

Machine fusion

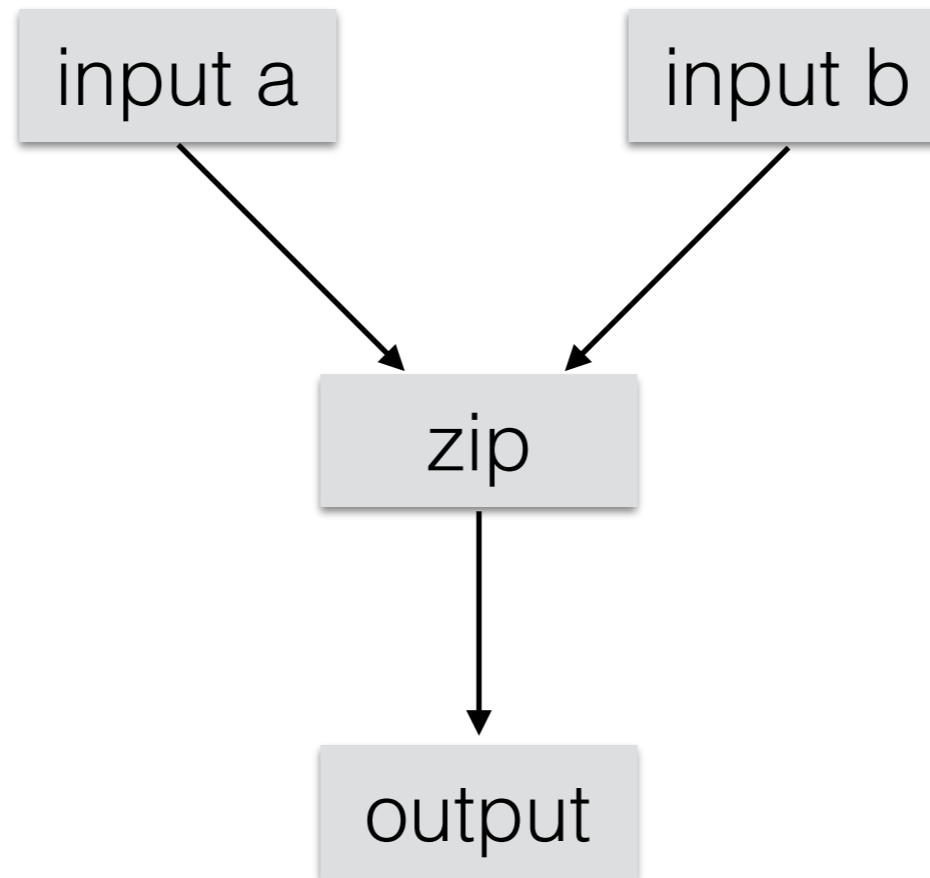
Amos Robinson, UNSW

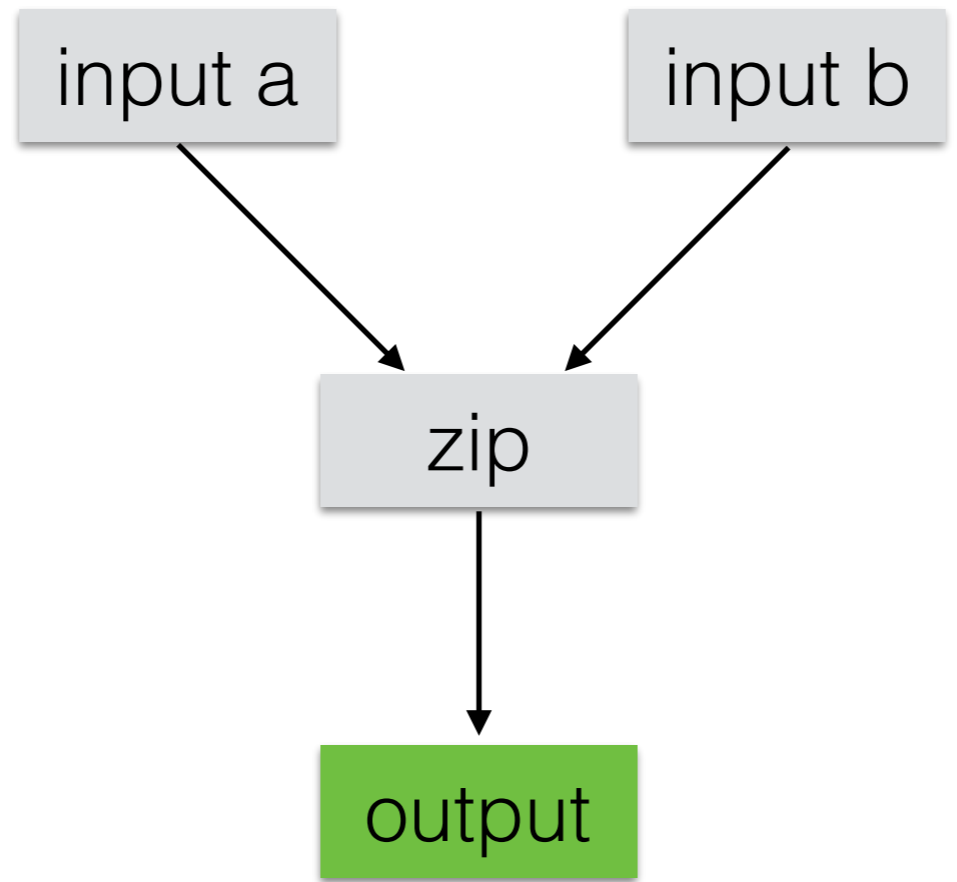
Ben Lippmeier, Digital Asset

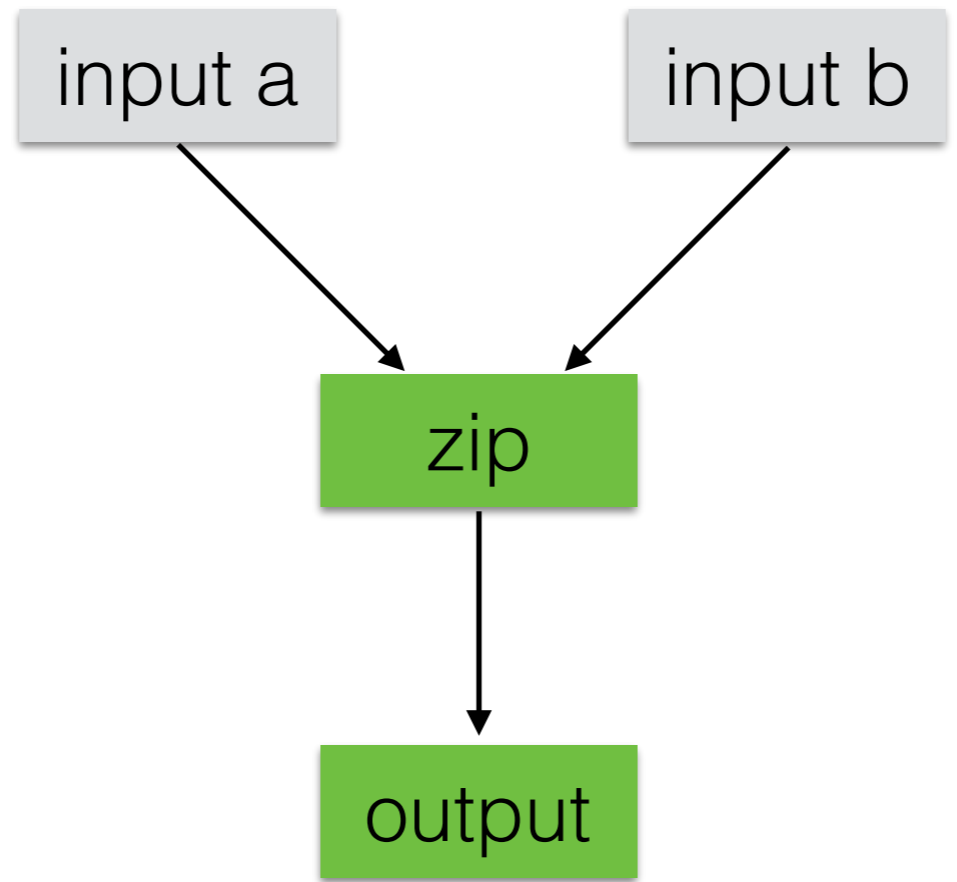
Pull streams

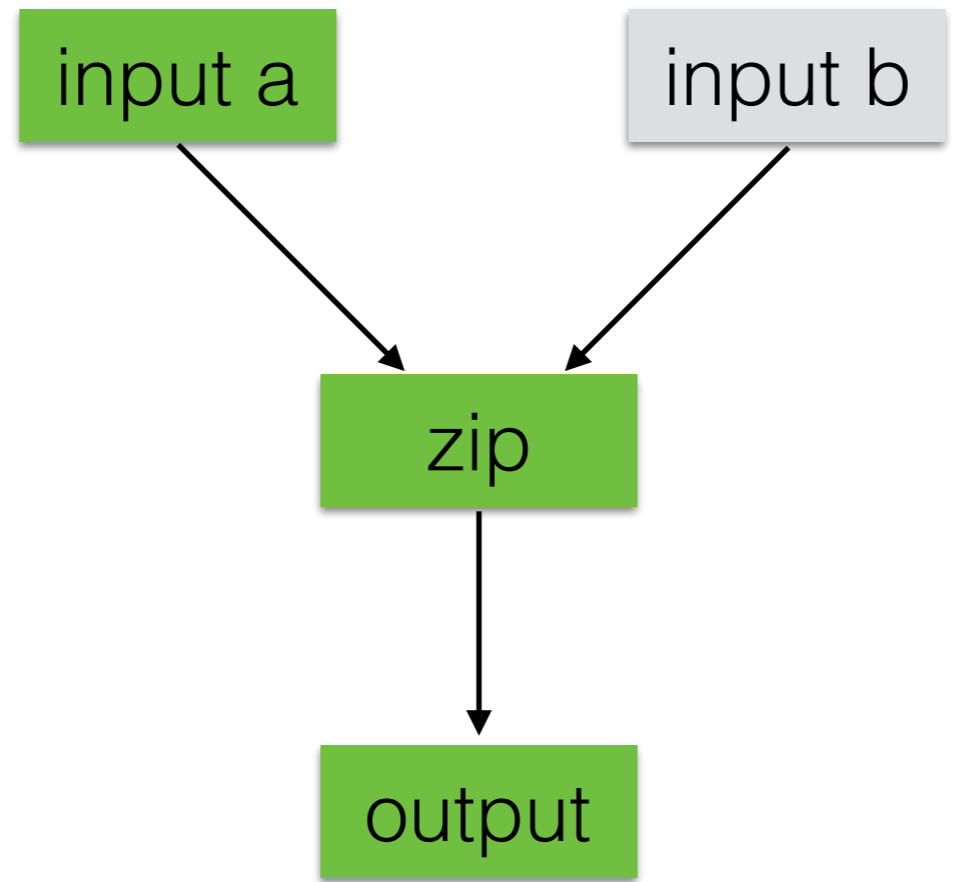
```
type Pull a = IO (Maybe a)
```

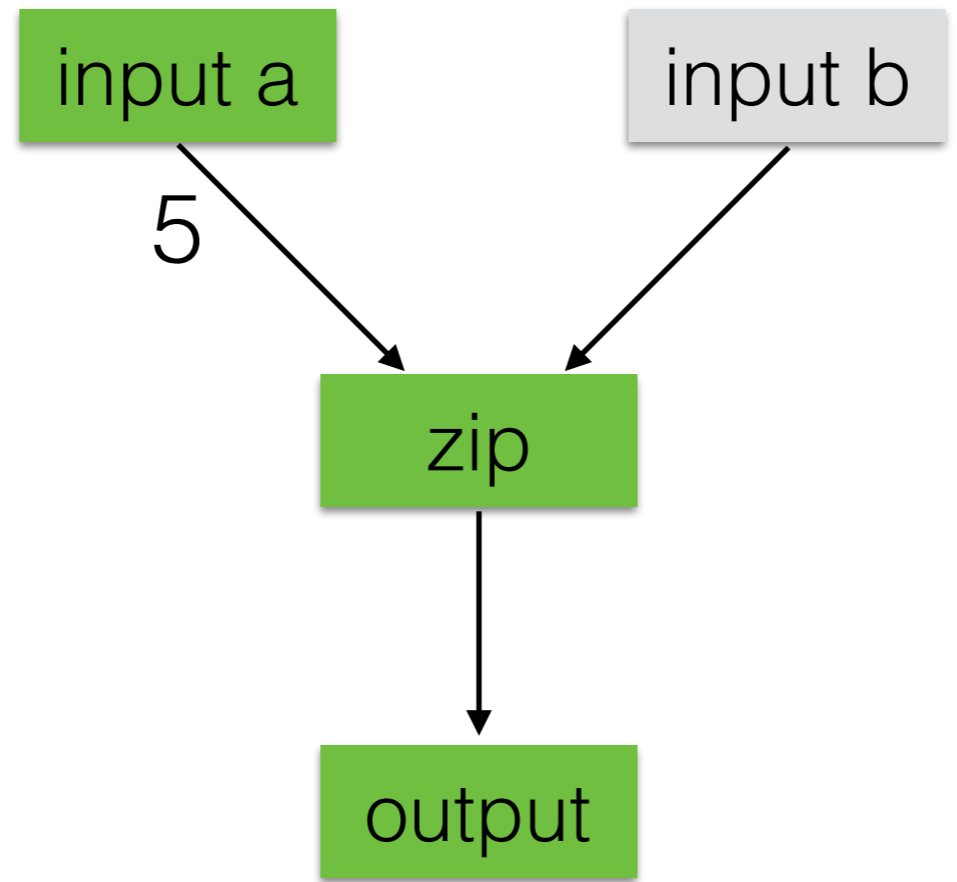
Zip

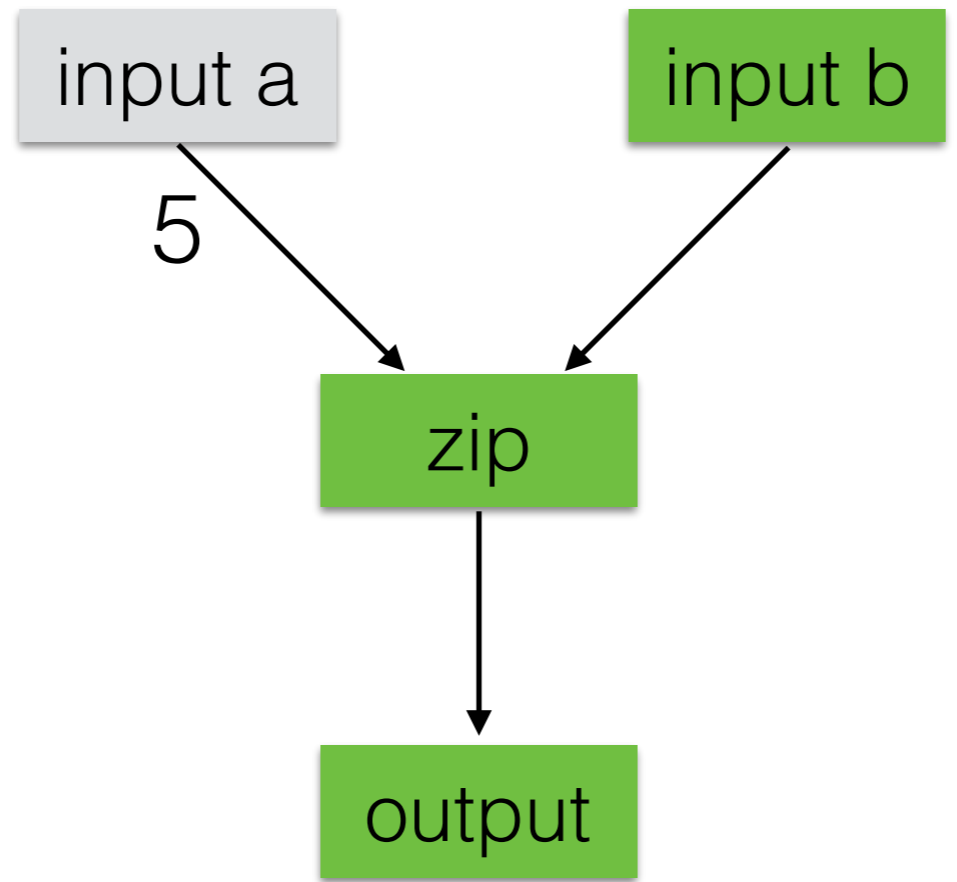


