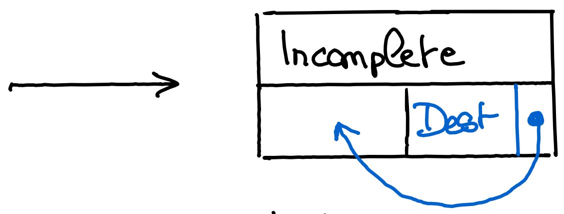
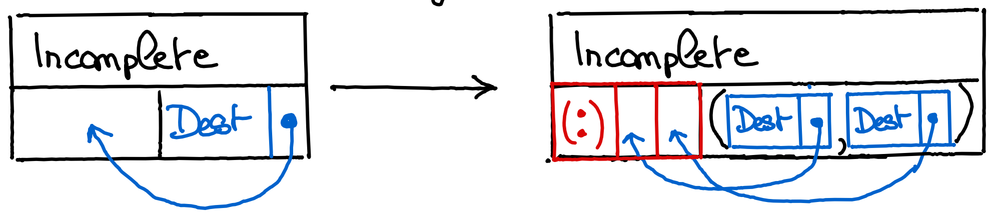
1

i1 = append alloc 1

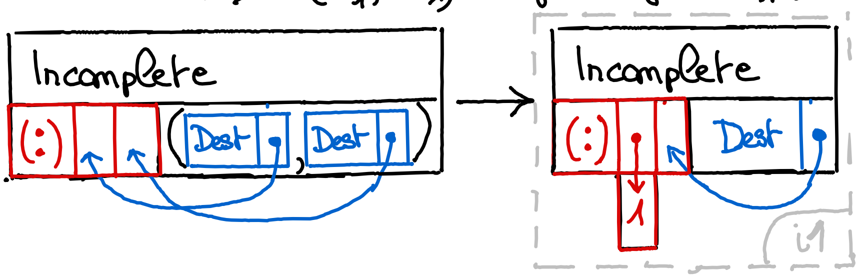
```
alloc :: Incomplete [a] (Dest [a])
```



```
... <&> \d1 -> fill @'(:) d1
```



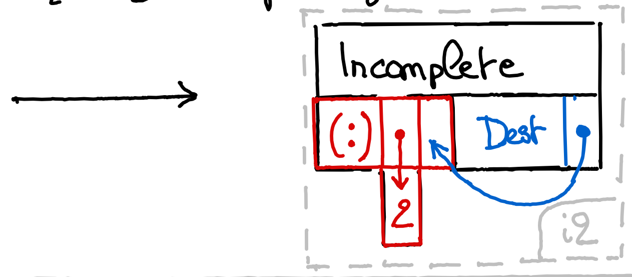
```
... <&> \(dh1, dt1) -> fillLeaf 1 dh1 ;; dt1
```



2

i2 = append alloc 2

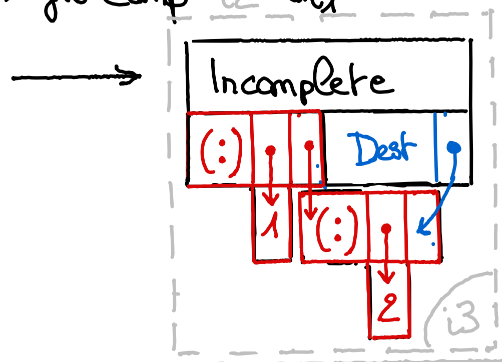
```
alloc <&> \d2 -> case fill @'(:) d2 of
  (dh2, dt2) -> fillLeaf 2 dh2 ;; dt2
```



