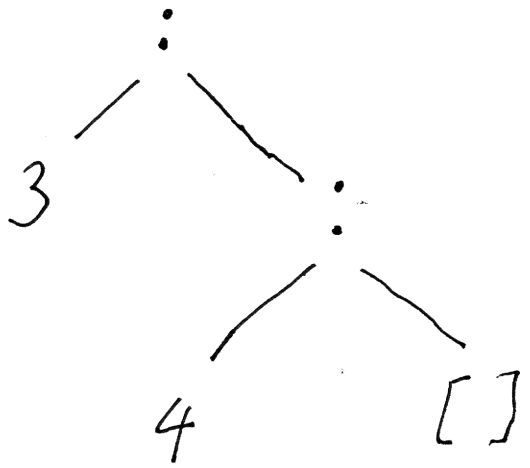
foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

f = foldr b l

f [3, 4]

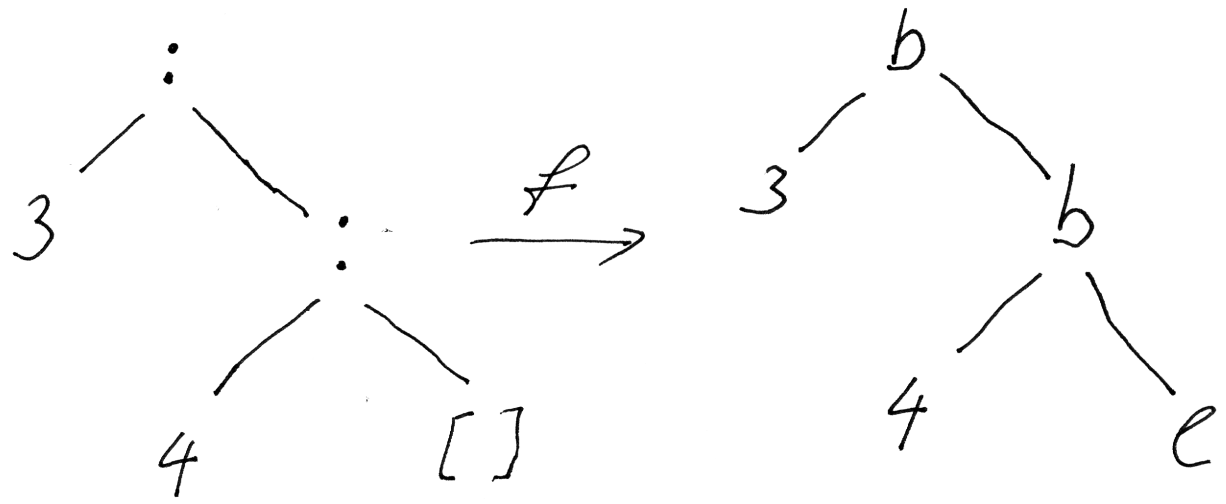


foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

f = foldr b e

f [3, 4]