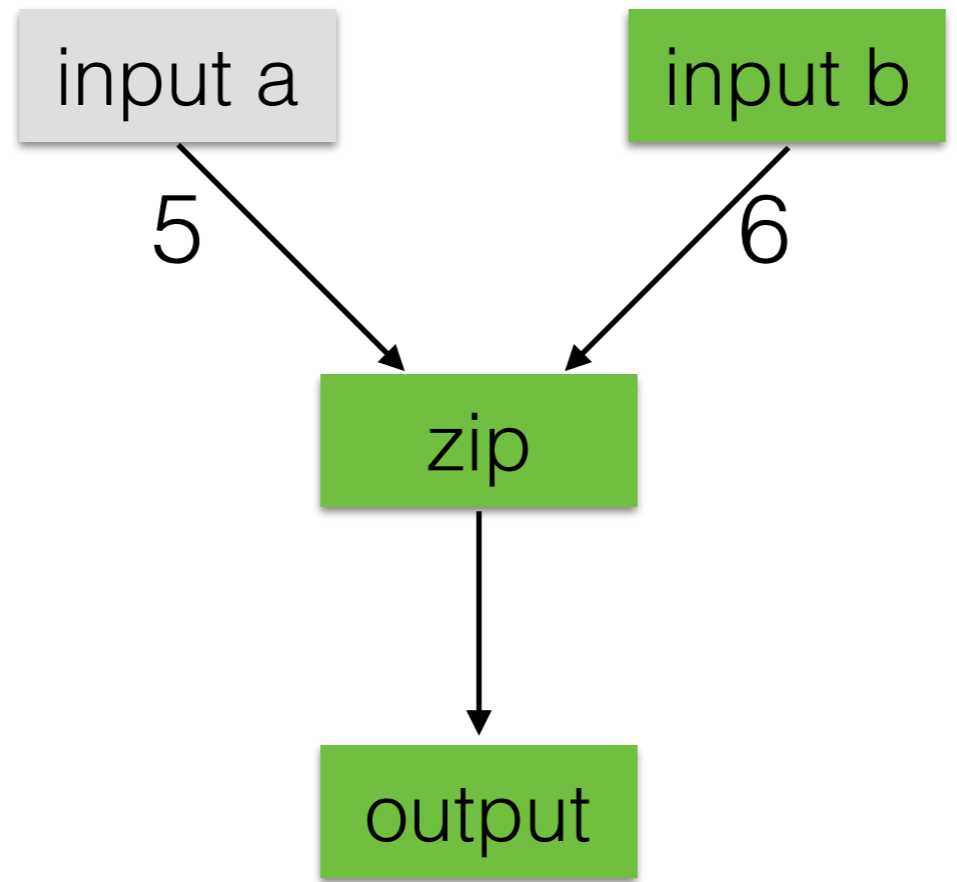


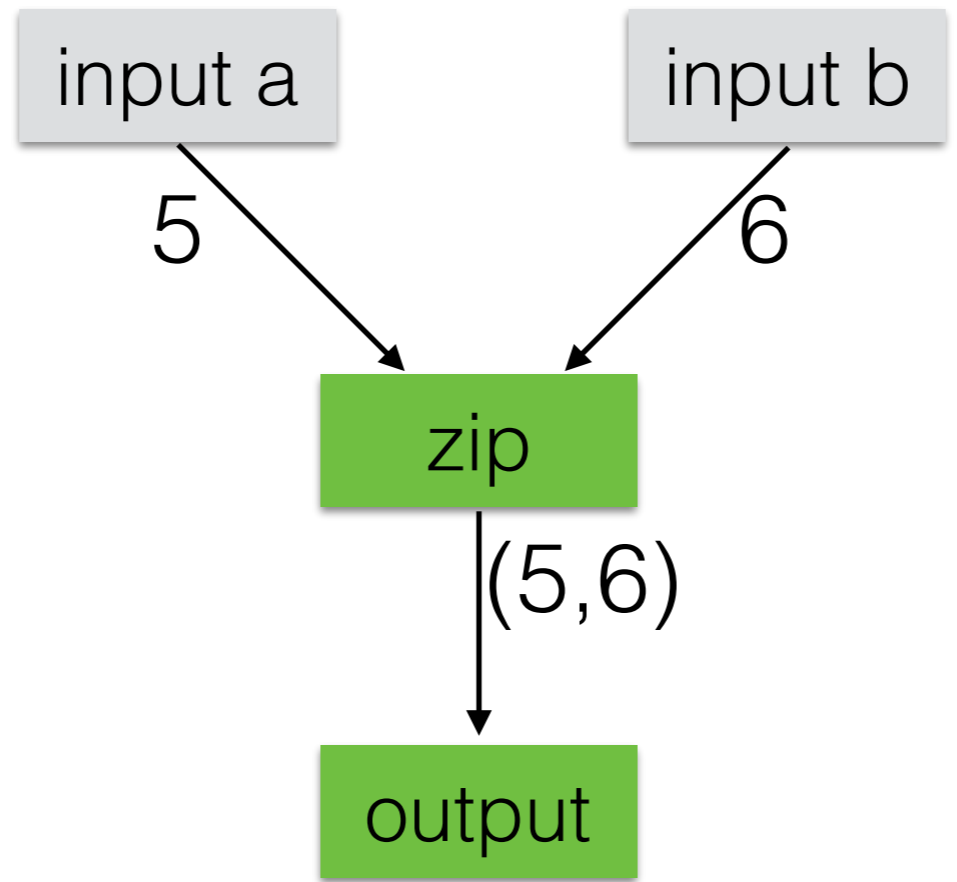


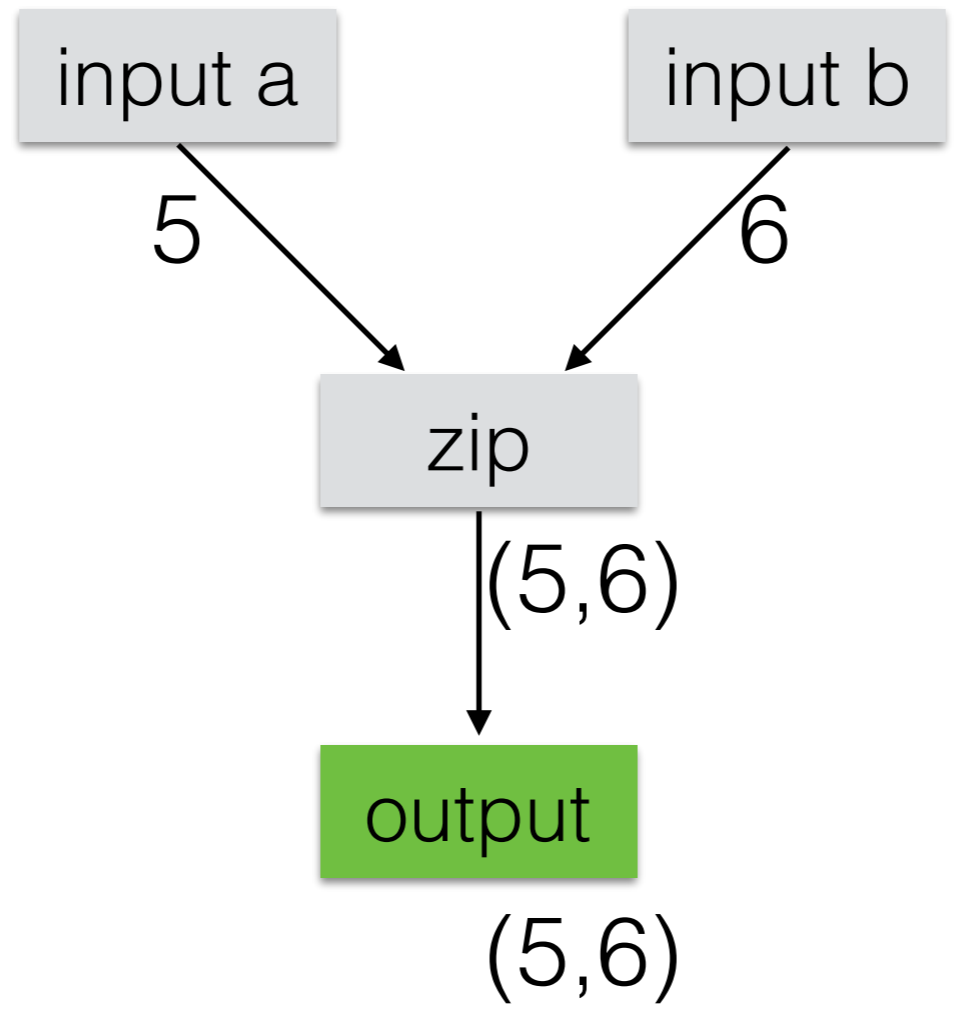




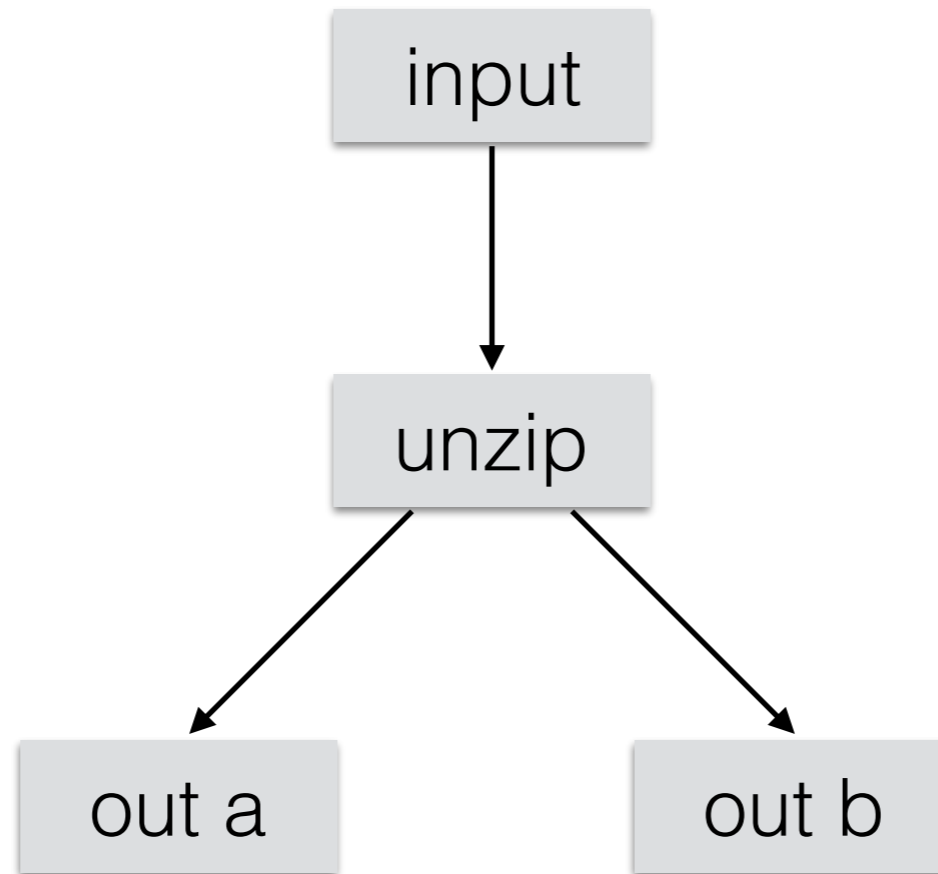


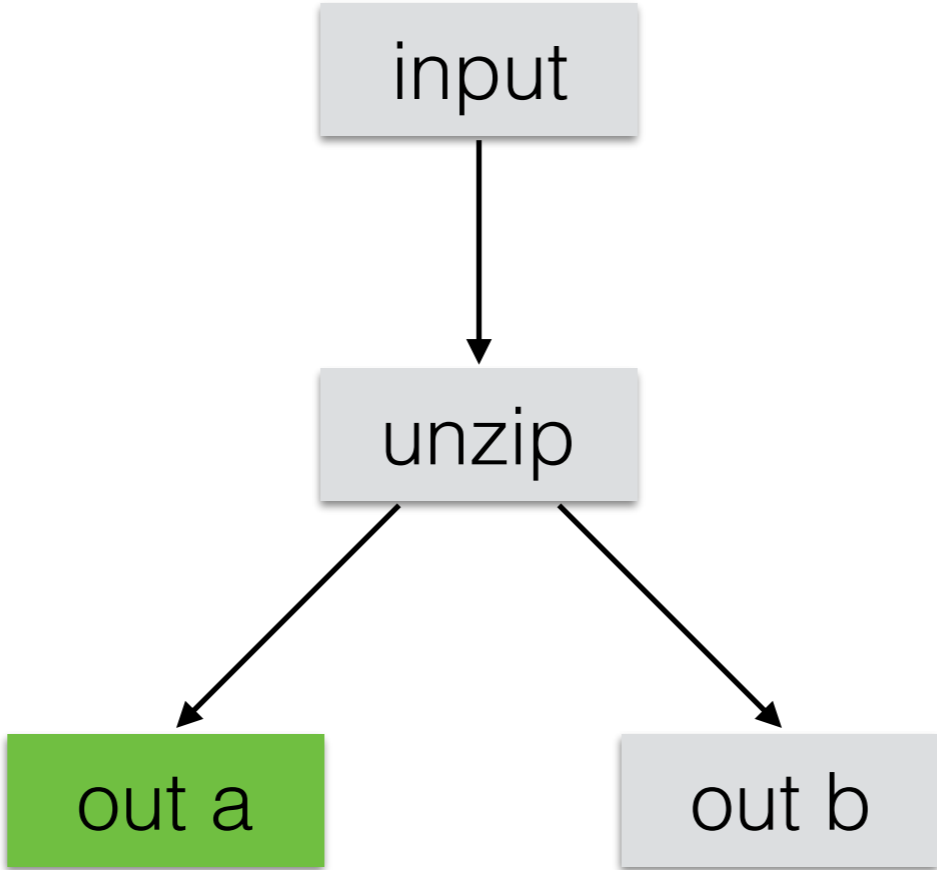


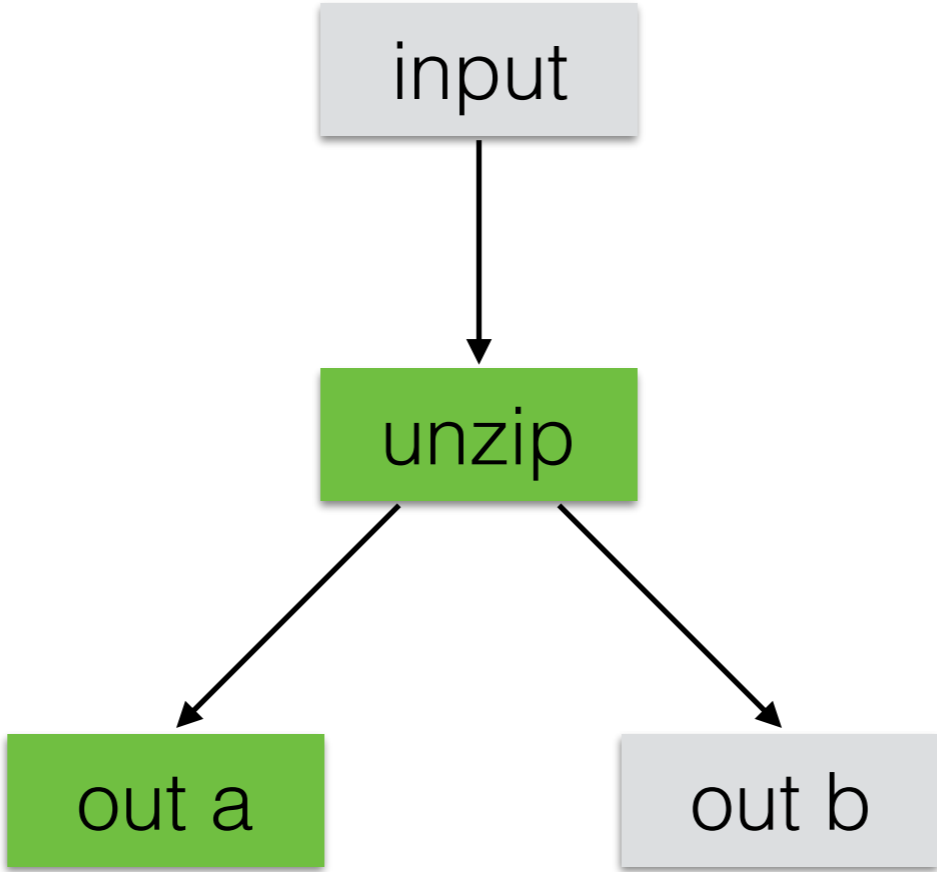


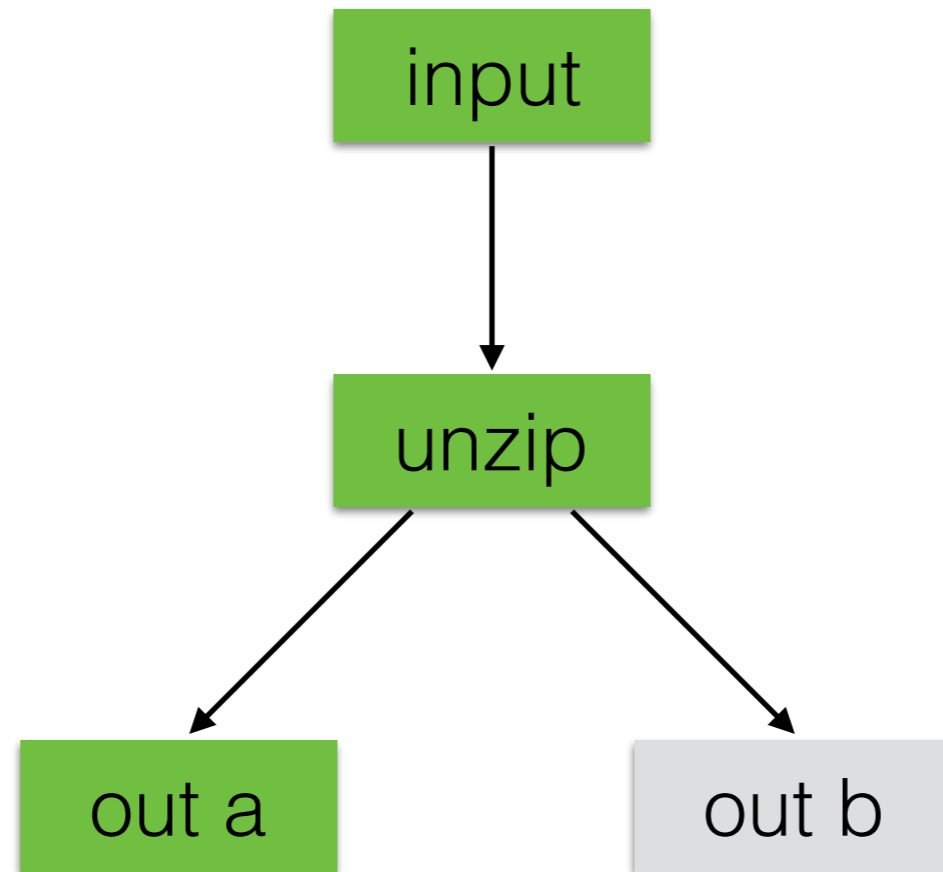


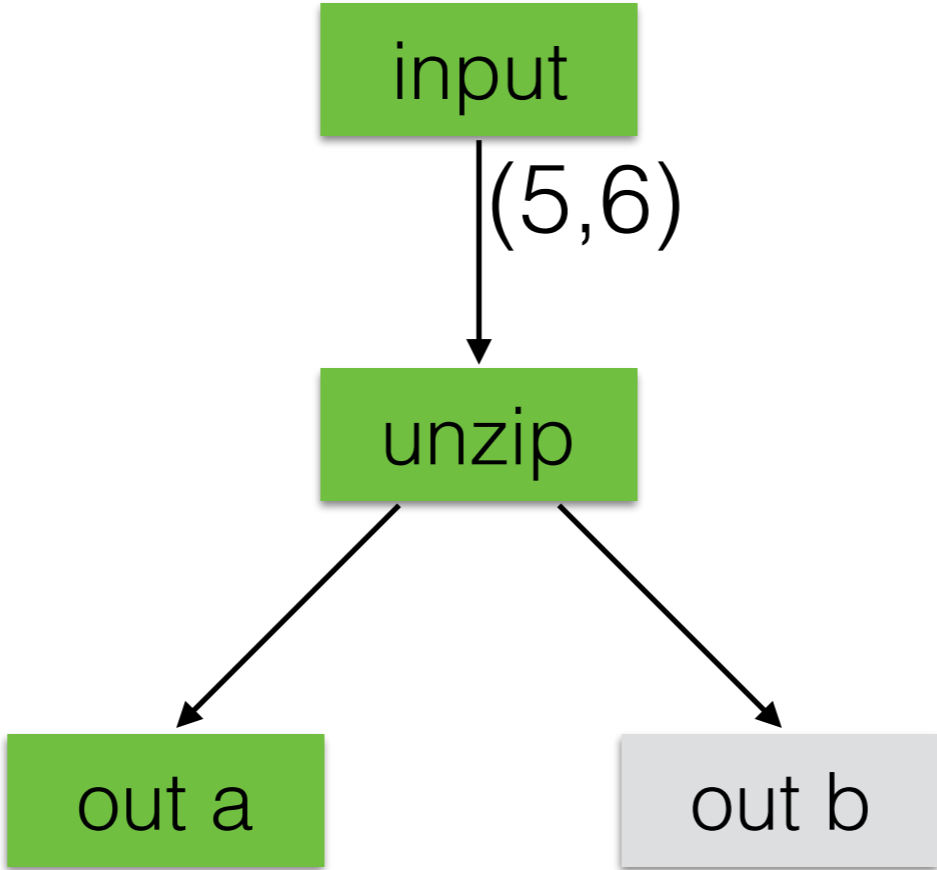
Unzip

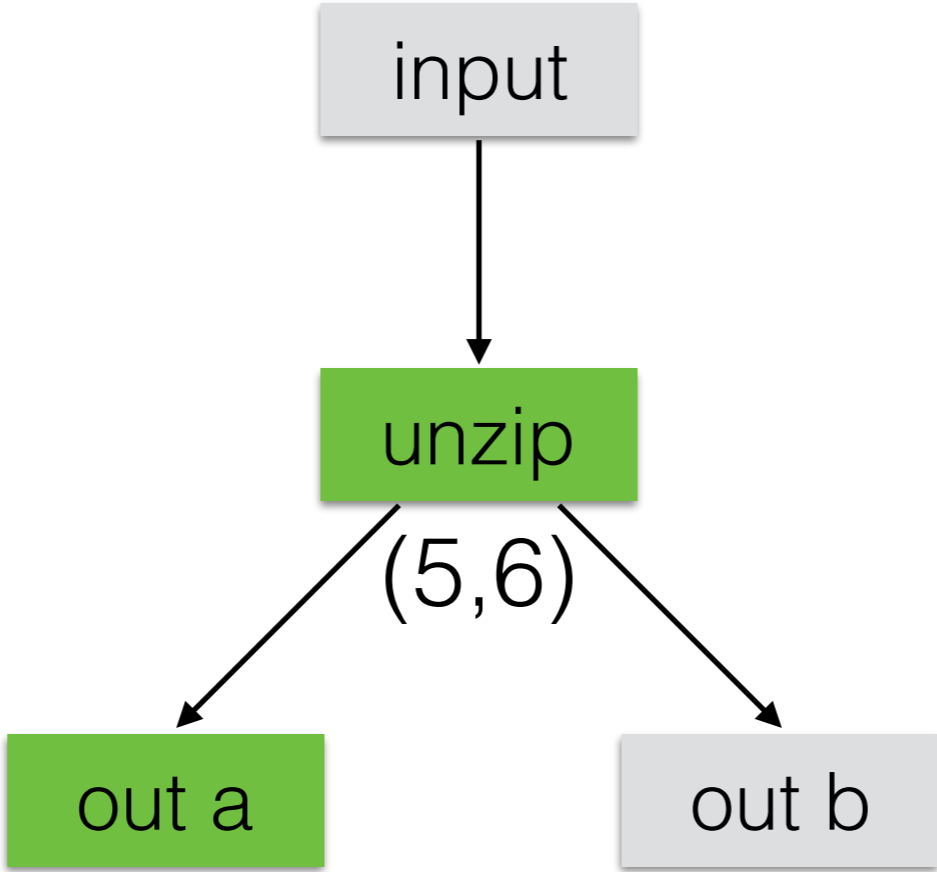