3

i3 = concat i1 i2

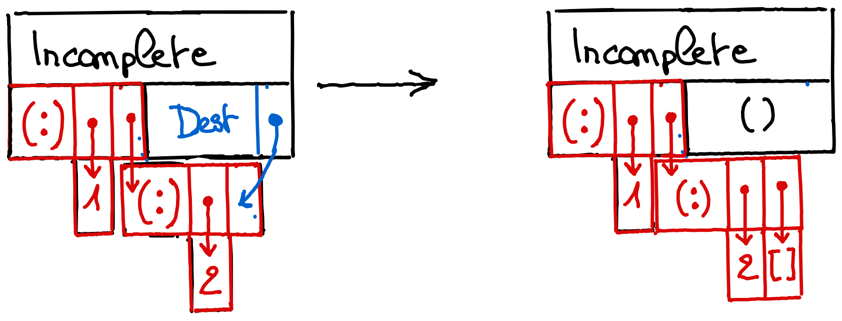
```
i1 <&> \dt1 -> fillComp i2 dt1
```



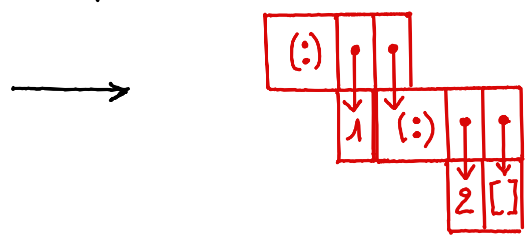
4

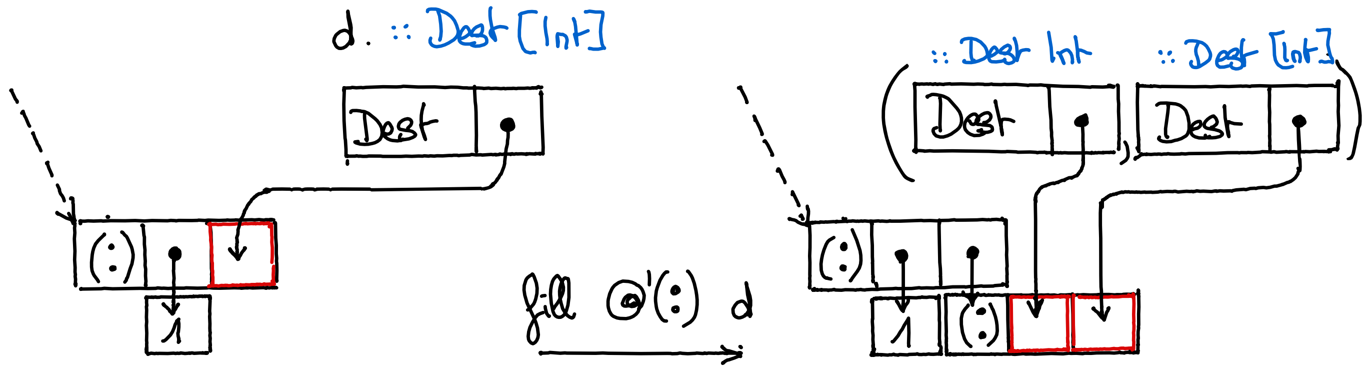
toList i3

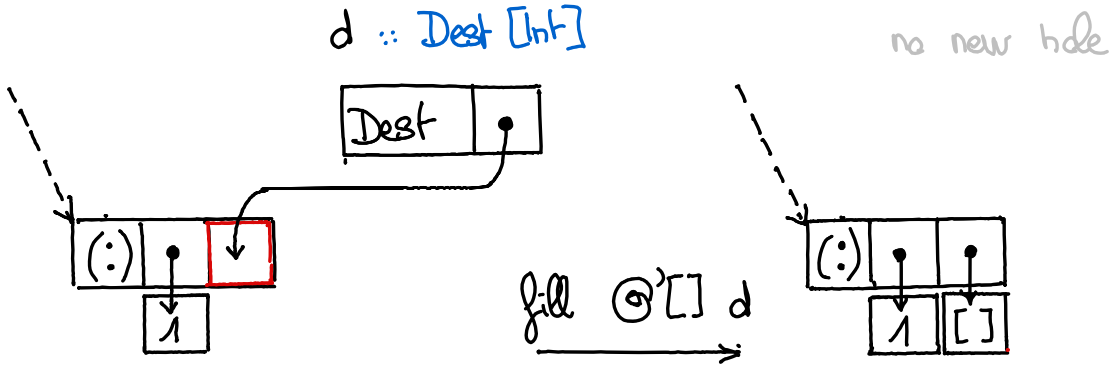