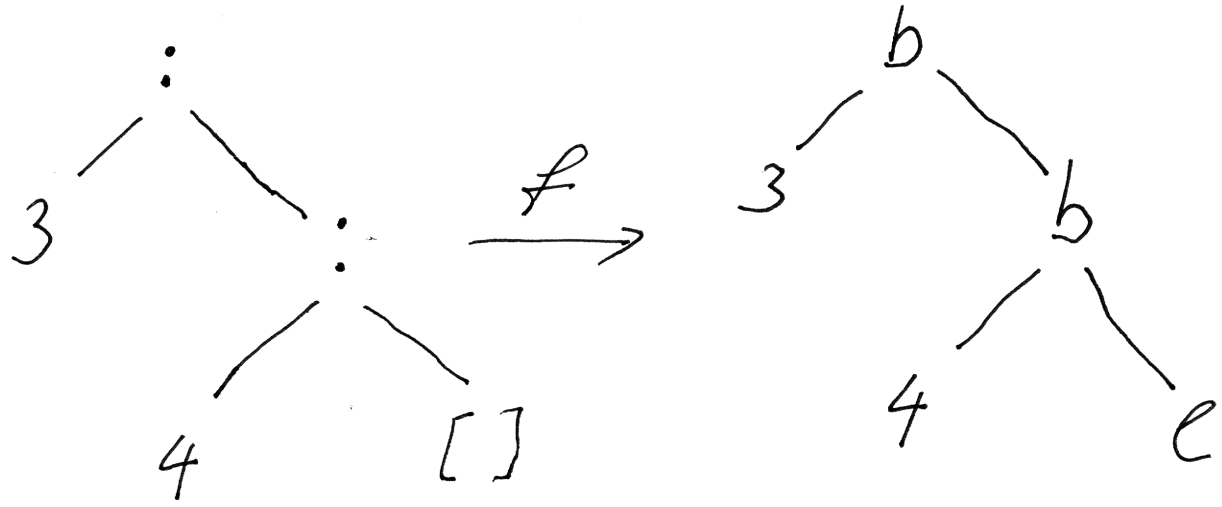


foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

f = foldr b l

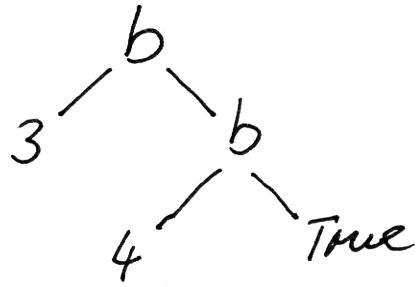
f [3, 4]




f = w_inc

l = True

b = ?

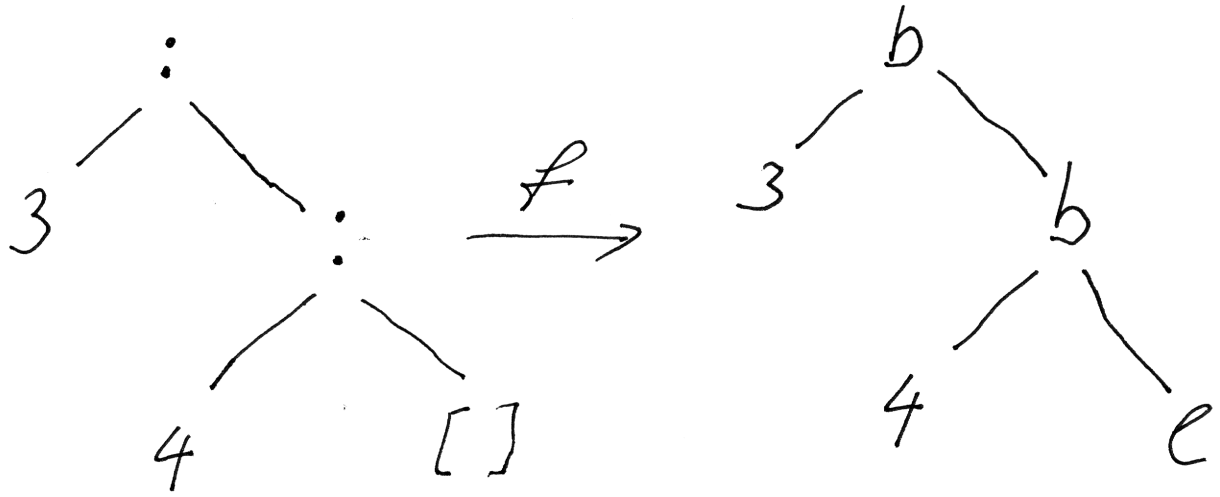


foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

f = foldr b c

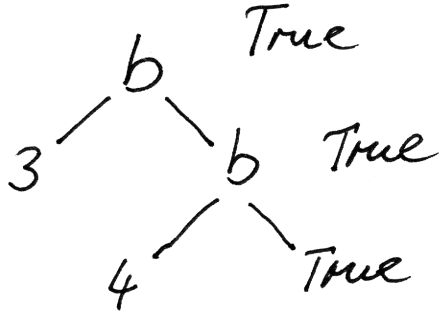
f [3, 4]




f = w_inc

c = True

b = ?



