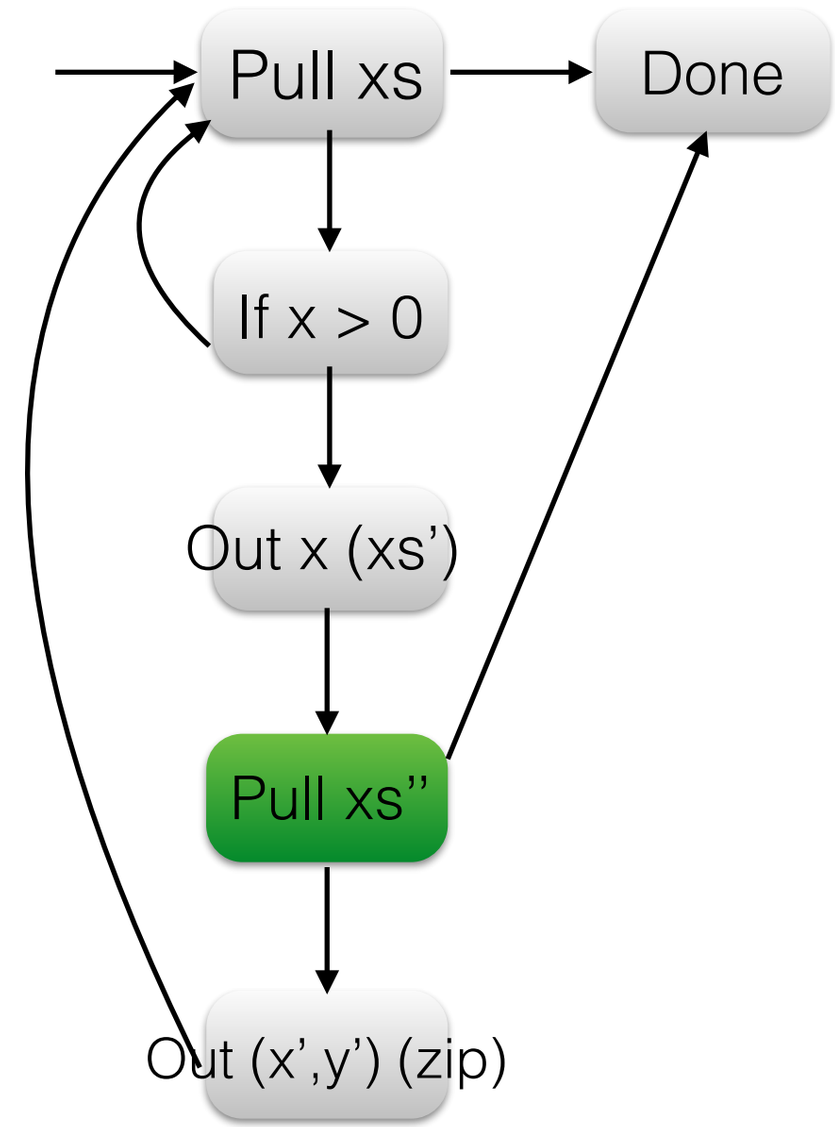
zip (filter (>0) xs) xs''