```
i3 <&> \dt2 -> fill @'[] dt2
```

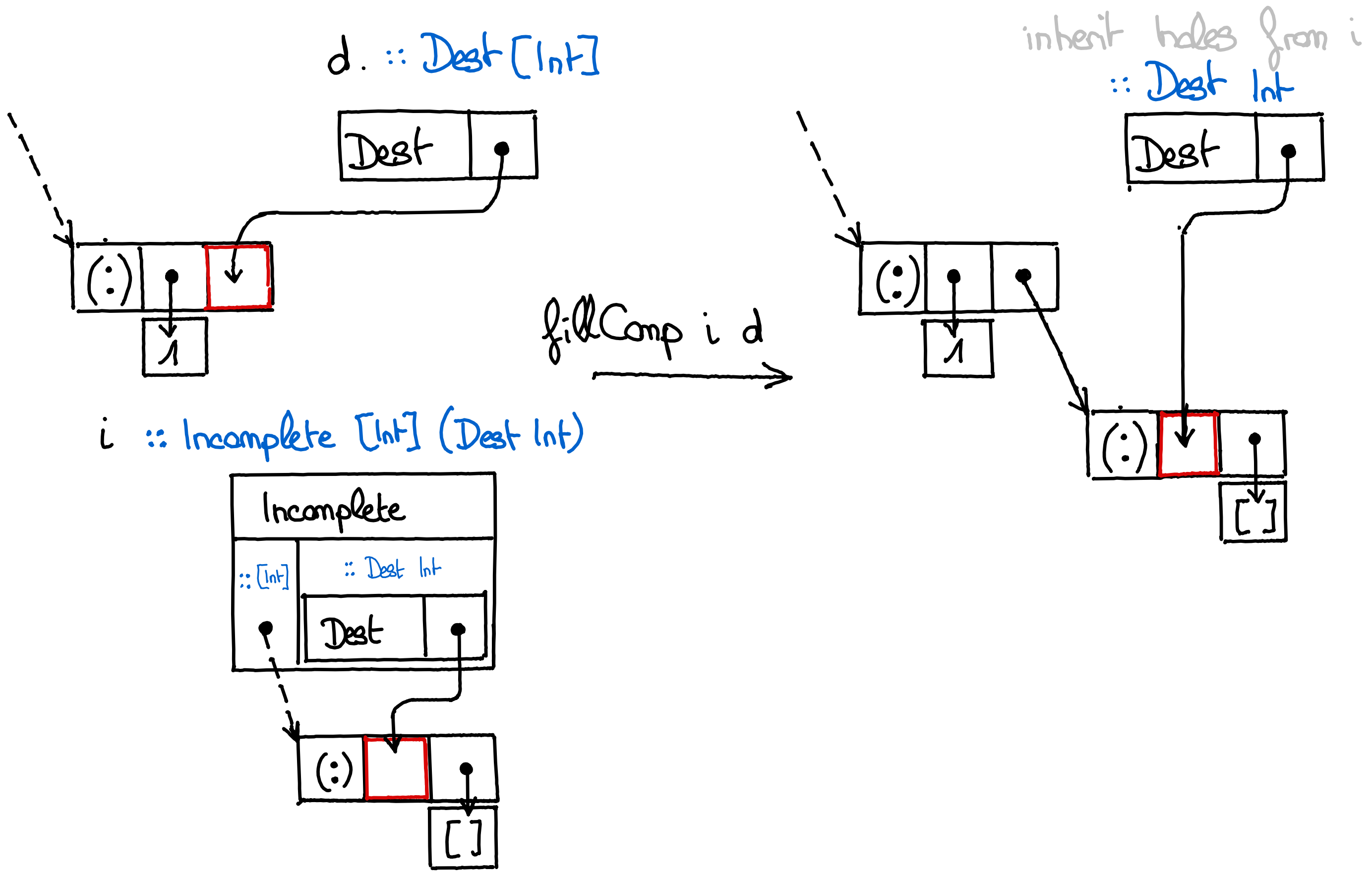


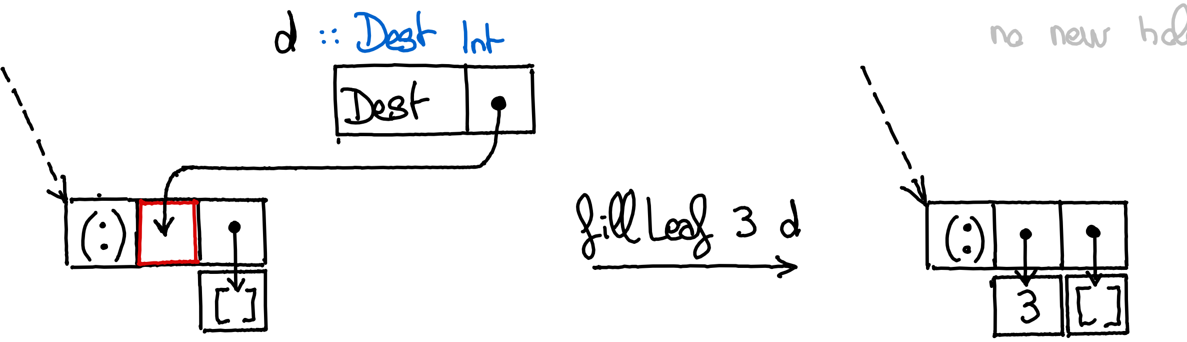
```
fromIncomplete' (___)
```

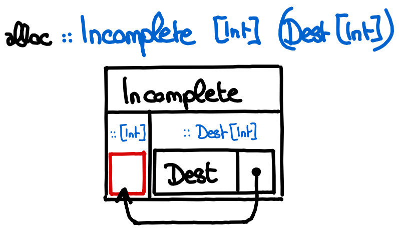