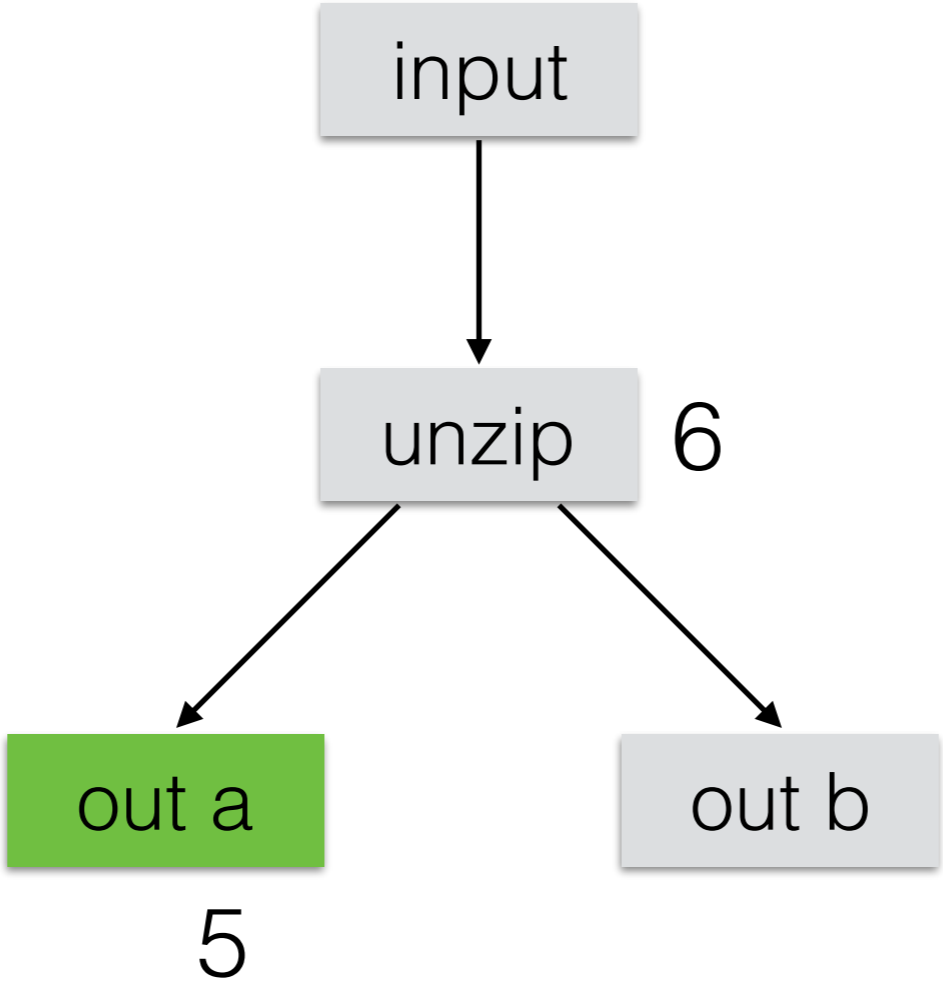


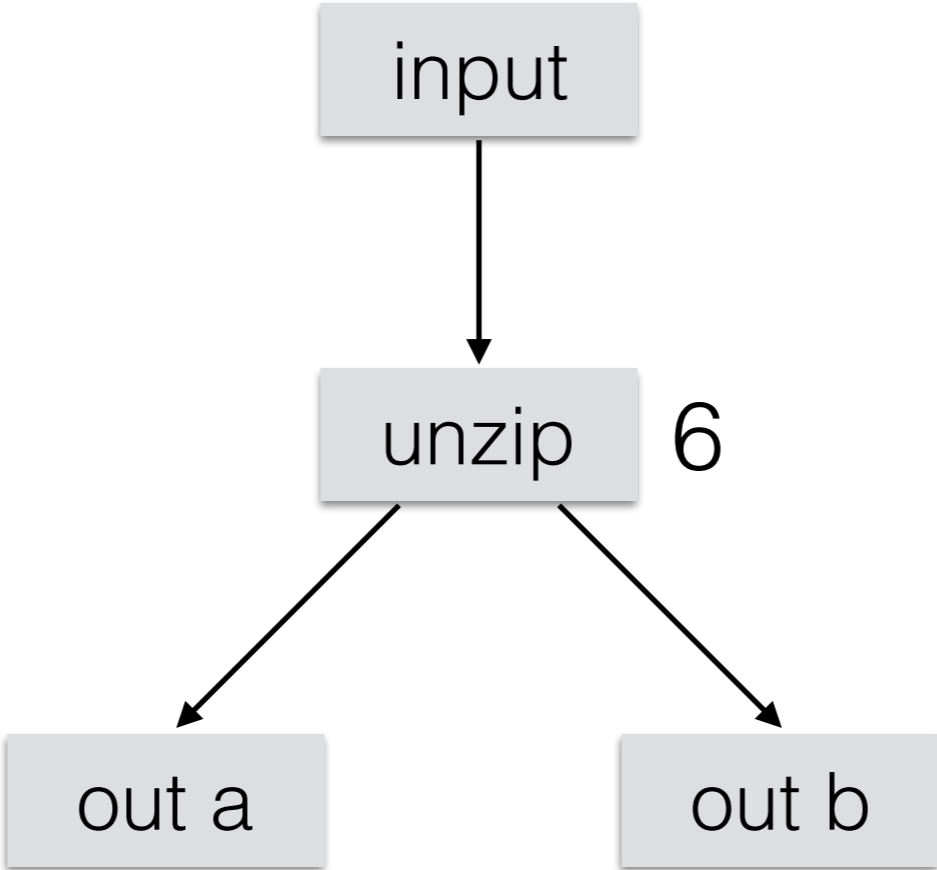


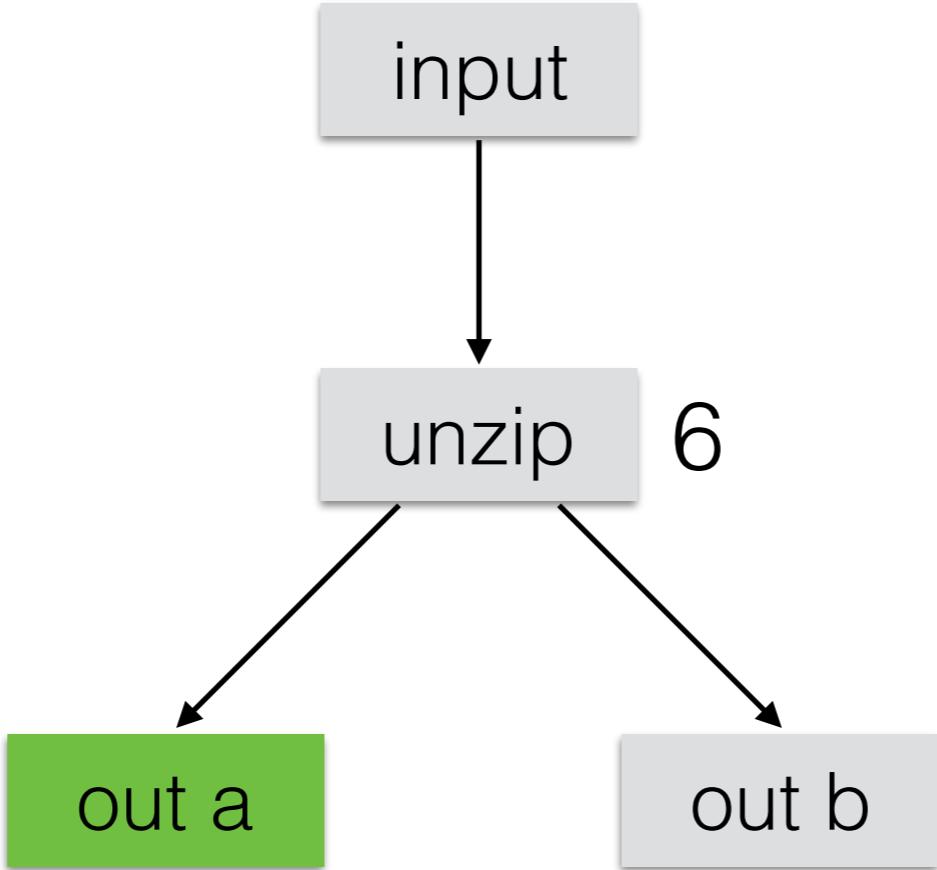


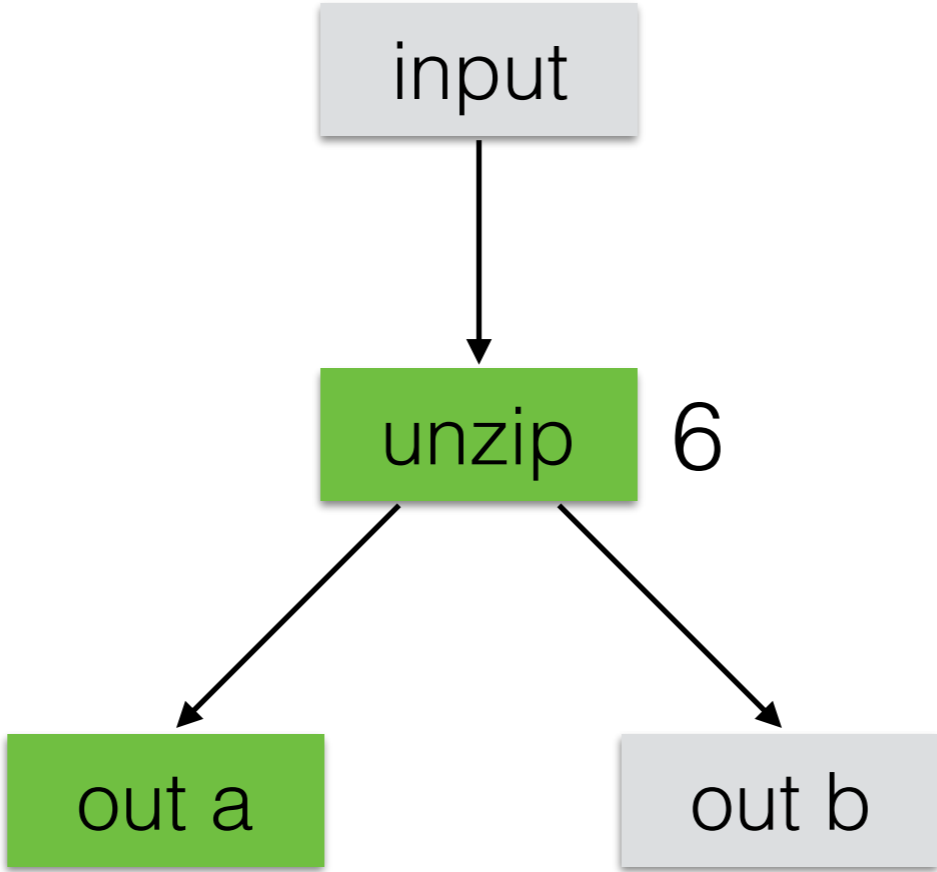


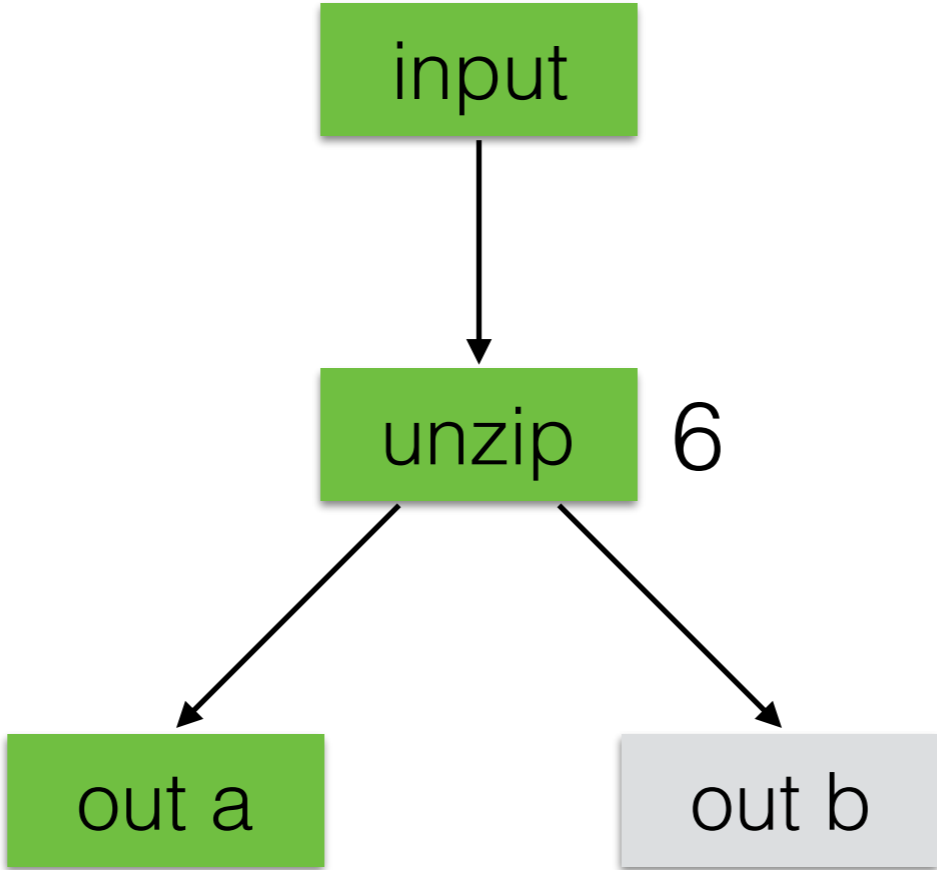


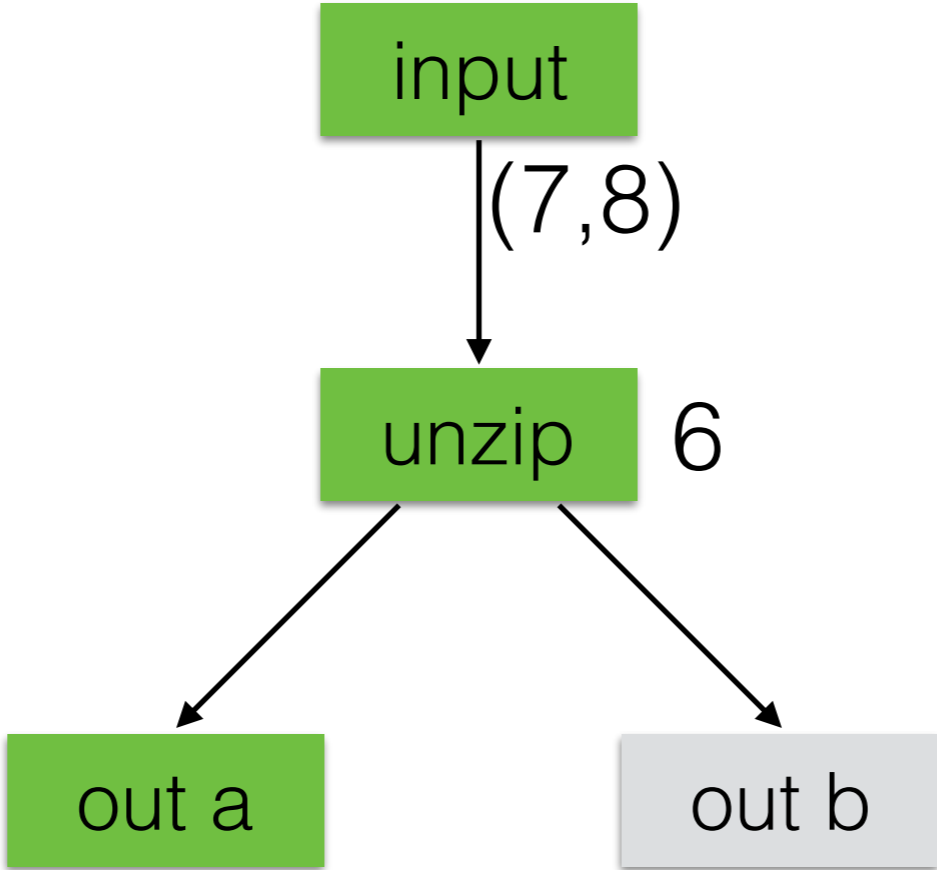


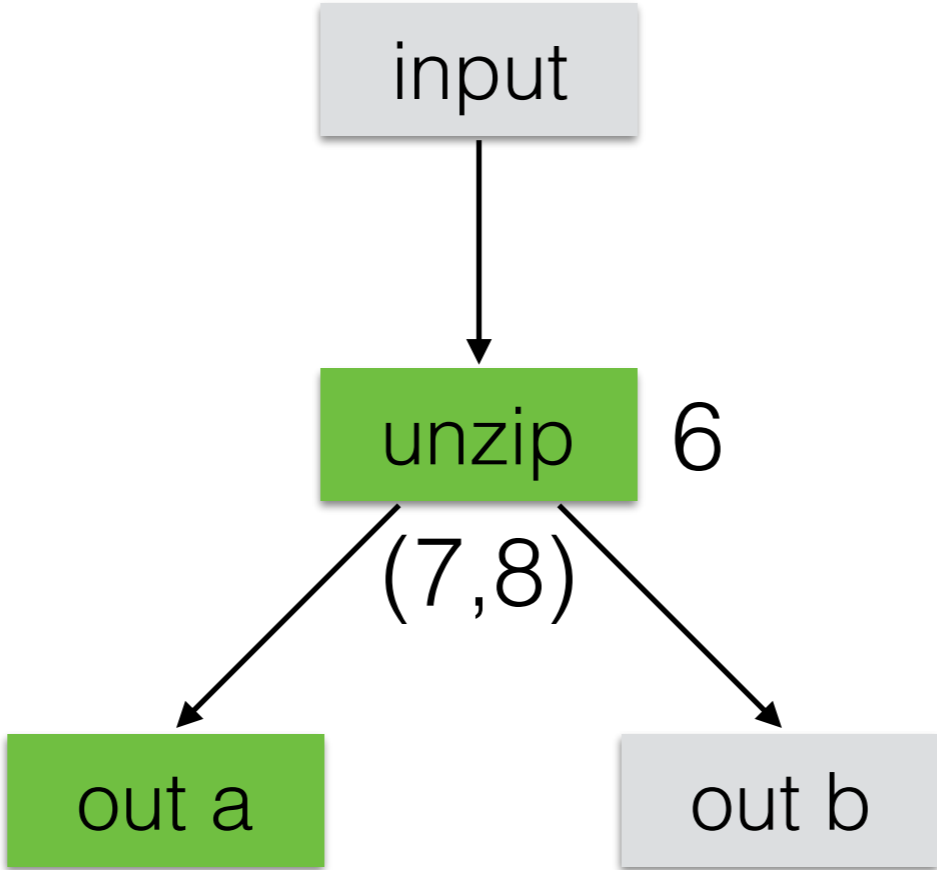


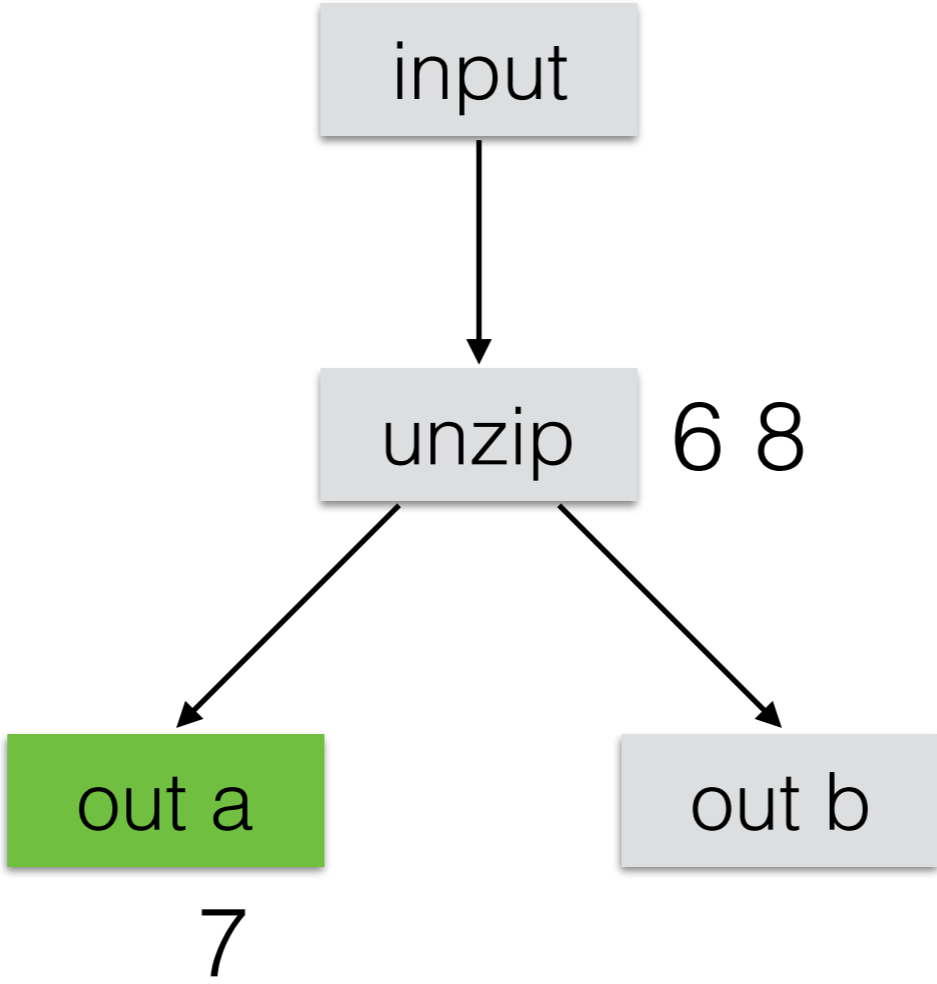