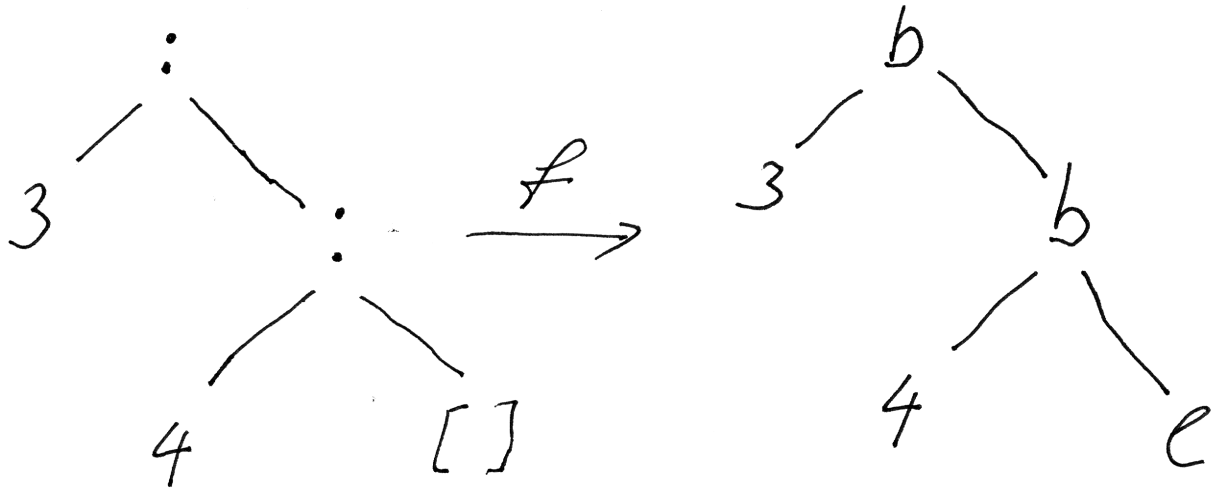

b_True = True

foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

f = foldr b c

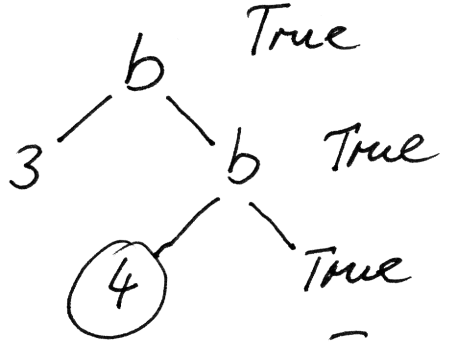
f [3, 4]




f = w_inc

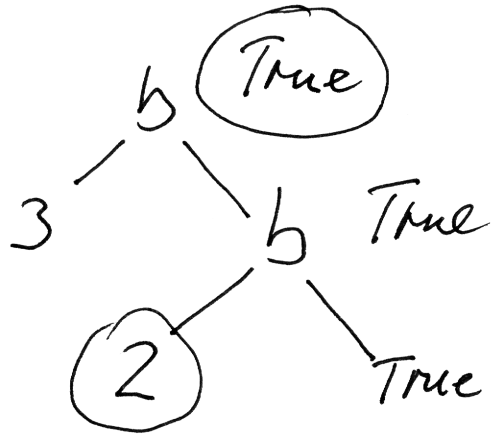
c = True

b = ?




b_True = True

aber damit auch:

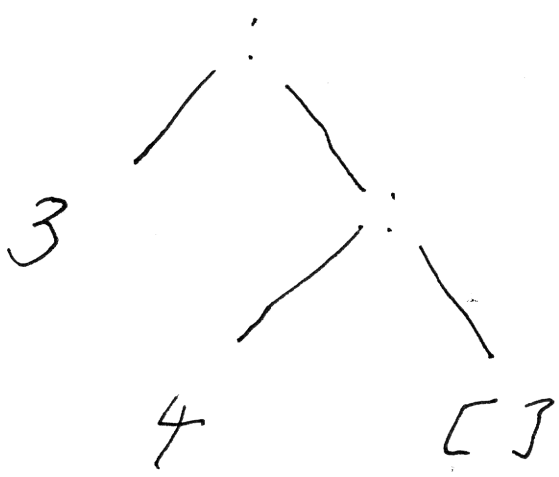


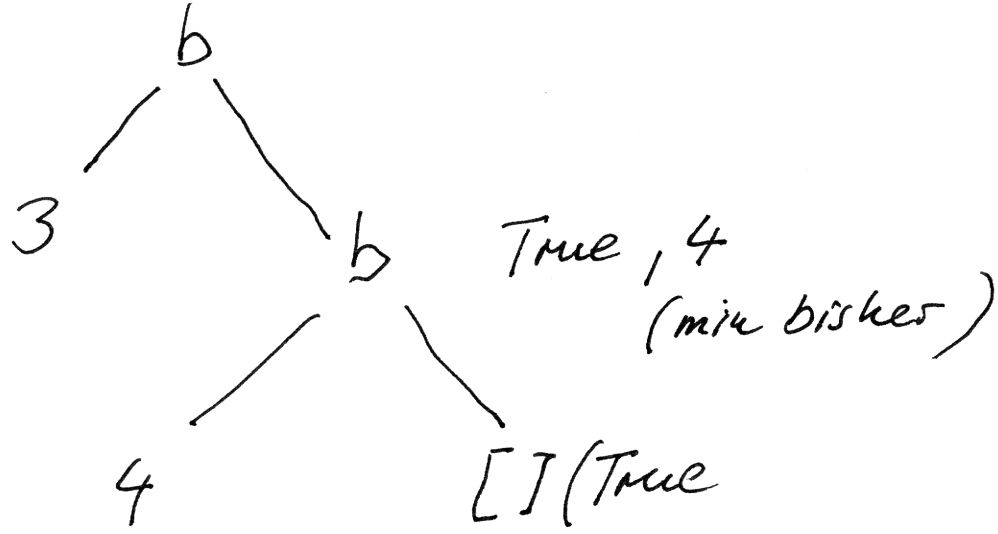
foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

für w_inc notwendig: Zusatzinformation

w_inc = foldr



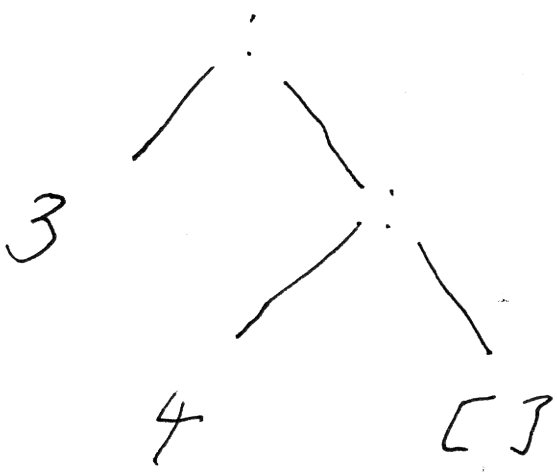



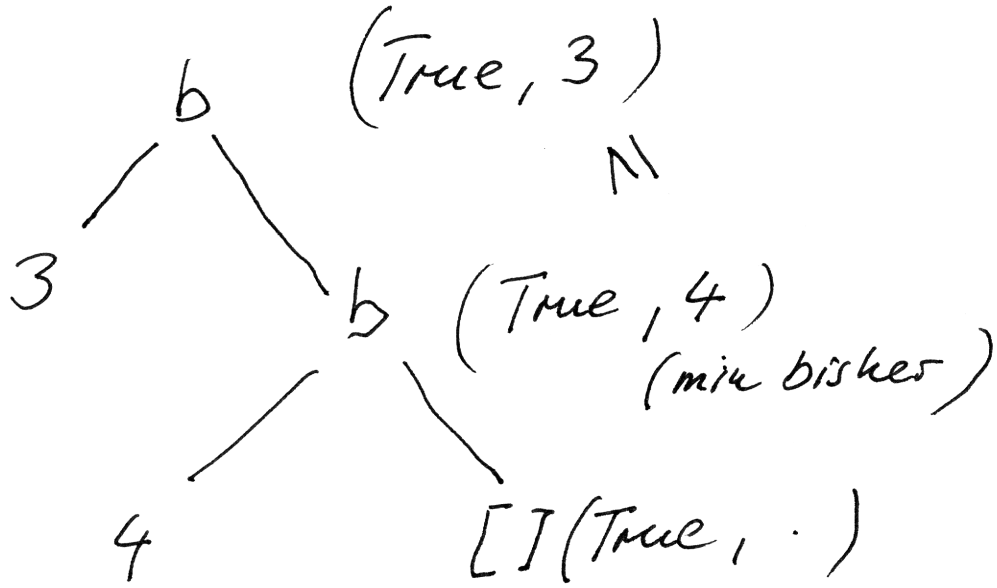
foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