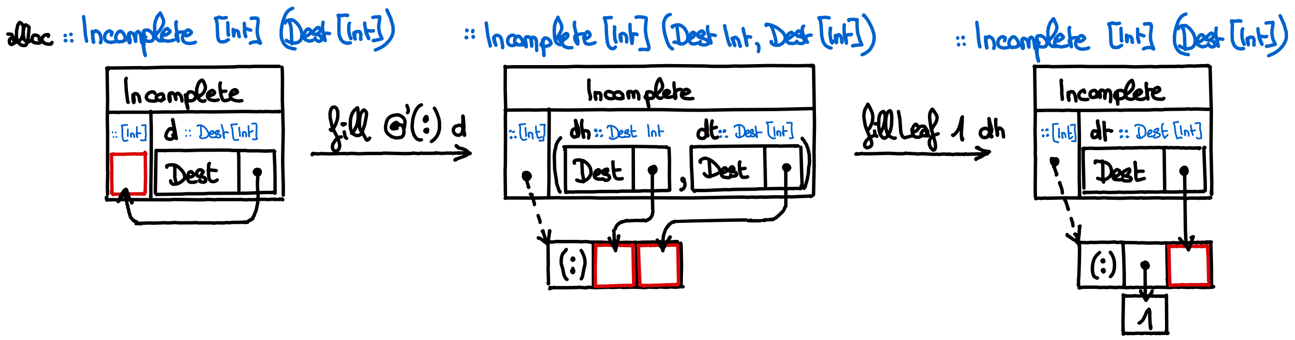

d . :: Dest [Int]
Dest
(:)
1
fill @'(:) d
:: Dest Int
:: Dest [Int]
Dest
Dest
(:)
1
(:)

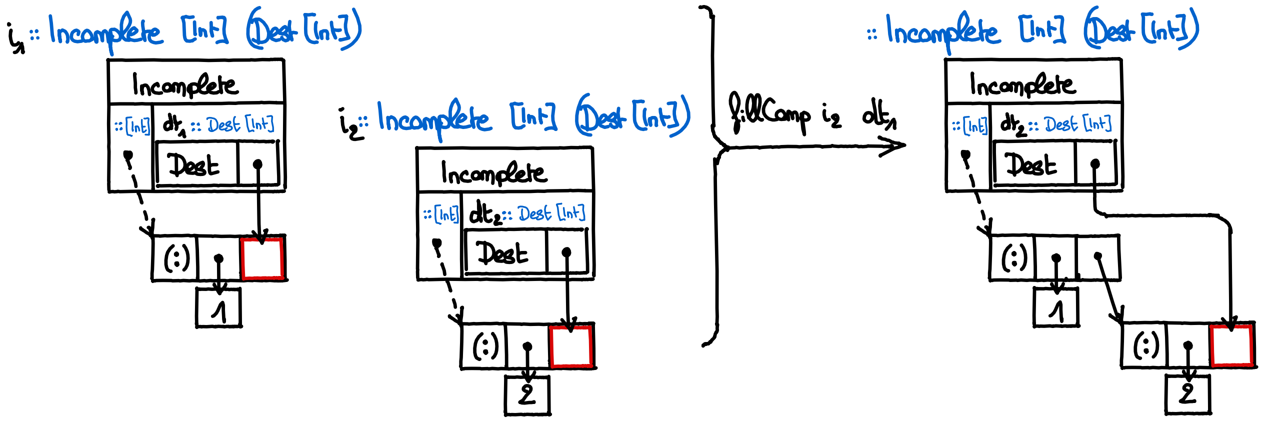

d :: Dest [Int]
Dest
(:)
1
fill @'[] d
no new hole
(:)
1
[ ]