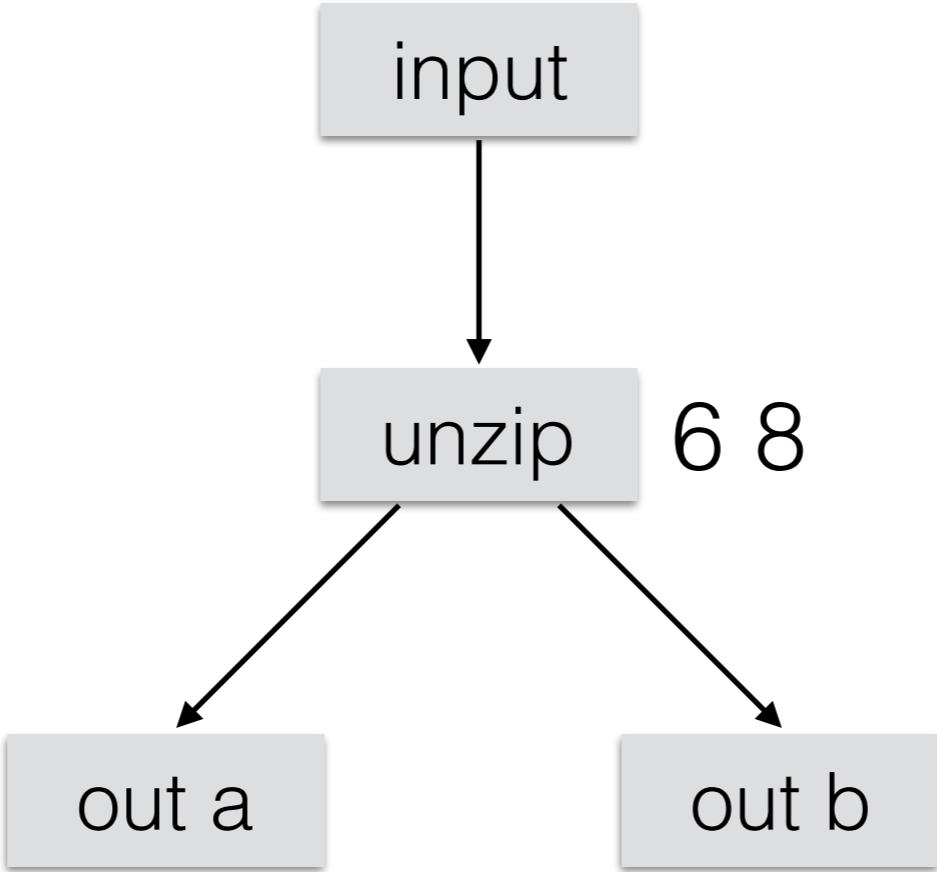




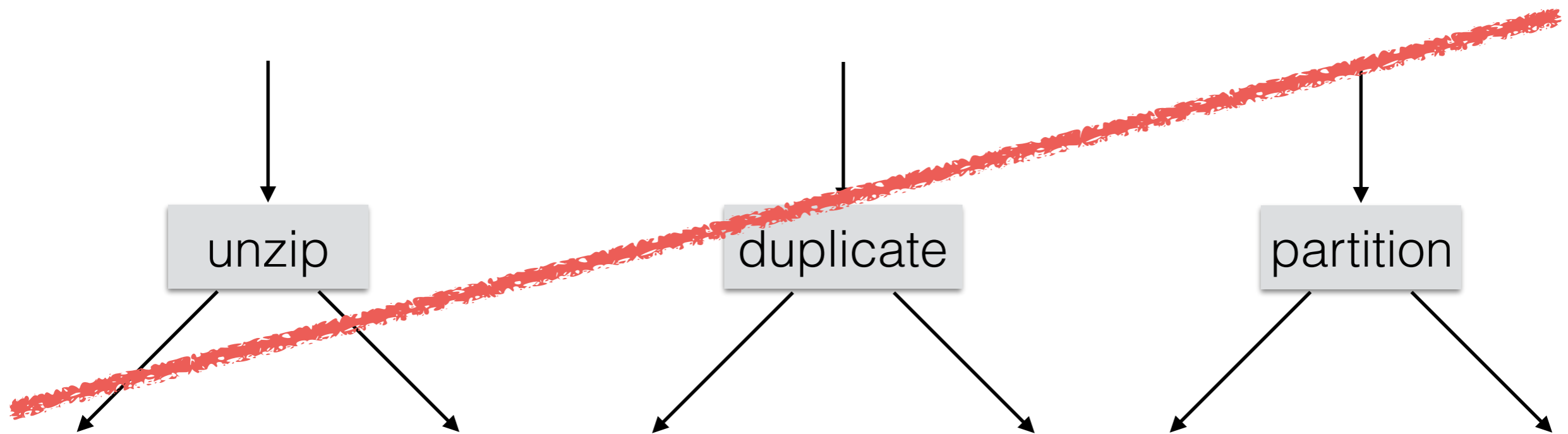




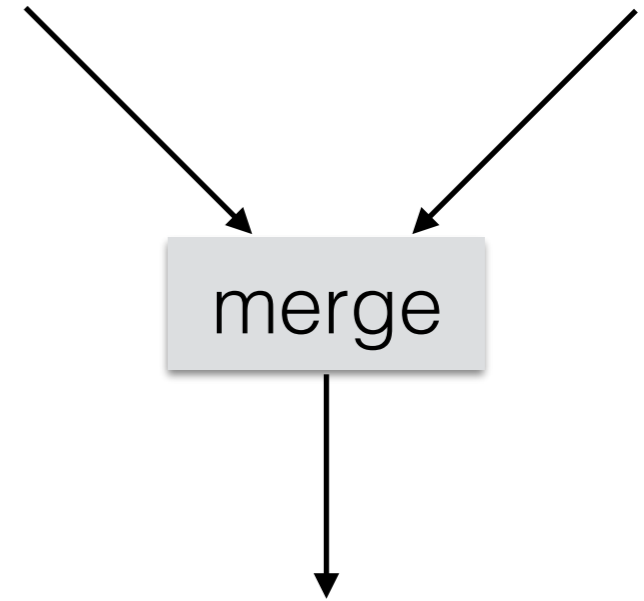
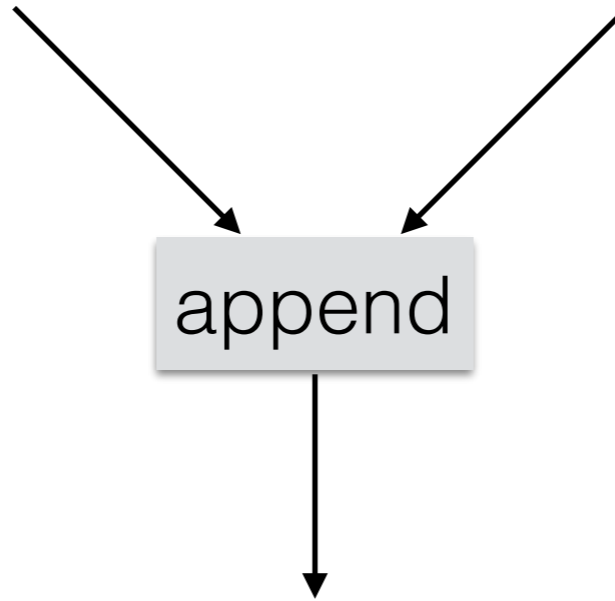
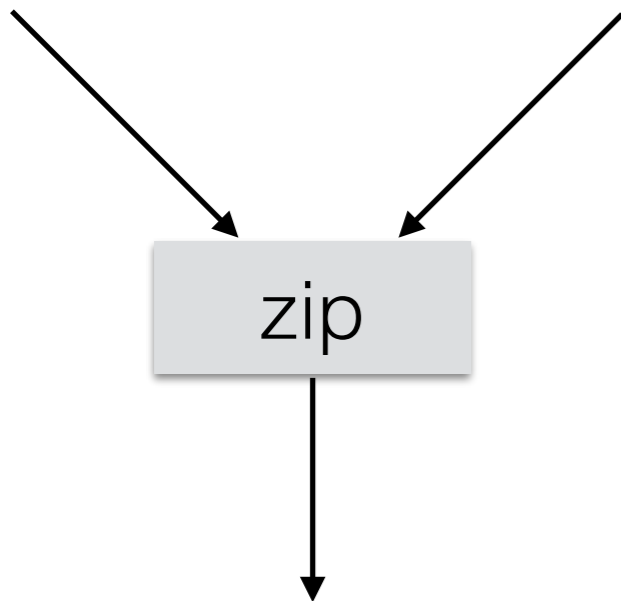




Pull: bad



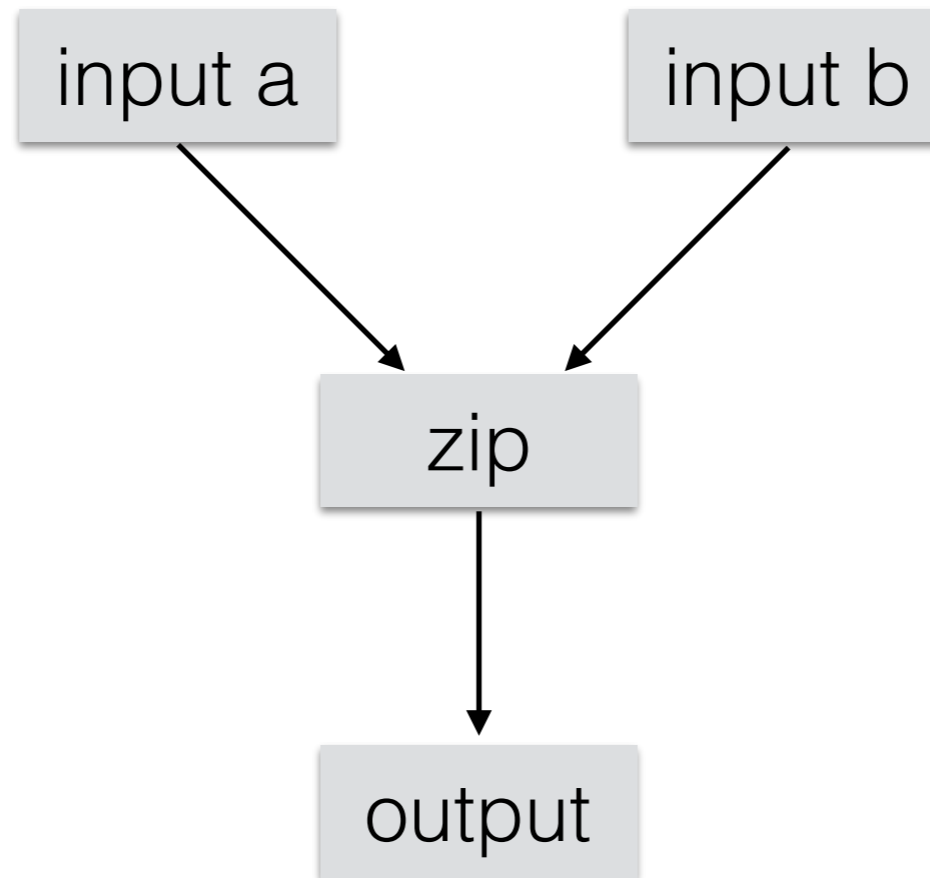
Pull: good



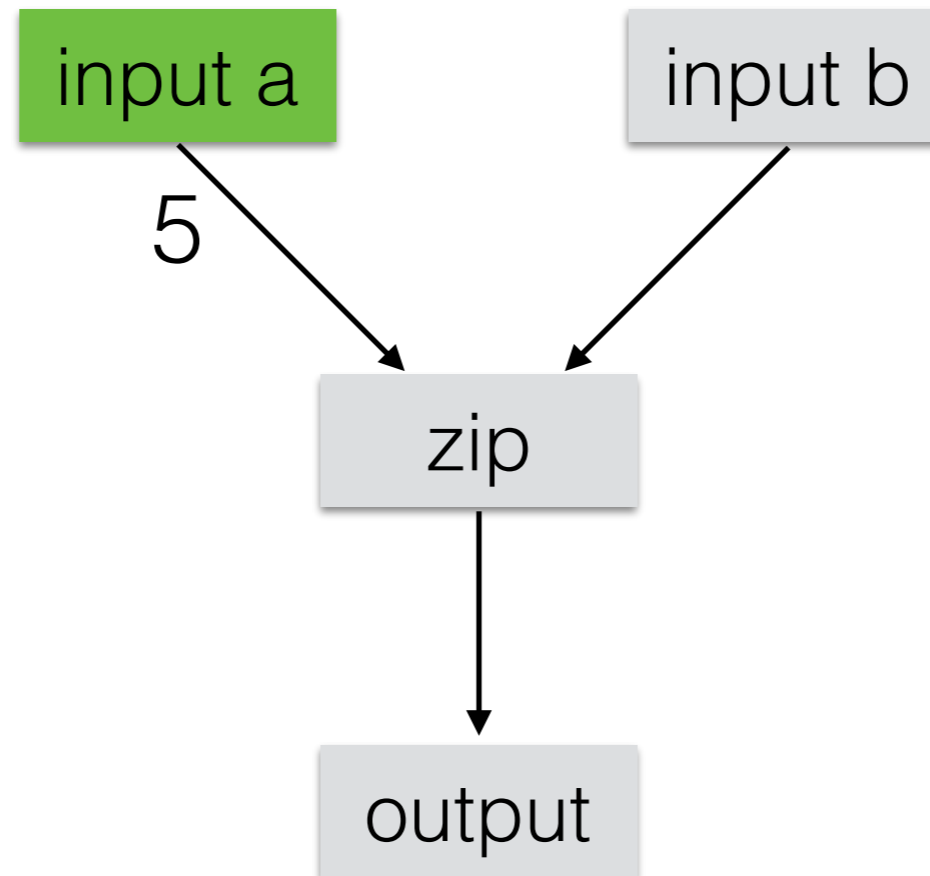
Push streams

```
type Push a = Maybe a -> IO ()
```