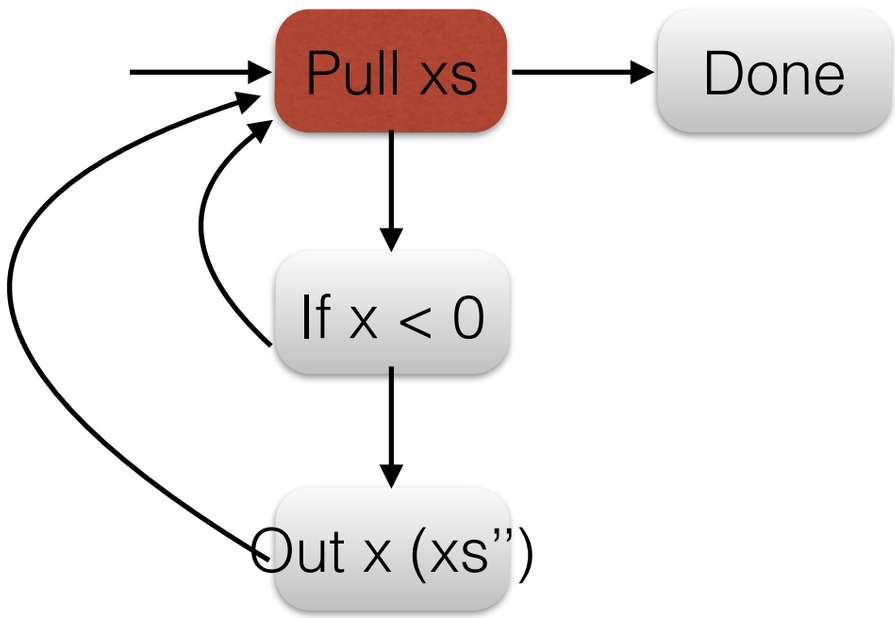
filter (<0) xs



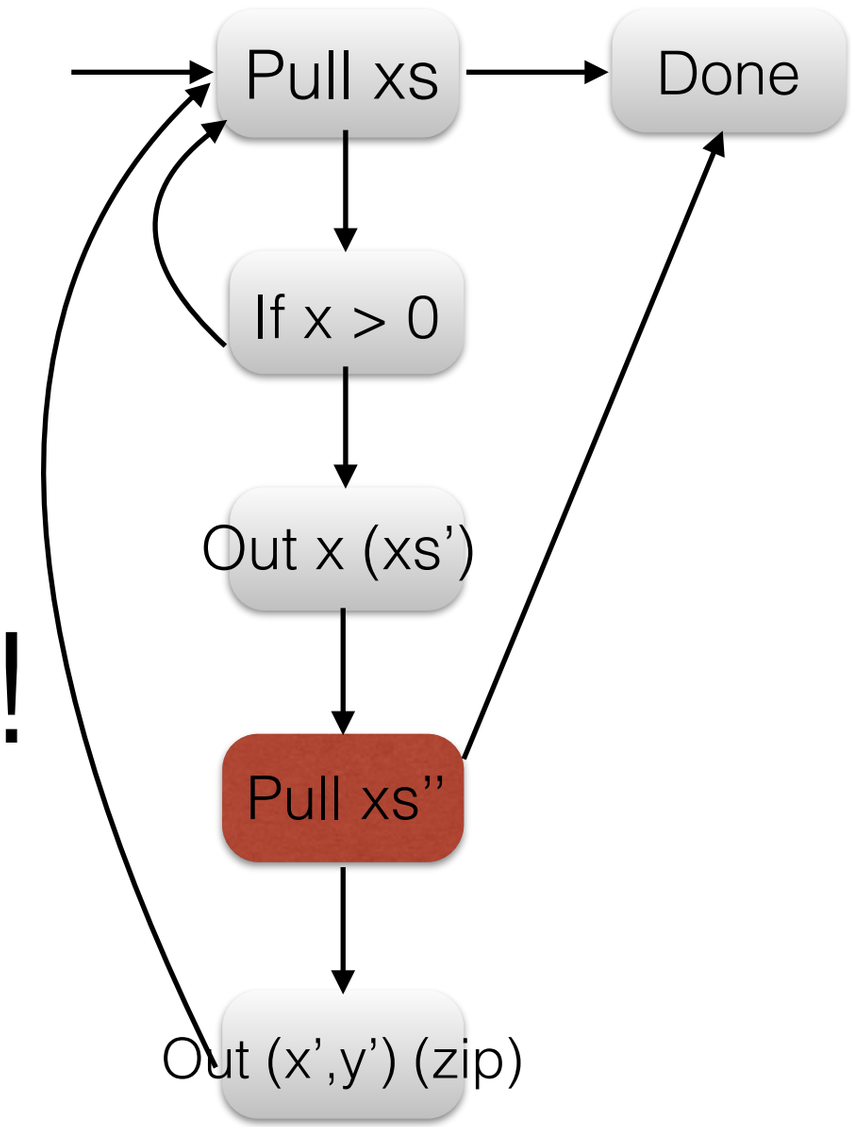
zip (filter (>0) xs) xs''



filter (<0) xs



zip (filter (>0) xs) xs''



Deadlock!