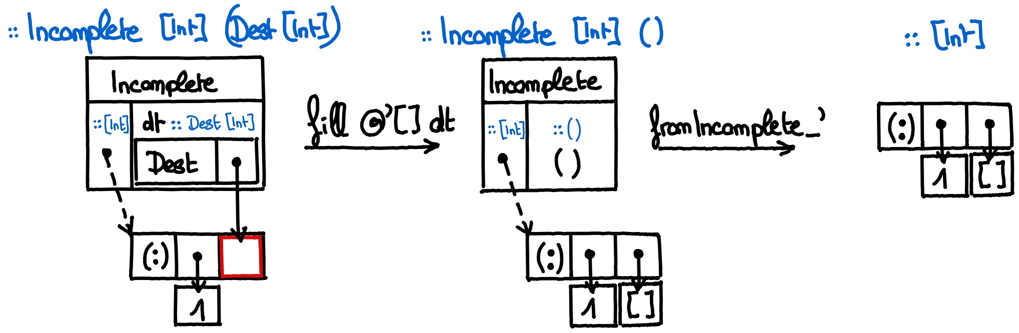

d :: Dest [Int]
Dest
(:)
1
i :: Incomplete [Int] (Dest Int)
Incomplete
:: [Int]
:: Dest Int
Dest
(:)
[ ]
fillComp i d
inherit holes from i
:: Dest Int
Dest
(:)
1
(:)
[ ]

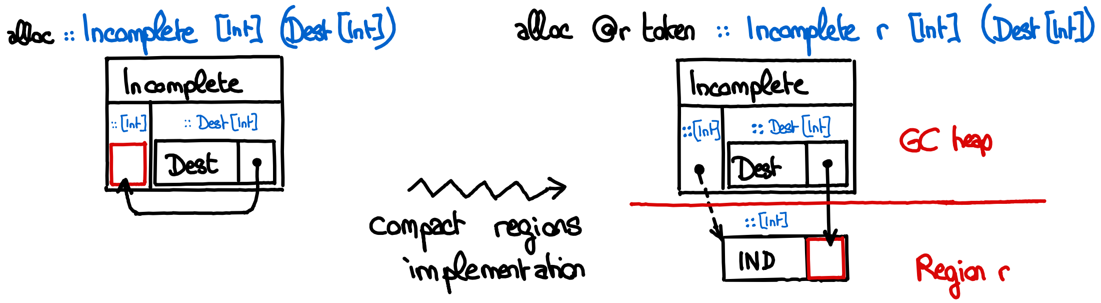

d :: Dest Int
Dest
(:)
[ ]
fillLeaf 3 d
no new ho
(:)
3
[ ]


alloc :: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
:: Dest [Int]
Dest


alloc :: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
d :: Dest [Int]
Dest
fill @'(:) d
:: Incomplete [Int] (Dest Int, Dest [Int])
Incomplete
:: [Int]
dh :: Dest Int
dt :: Dest [Int]
Dest
Dest
(:)
fillLeaf 1 dh
:: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
dt :: Dest [Int]
Dest
(:)
1


i1 :: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
dt1 :: Dest [Int]
Dest
(:)
1
i2 :: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
dt2 :: Dest [Int]
Dest
(:)
2
fillComp i2 dt1
:: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
dt2 :: Dest [Int]
Dest
(:)
1
(:)
2


:: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
dt :: Dest [Int]
Dest
(:)
1
fill @'[] dt
:: Incomplete [Int] ()
Incomplete
:: [Int]
:: ()
()
(:)
1
[]
fromIncomplete_'
:: [Int]
(:)
1
[]


alloc :: Incomplete [Int] (Dest [Int])
Incomplete
:: [Int]
:: Dest [Int]
Dest
compact regions implementation
alloc @r token :: Incomplete r [Int] (Dest [Int])
Incomplete
:: [Int]
:: Dest [Int]
Dest
GC heap
:: [Int]
IND
Region r