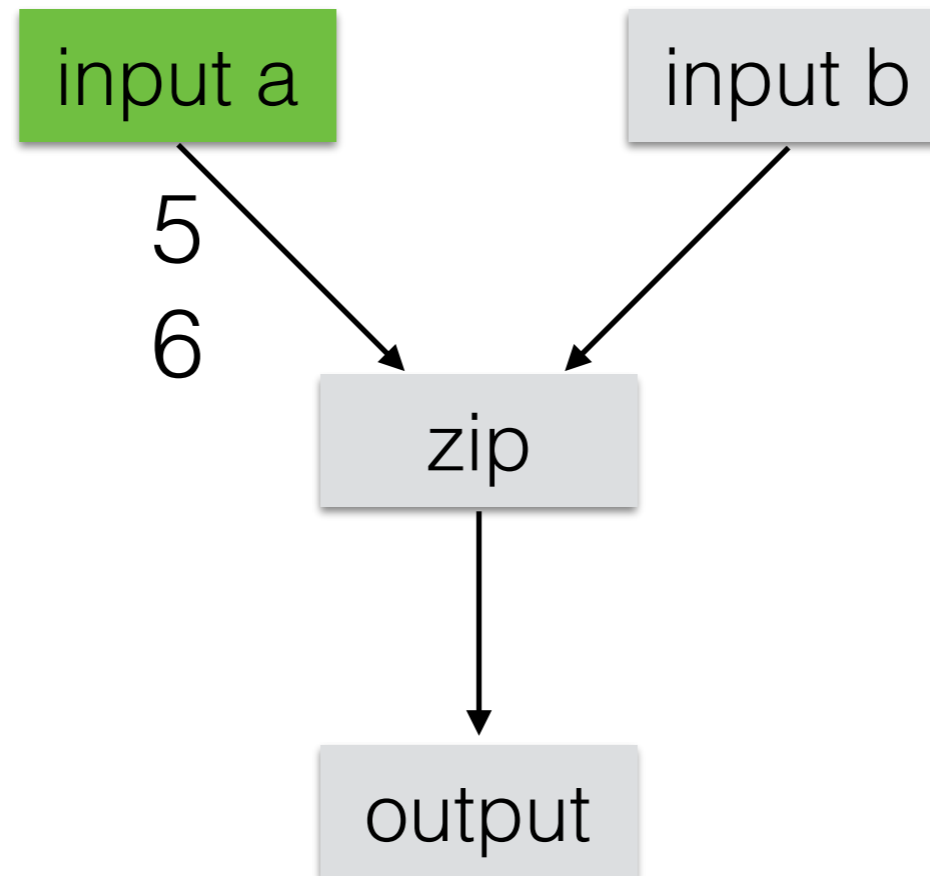
Zip



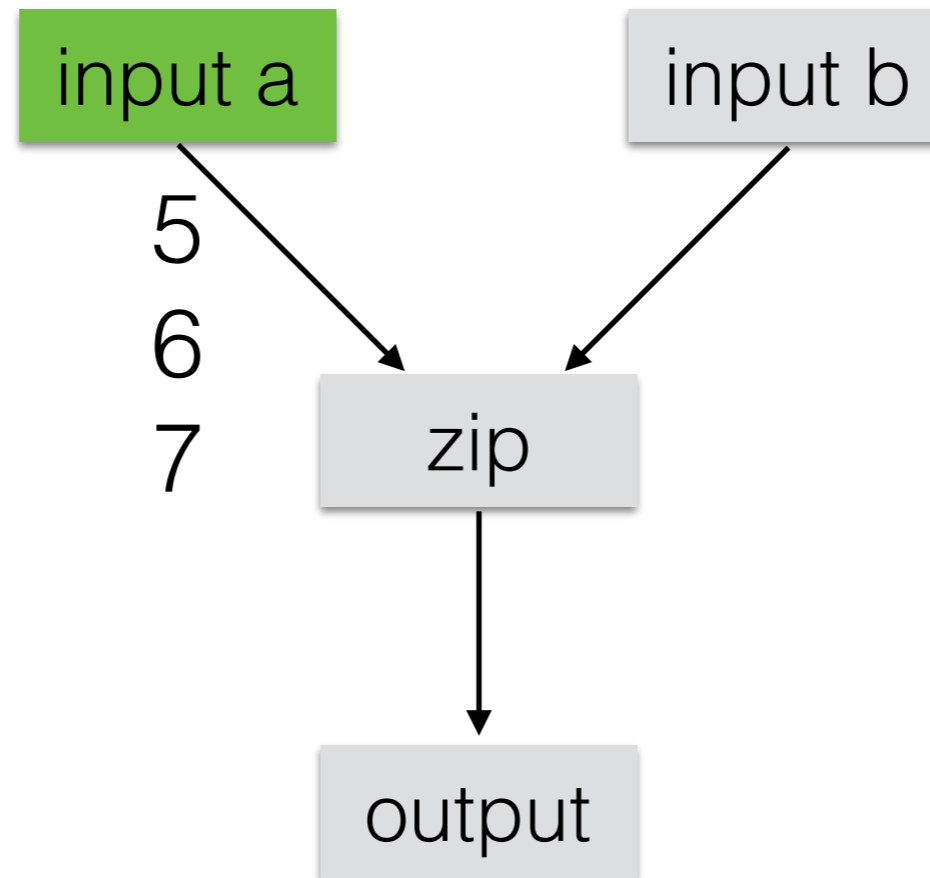
Zip



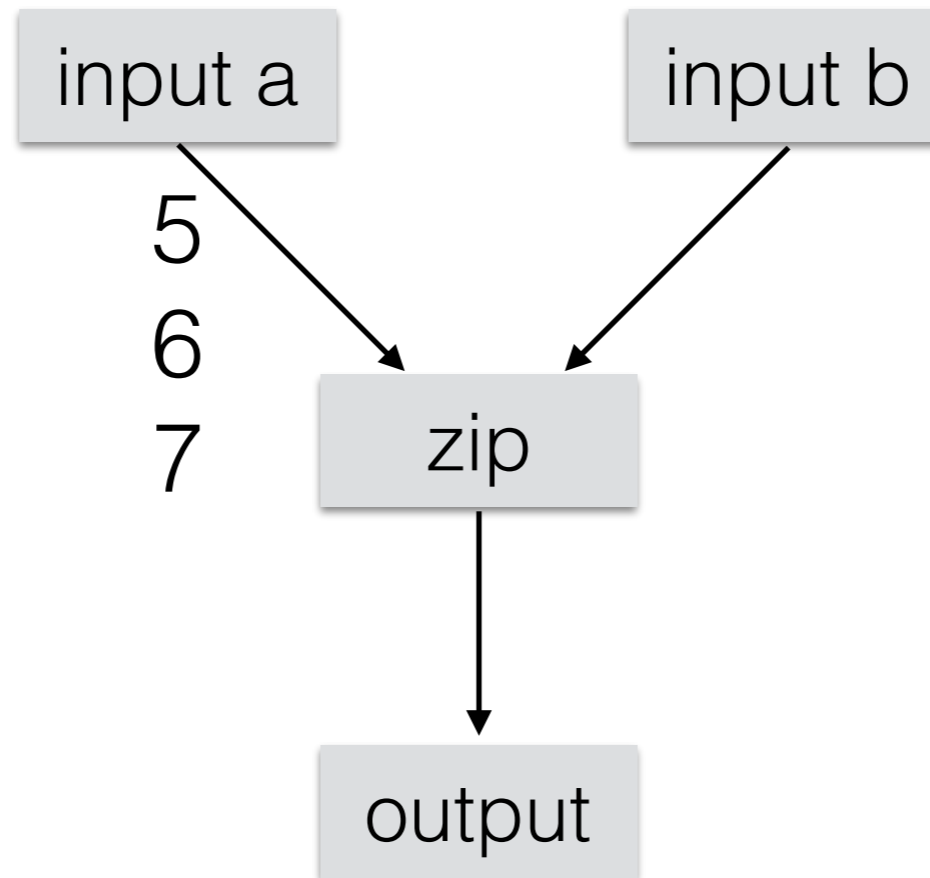
Zip



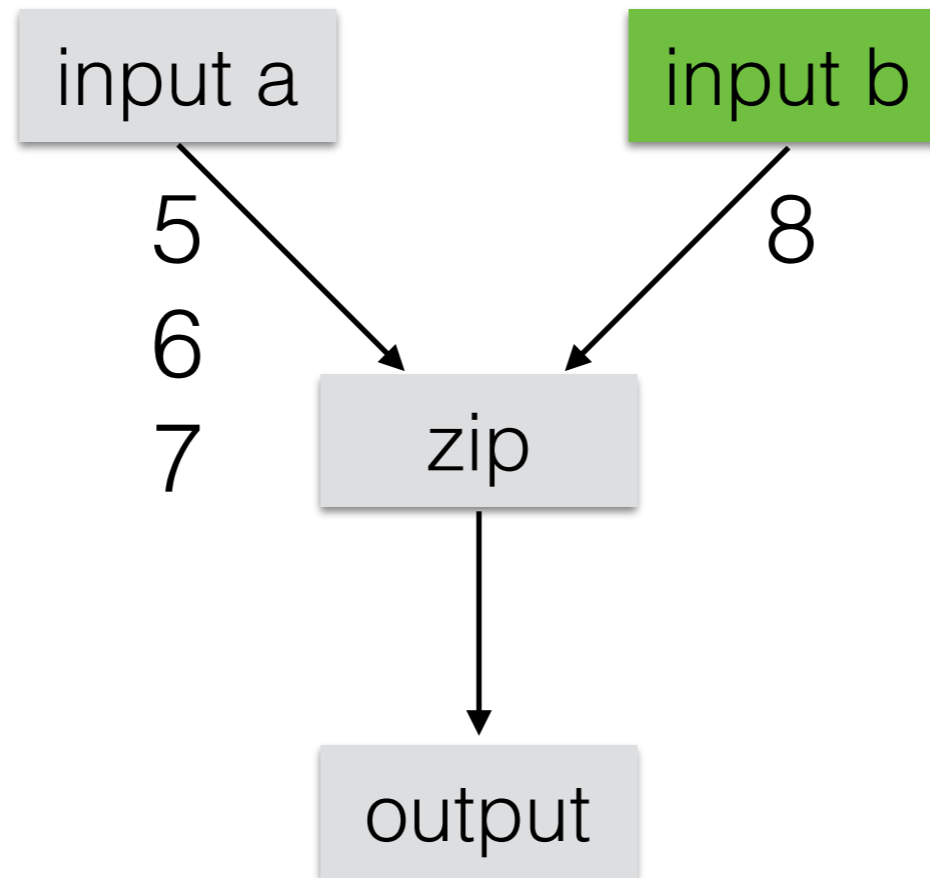
Zip



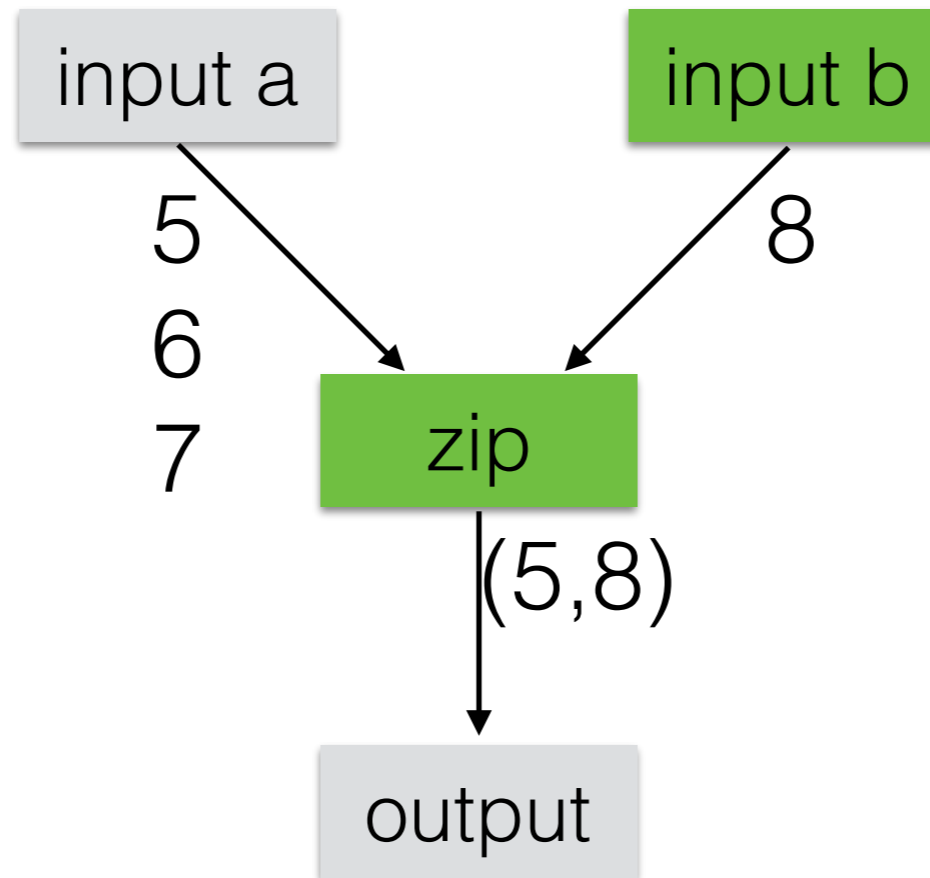
Zip



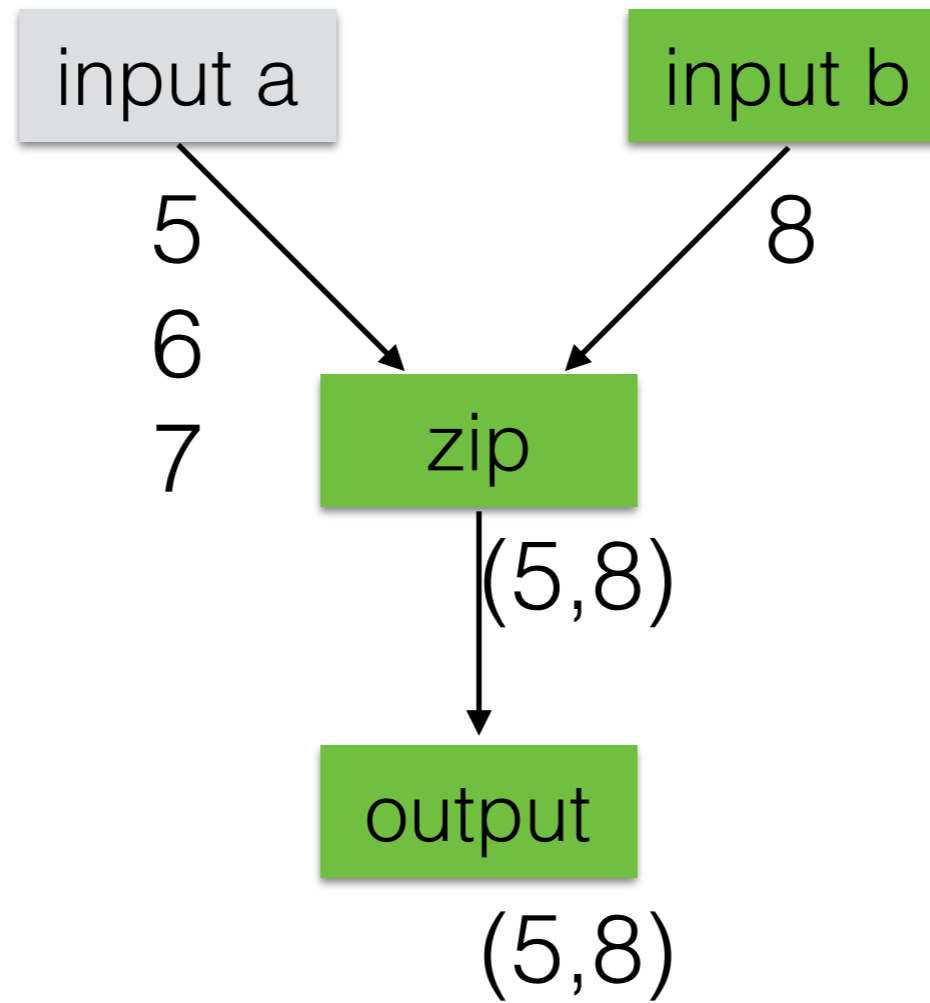
Zip



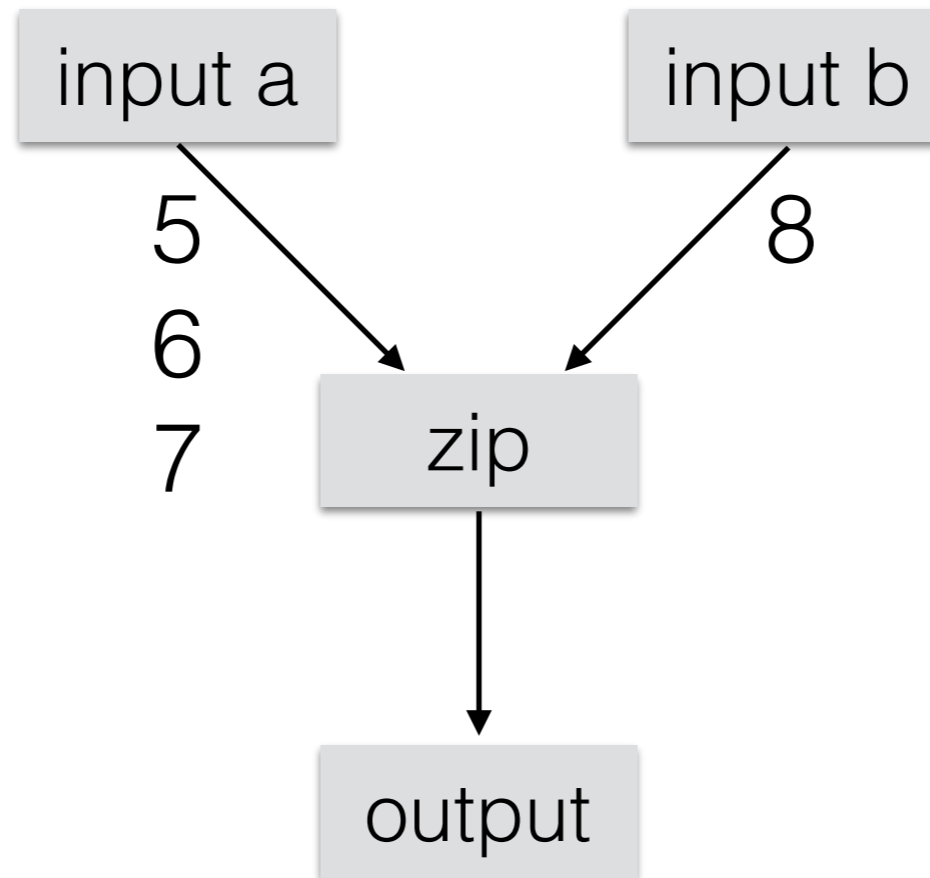
Zip



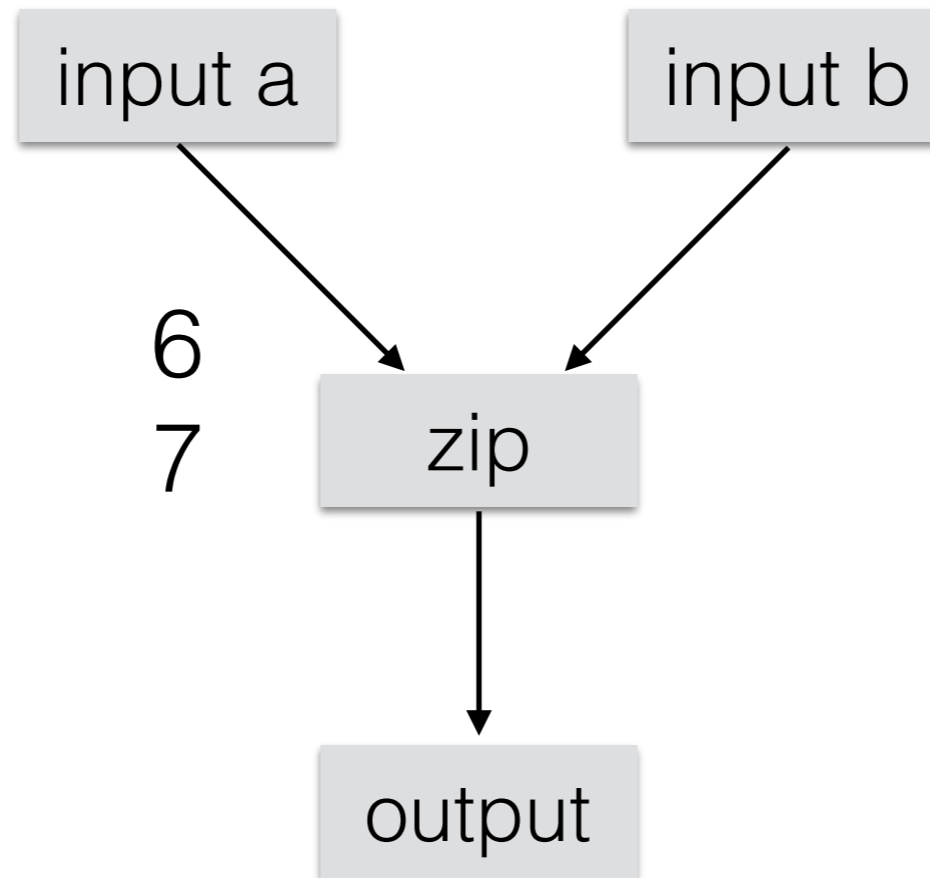
Zip



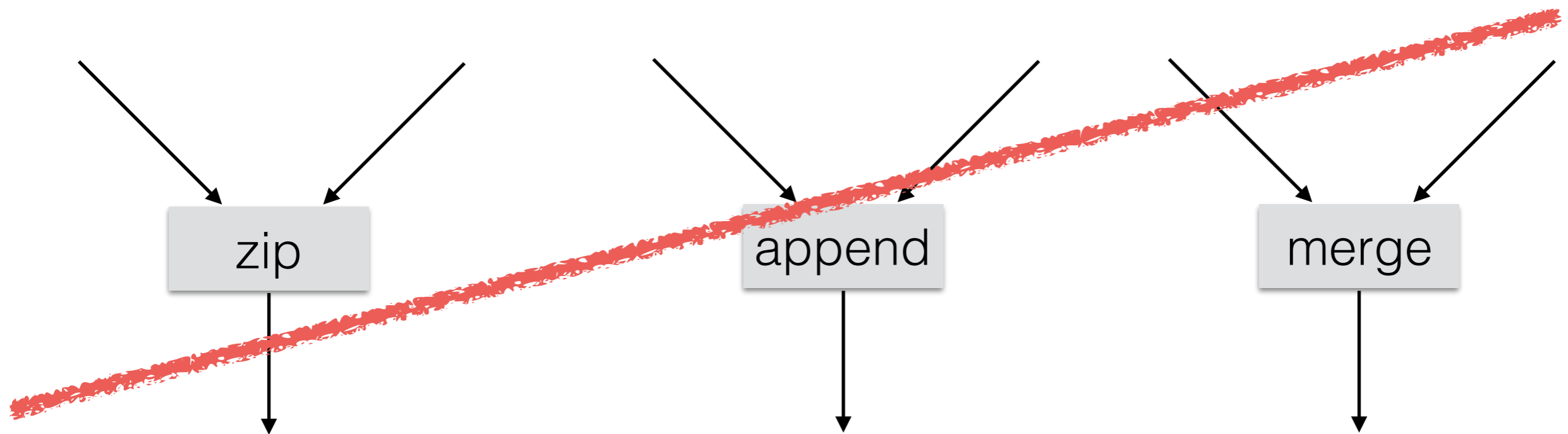
Zip



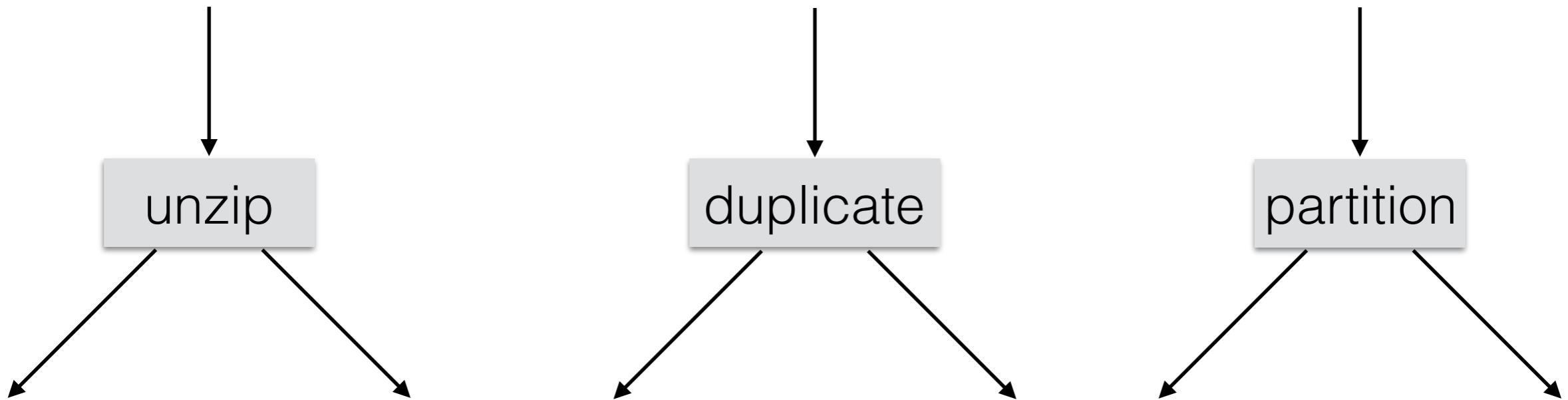
Zip



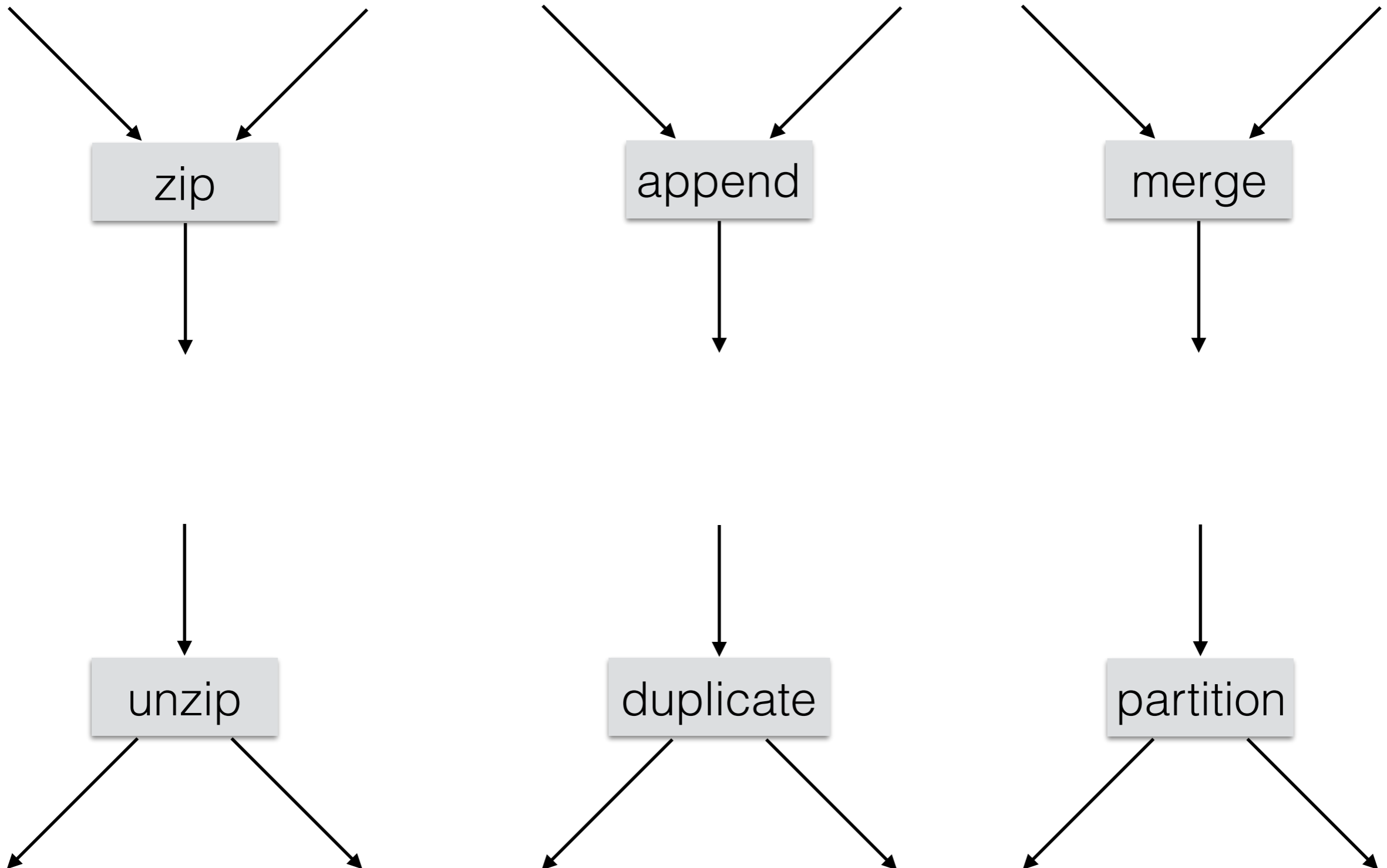
Push: bad