für w_inc notwendig: Zusatzinformation

w_inc = foldr



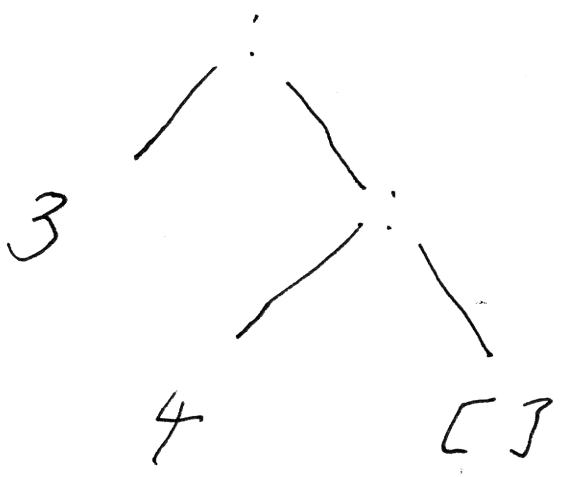



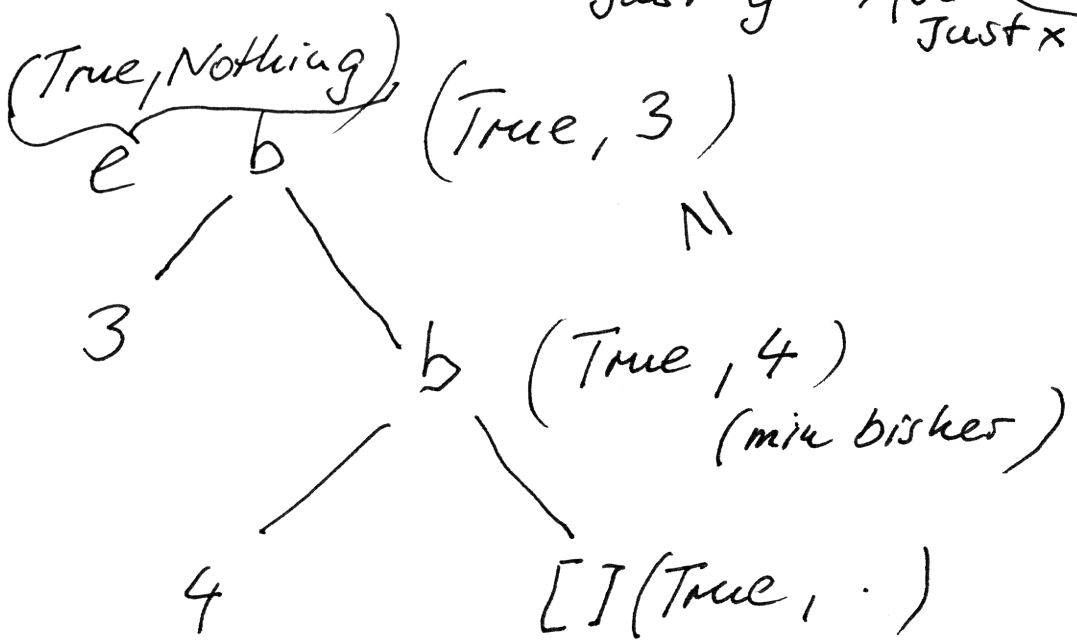
foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

für W_inc notwendig: Zusatzinformation

```
W_inc = foldr (λx (ok, m) → case m of
                Nothing → (ok, Just x)
                Just y  → (ok ∧ (x ≤ y), Just x))
```



foldr auf Listen

foldr :: $(a \to b \to b) \to b \to [a] \to b$

für w_inc notwendig: Zusatzinformation

```
w_inc = fst . foldr (λx (ok, m) → case m of
                                    Nothing → (ok, Just x)
                                    Just y  → (ok ∧ (x ≤ y), Just x)) (True, Nothing)
```

Projektion