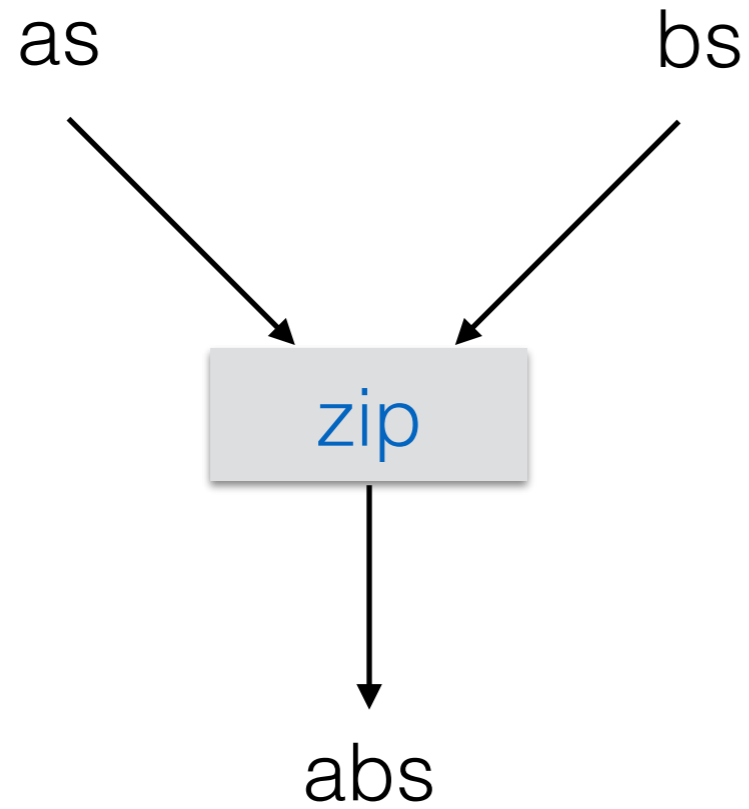
Push: good



Pull in, push out

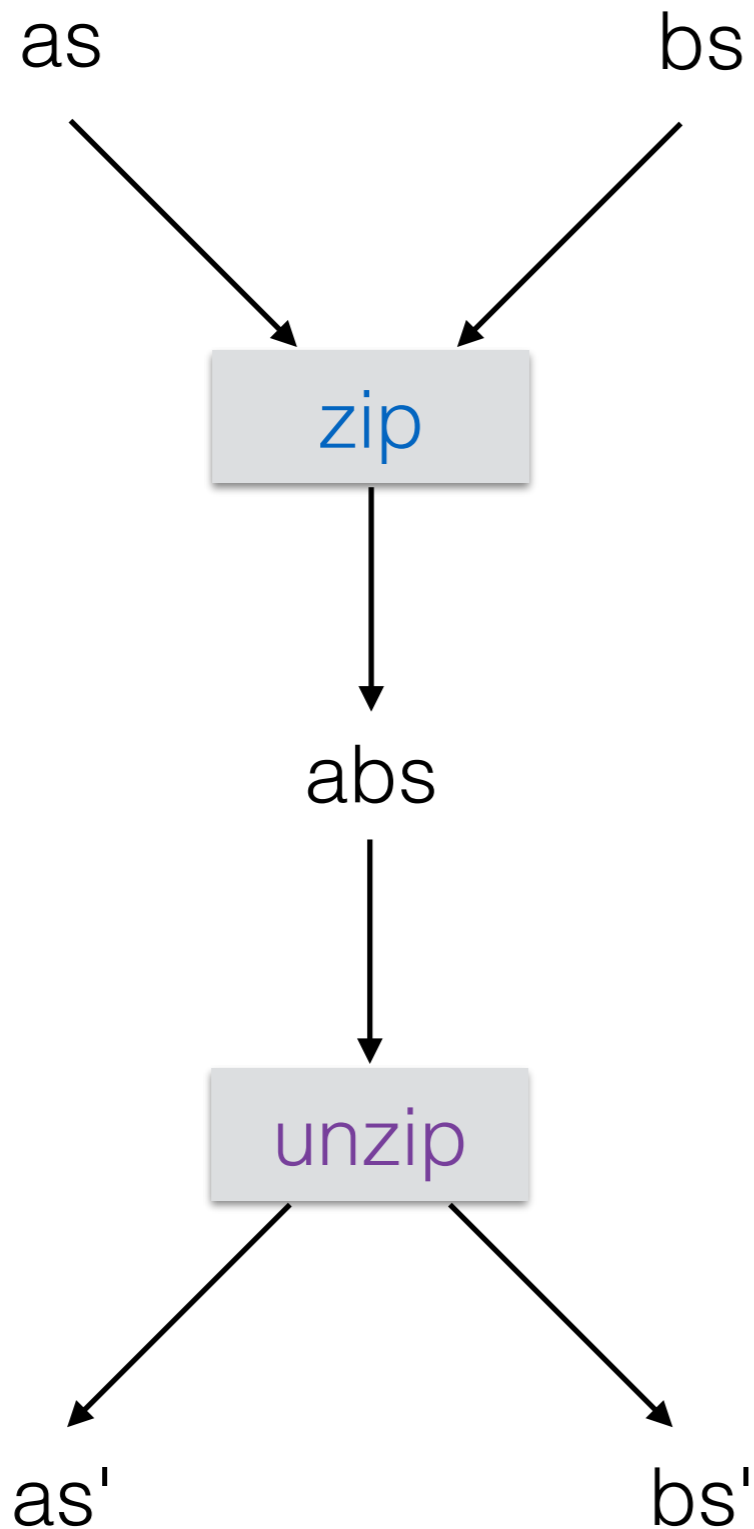


Processes



```
a = pull as  
b = pull bs  
push abs (a,b)
```

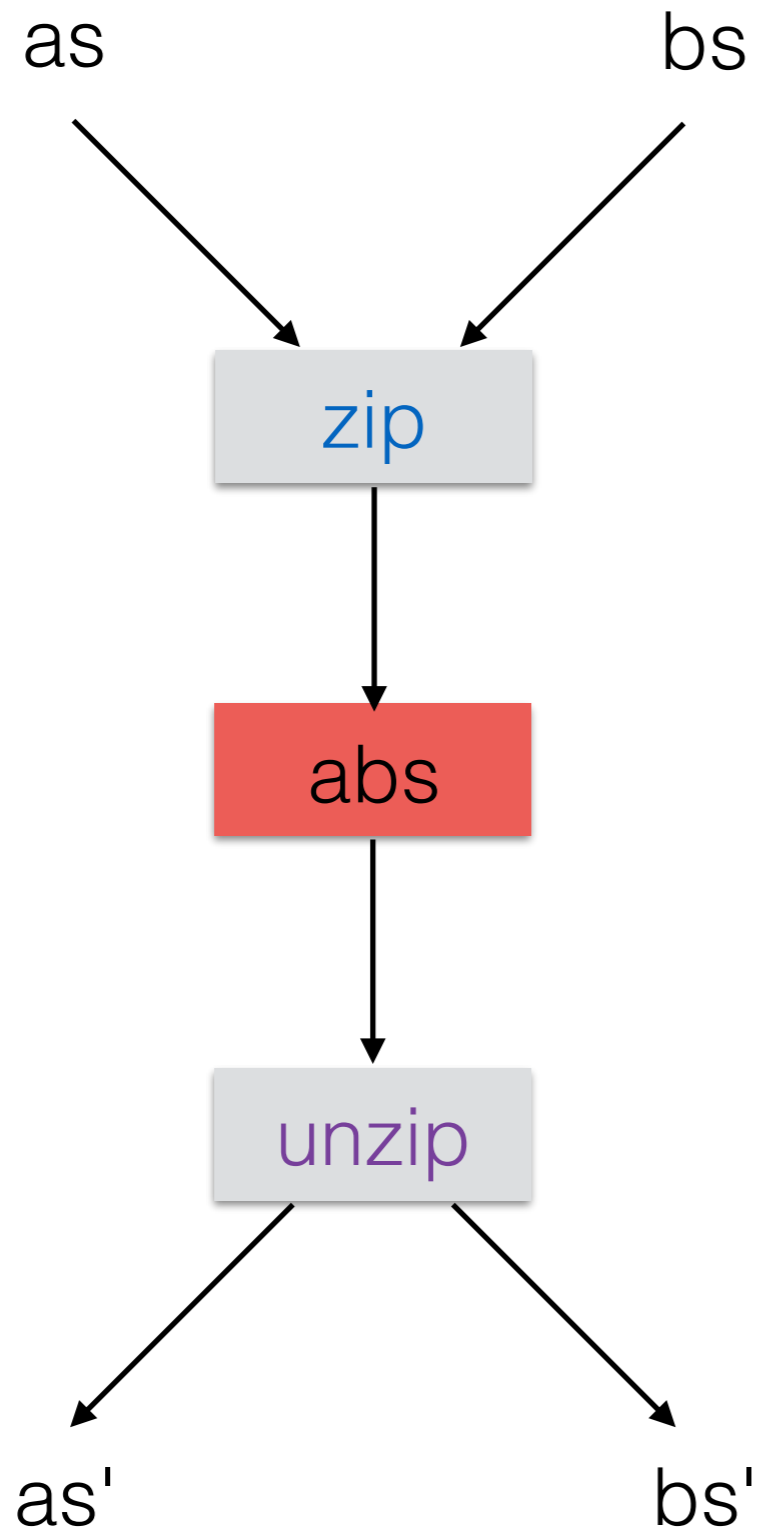
Processes



```
a = pull as  
b = pull bs  
push abs (a,b)
```

```
ab = pull abs  
push as' (fst ab)  
push bs' (snd ab)
```

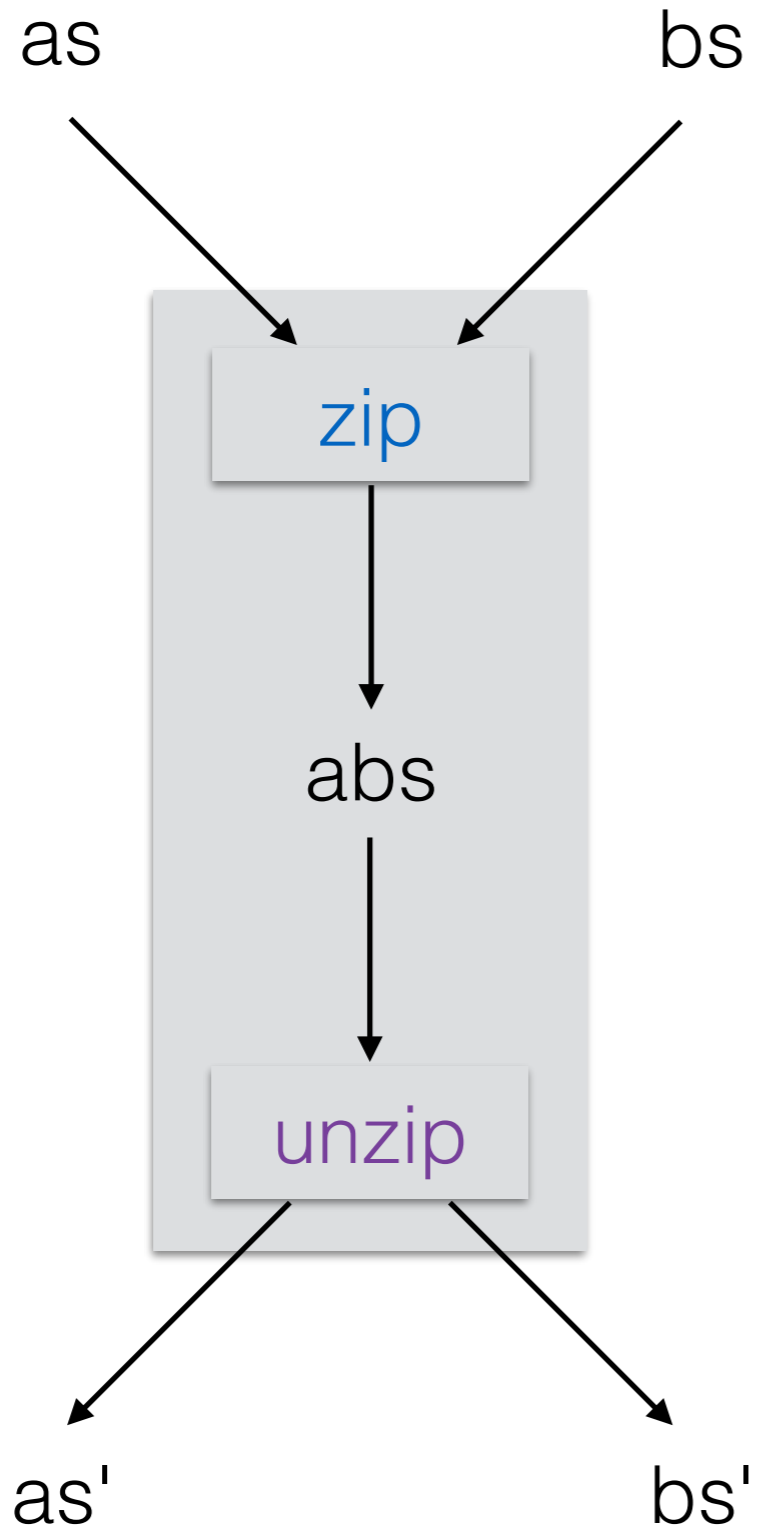
Communication overhead



```
a = pull as  
b = pull bs  
push abs (a,b)
```

```
ab = pull abs  
push as' (fst ab)  
push bs' (snd ab)
```

Fusion



$a = \text{pull } as$

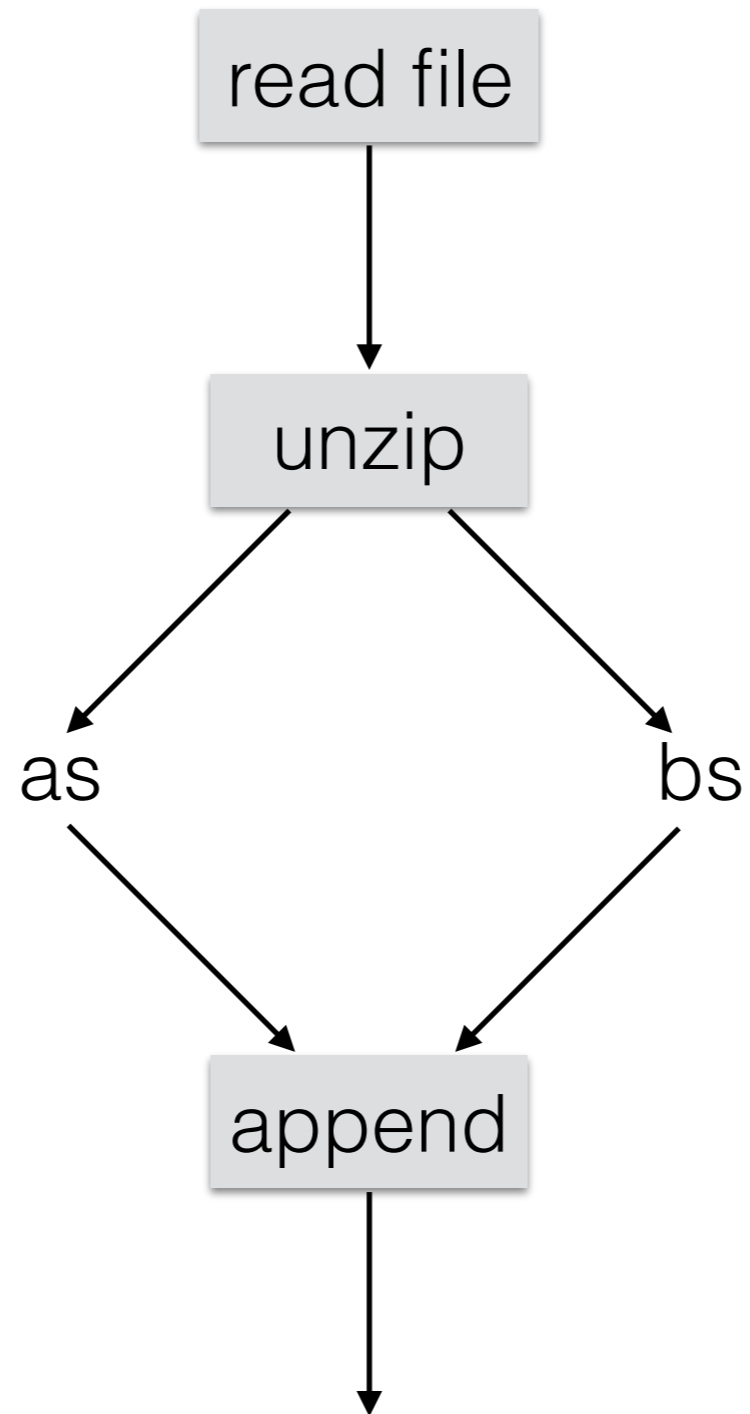
$b = \text{pull } bs$

$ab = (a, b)$

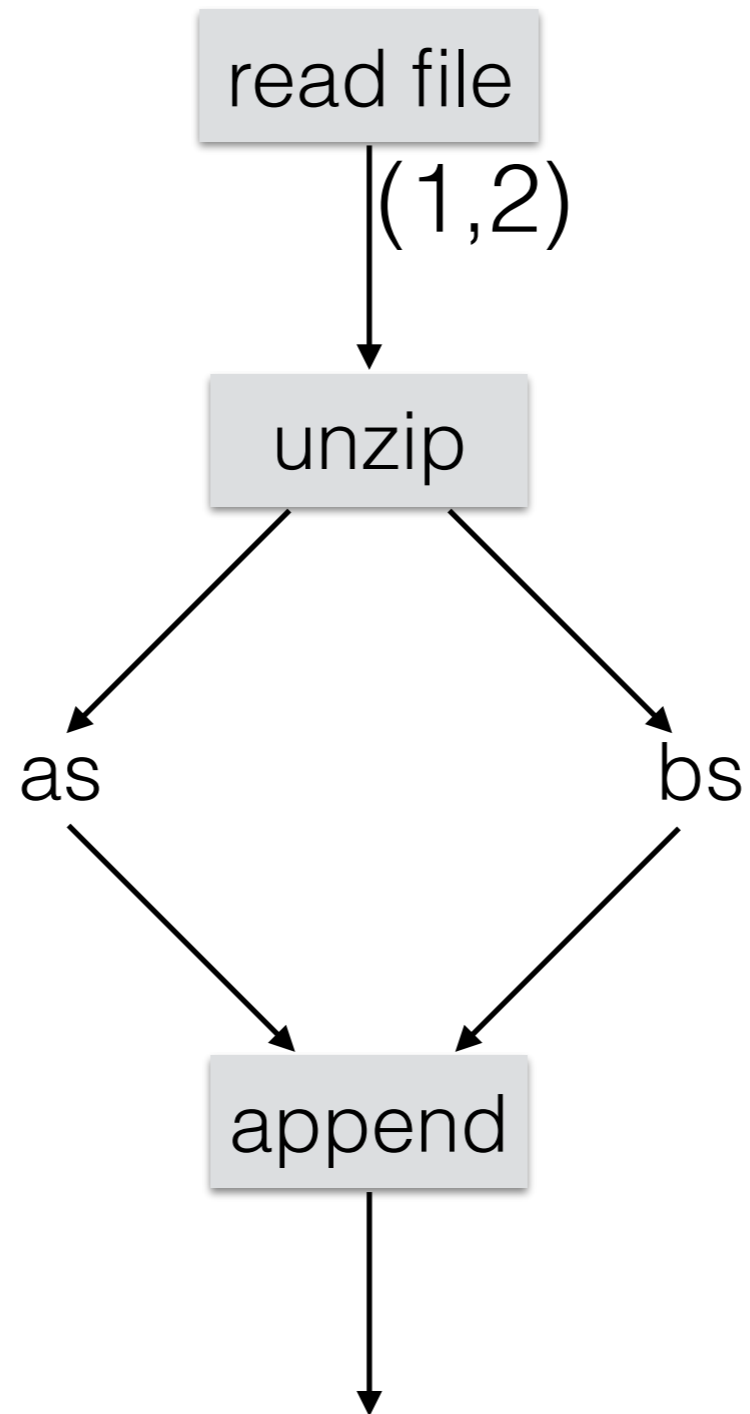
$\text{push } as' (\text{fst } ab)$

$\text{push } bs' (\text{snd } ab)$

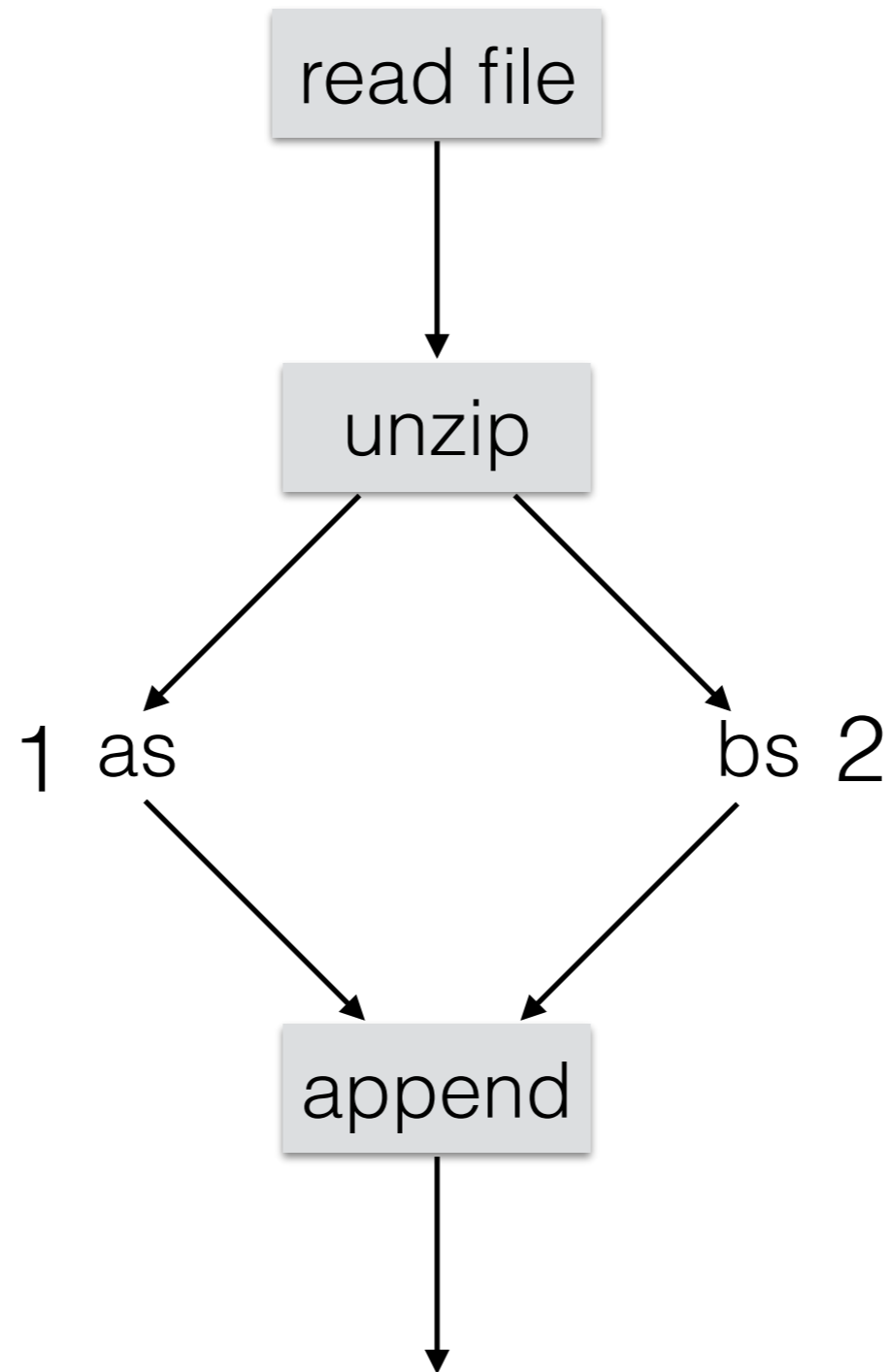
Unbounded buffers still required



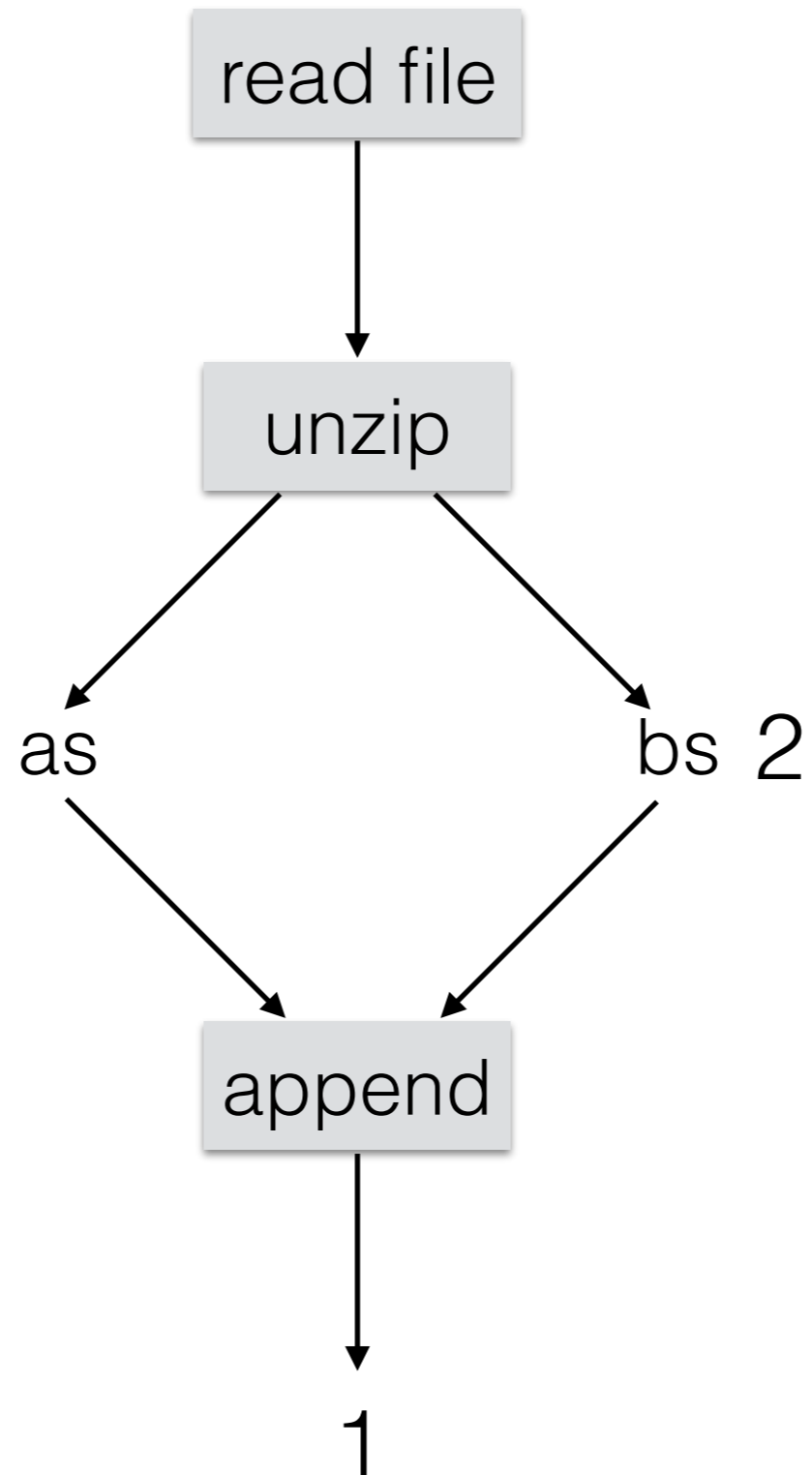
Unbounded buffers still required



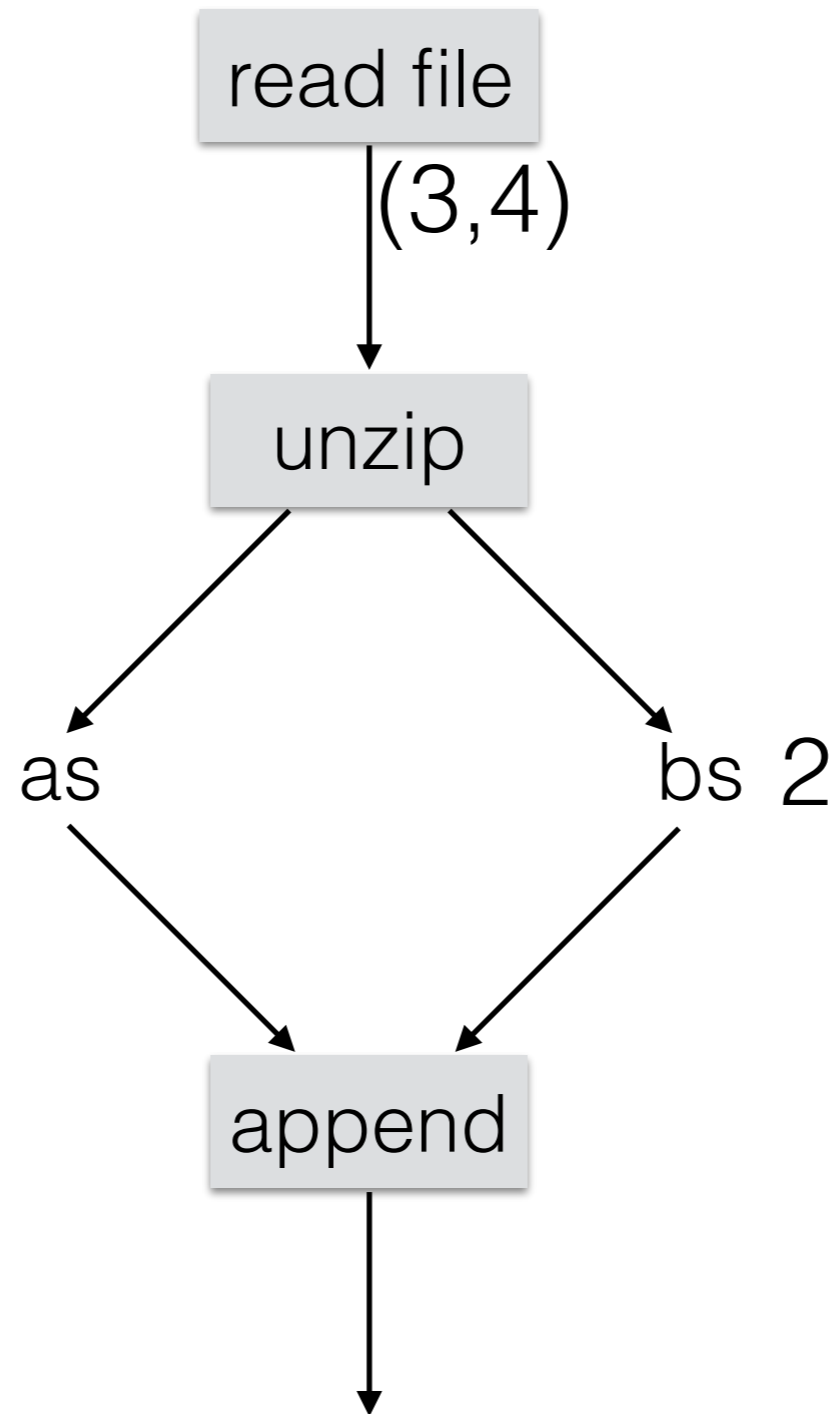
Unbounded buffers still required



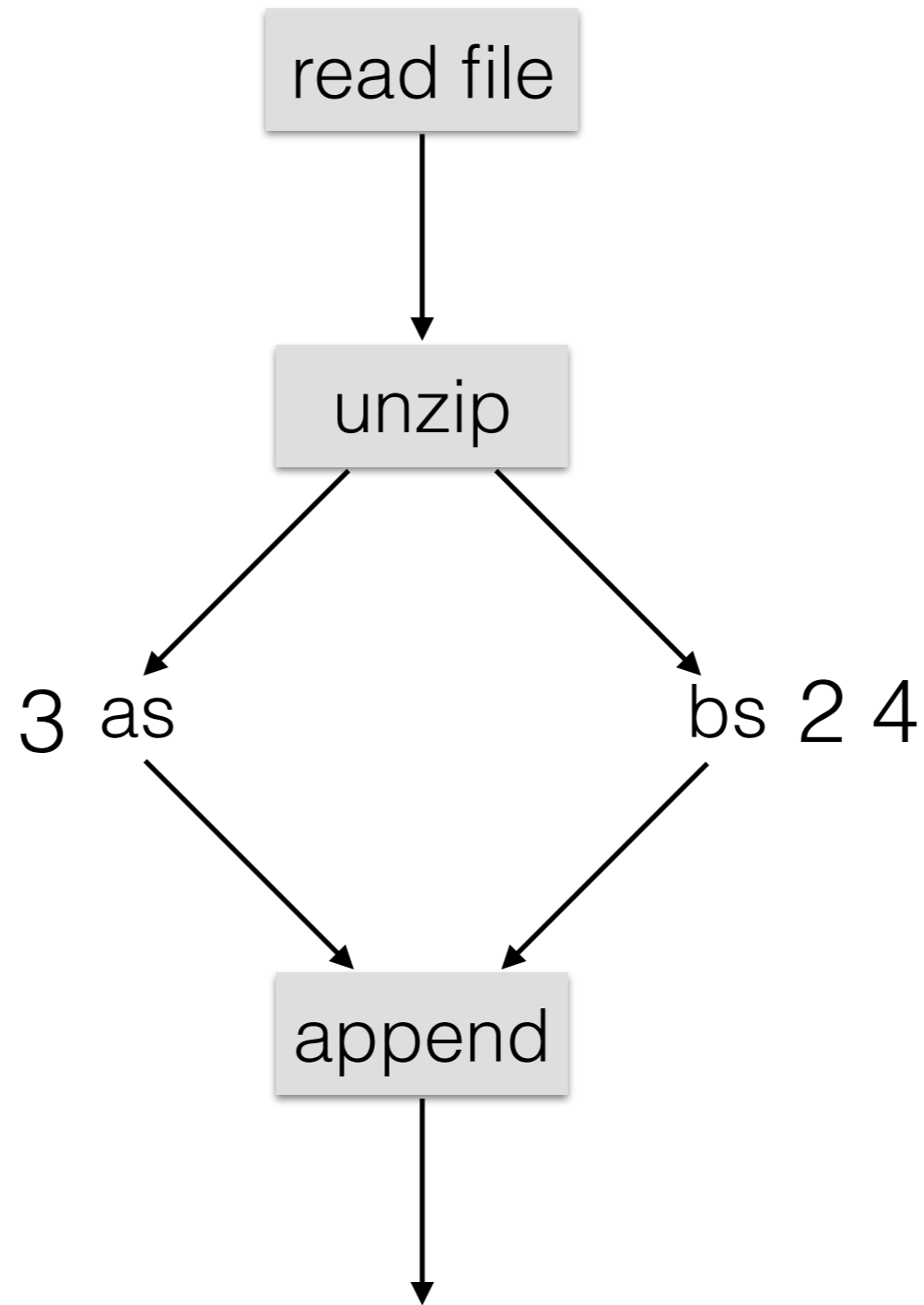
Unbounded buffers still required



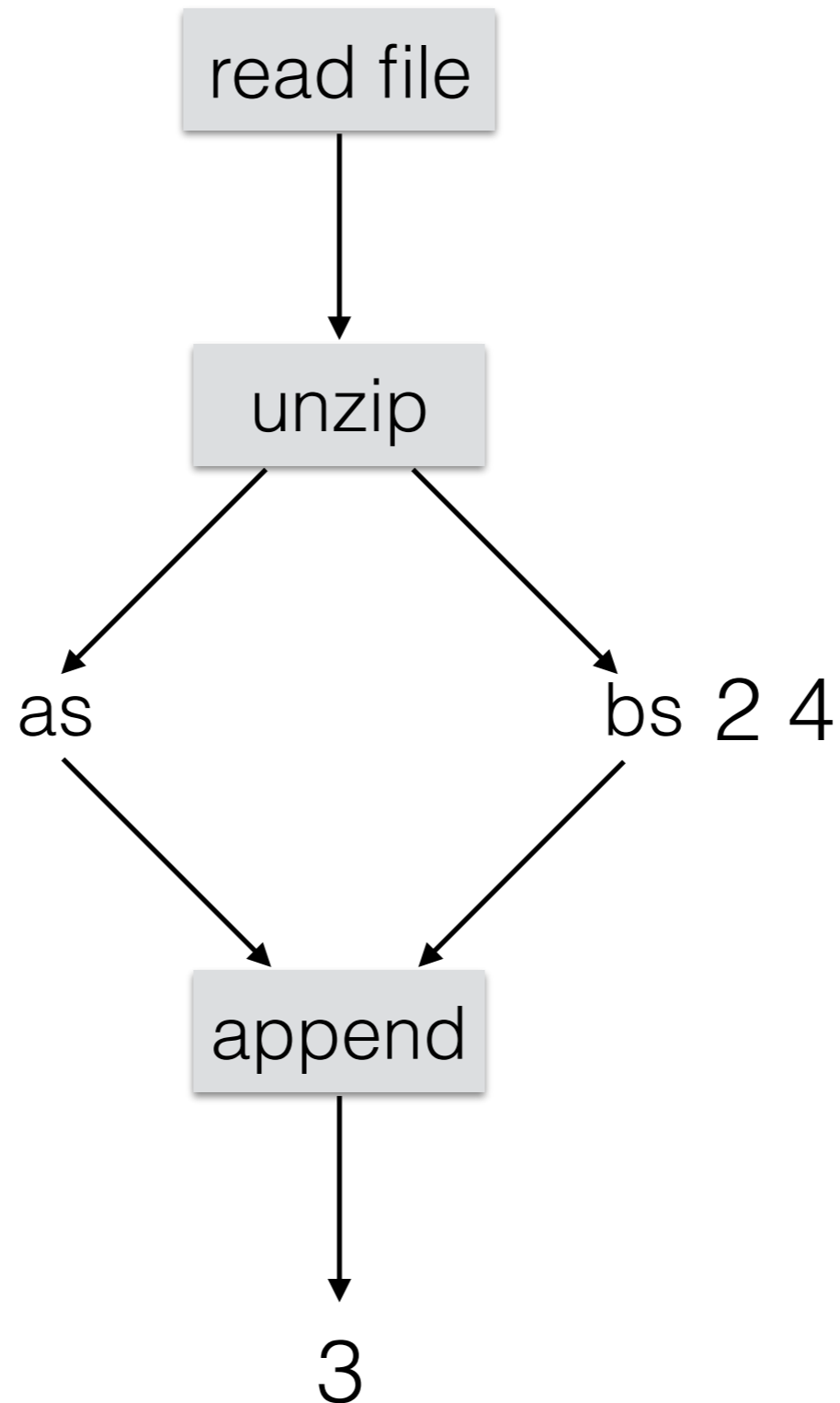
Unbounded buffers still required



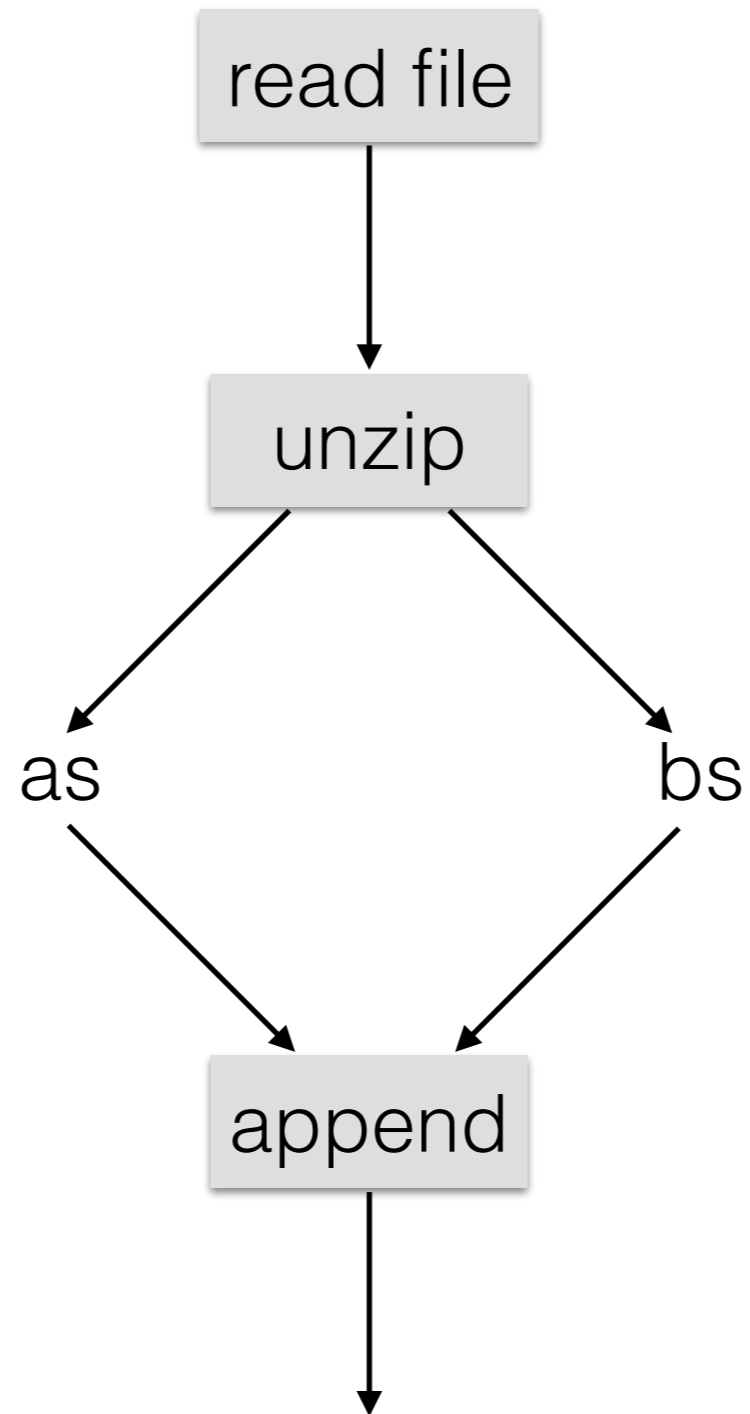
Unbounded buffers still required



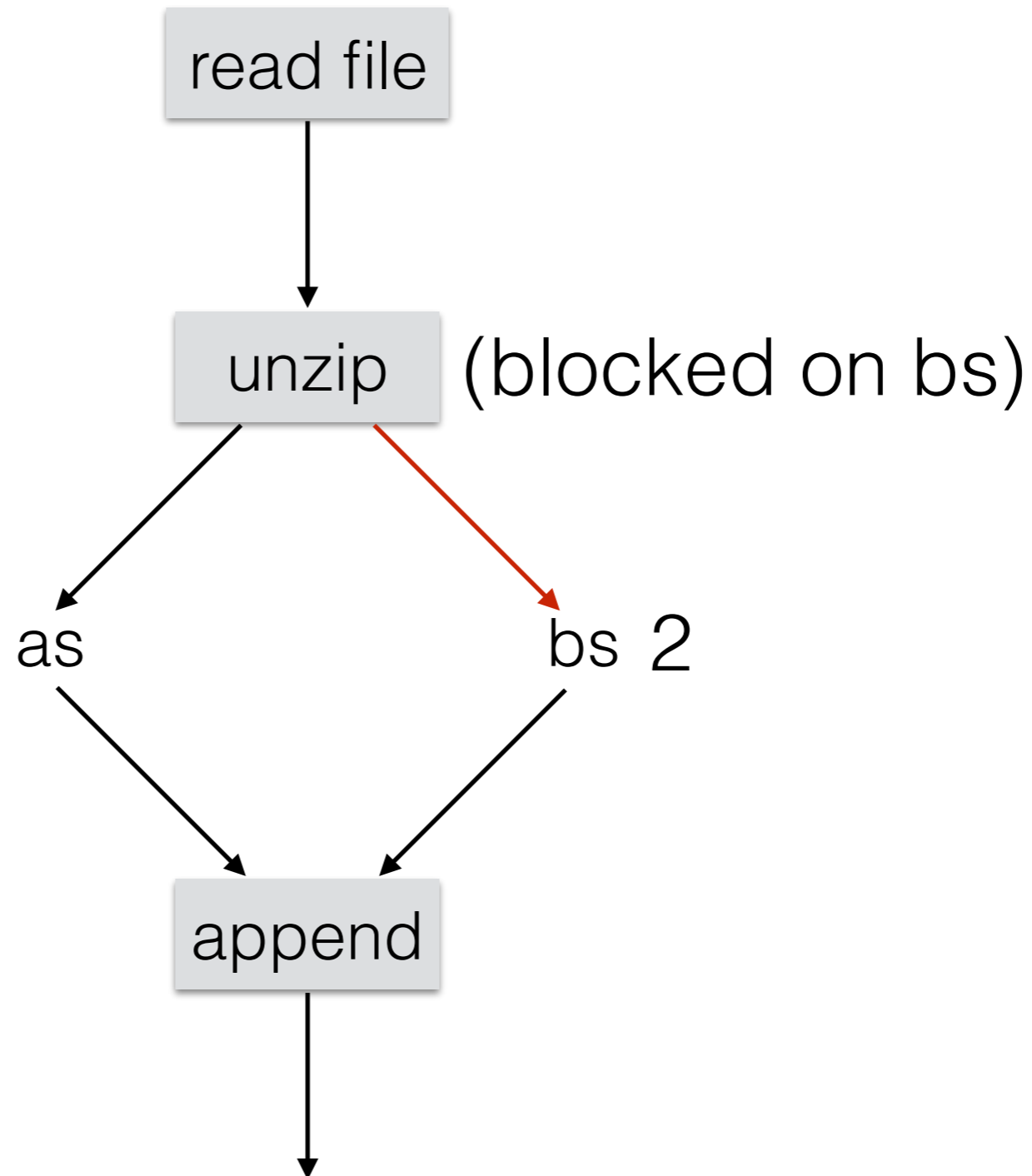
Unbounded buffers still required



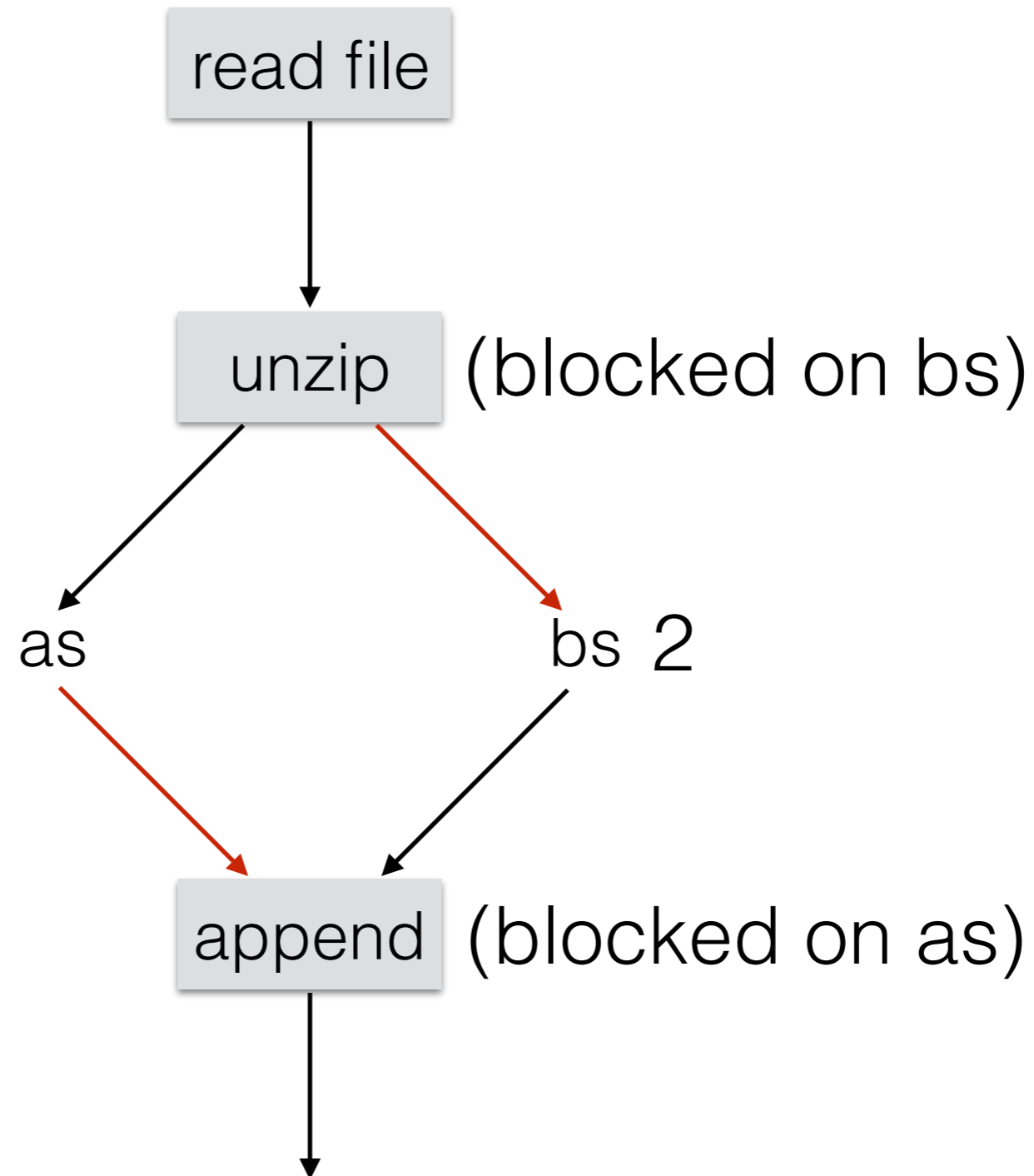
Inherent buffering



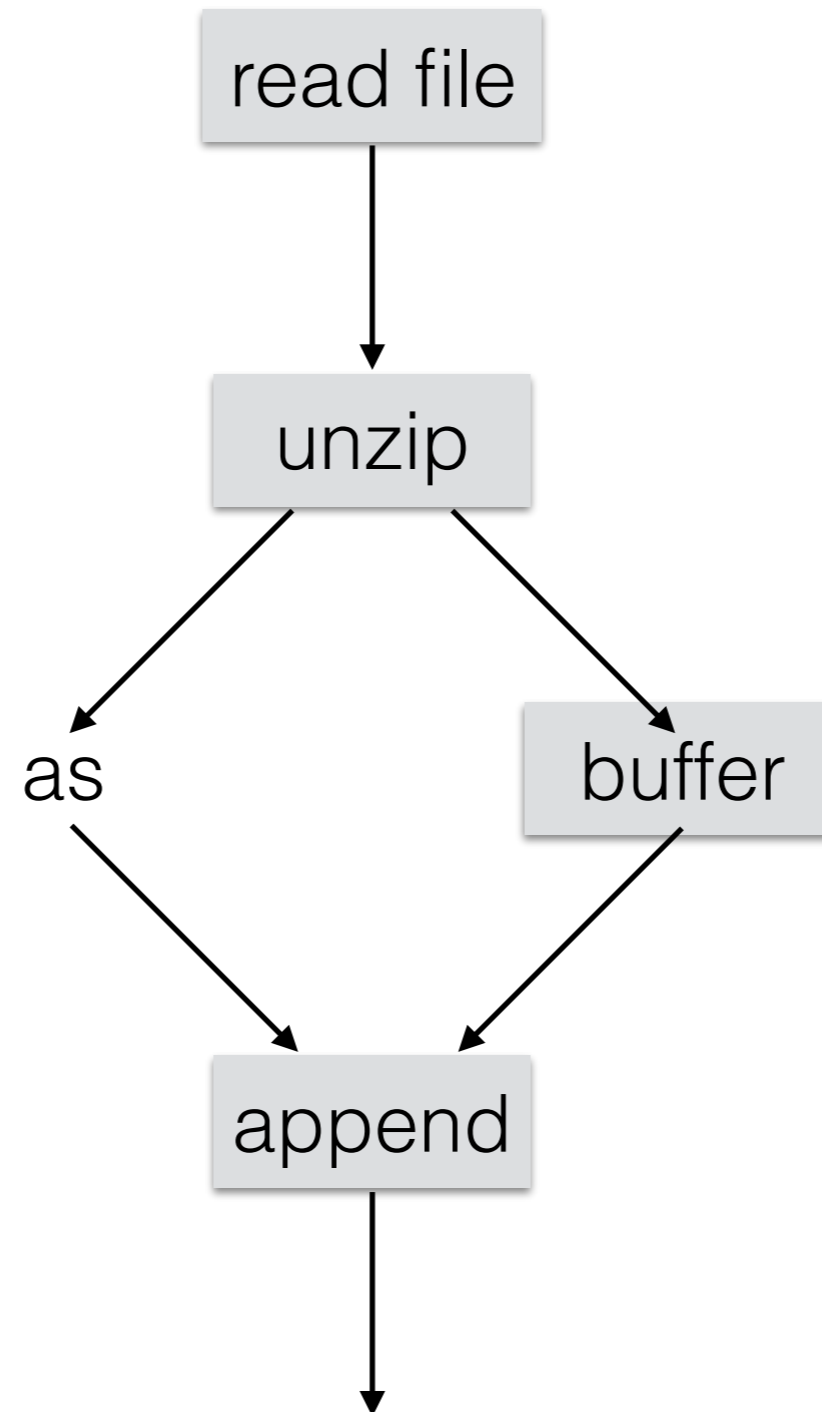
Blocking push



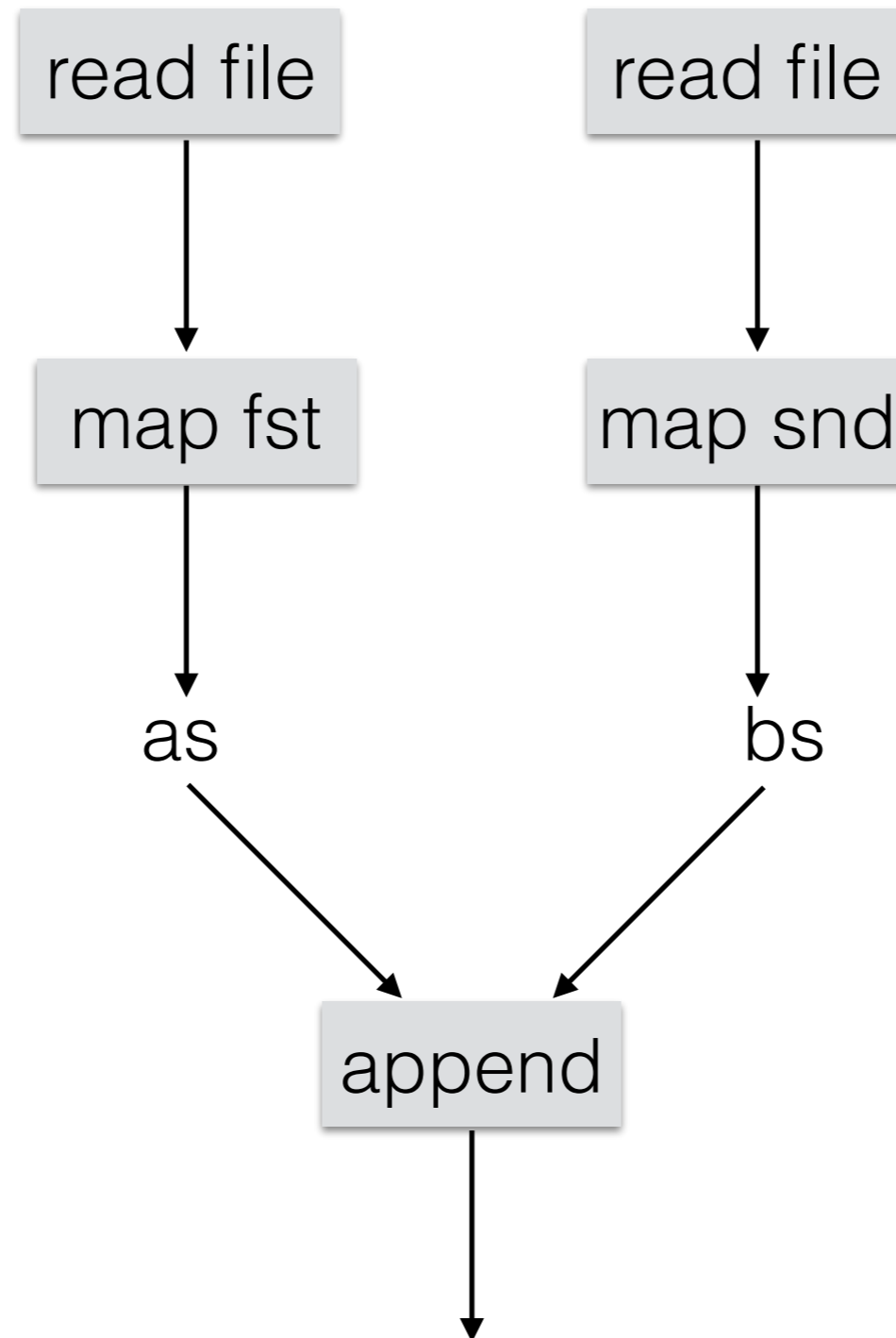
Deadlock



Explicit buffer



Explicit duplication



Kahn Process Networks

Expressive

Fuse processes to remove overhead

Detect unbounded buffers so user chooses