AFP Exercises Week 1

Tom Verhoeff

1 Exercises

1. Define $nats = [0, 1, 2, \ldots]$.
2. Define *map* such that $map.f.as.i = f.(as.i)$ for $0 \leq i < length.as$.
3. Define *ftake* such that $ftake.as.n$ returns list bs such that there exists a list cs with $as = bs +\!\!+ cs$ where $length.bs \leq n$ and $cs = [\,]$ if $length.bs < n$.

In each case, try the following approaches:

- Explicit recursive definition
- Using *fix*
- Using catamorphisms (folds) or anamorphisms (unfolds)

2 Explicit recursive definitions

1.

$$nats \in \mathbb{L}.\mathbb{N} \tag{1}$$
$$nats = rec.0 \quad \text{where} \tag{2}$$
$$rec.n = n \vdash rec.(n+1) \tag{3}$$
$$rec = \vdash \circ (\mathsf{id} \times rec \circ (+1)) \tag{4}$$

```
nats :: [Int]
nats = rec 0 where
  rec n = n : rec (n + 1)
-- alternative
nats = 0 : map (+1) nats
```

2.

$$map \in (A \to B) \to \mathbb{L}.A \to \mathbb{L}.B \tag{5}$$
$$map.f = rec \quad \text{where} \tag{6}$$
$$rec.[\,] = [\,] \tag{7}$$
$$rec.\,(a \vdash as) = f.a \vdash rec.as \tag{8}$$

```
1   mymap :: (a -> b) -> [a] -> [b]
2   mymap f = rec where
3     rec []       = []
4     rec (a : as) = f a : rec as
```

3.

$$ftake \in \mathbb{L}.A \to \mathbb{N} \to \mathbb{L}.A \tag{9}$$
$$ftake = rec \quad \text{where} \tag{10}$$
$$rec.[\,] = (\lambda\, n : [\,]) \tag{11}$$
$$= [\,]^\bullet \tag{12}$$
$$rec.\,(a \vdash as) = (\lambda\, n : \text{if } n = 0 \text{ then } [\,] \text{ else } a \vdash rec.as.\,(n-1)) \tag{13}$$
$$= ([\,]^\bullet \triangledown (a \vdash) \circ rec.as \circ (-1)) \circ (= 0)? \tag{14}$$

```
1   ftake :: [a] -> Int -> [a]
2   ftake = rec where
3     rec [] _       = []
4     rec (a : as) n = if n == 0 then []
5                      else a : rec as (n - 1)
```

3 Using *fix*

Recall how *fix* is defined:

$$fix \in (A \to A) \to A \tag{15}$$
$$fix.f = f.\,(fix.f) \tag{16}$$

```
1   import Data.Function (fix)
```

1.

$$nats = \mathit{fix}.(\lambda\, x : (\lambda\, n : n \vdash x.\,(n+1))).0 \tag{17}$$
$$= \mathit{fix}.\,((\lambda\, x : \vdash \circ\, (\mathsf{id} \times x \circ (+1)))).0 \tag{18}$$
$$= \mathit{fix}.\,((\vdash \circ\,) \circ (\mathsf{id}\times) \circ (\circ\,(+1))).0 \tag{19}$$
$$\tag{20}$$
$$nats = \mathit{fix}.\,((0 \vdash\,) \circ \mathit{map}.(+1)) \tag{21}$$

```
nats = fix (\x n -> n : x (n + 1)) 0

nats = fix (ap (:) . (. (1 +))) 0

nats = fix (\x -> 0 : map (+1) x)

nats = fix ((0 :) . map (+1))
```

2.

$$\mathit{map}.f = \mathit{fix}.(\lambda\, x : ([\,]^{\bullet} \triangledown \vdash \circ\, (f \circ \mathit{head} \vartriangle x \circ \mathit{tail})) \circ (= [\,])?) \tag{22}$$
$$= \mathit{fix}.(\lambda\, x : ([\,]^{\bullet} \triangledown \vdash \circ\, (f \vartriangle x)) \circ \mathit{out}) \tag{23}$$

```
mymap f = fix (\x as -> if null as then []
                        else f (head as) : x (tail as)
              )
```

3.

$$\mathit{ftake} = \mathit{fix}.\left(\lambda\, x : \left(\lambda\, \mathit{as} : \left(\lambda\, n : \begin{array}{l}\mathsf{if}\ \ \mathit{as} = [\,] \vee n = 0\ \mathsf{then}\ [\,] \\ \quad\ \mathsf{else}\ \mathit{head}.\mathit{as} \vdash x.\,(\mathit{tail}.\mathit{as})\,.\,(n-1)\end{array}\right)\right)\right) \tag{24}$$

```
ftake = fix (\x as n -> if null as || n == 0 then []
                        else head as : x (tail as))
```

4 Using folds and unfolds

For $\mathbb{L}.A$ we have

$$\mathbb{L}.A \cong F.(\mathbb{L}.A) \quad \text{where} \tag{25}$$
$$F.X = \mathbb{1} + A \times X \tag{26}$$
$$F.f = \mathsf{id} + \mathsf{id} \times f \tag{27}$$
$$in = [\,]^{\bullet} \triangledown \vdash \tag{28}$$
$$out = (\_^{\bullet} + head \vartriangle tail) \circ (= [\,])? \tag{29}$$
$$(\!|\, \rho \,|\!) = \rho \circ F.(\!|\, \rho \,|\!) \circ out \tag{30}$$
$$= fix(\lambda\, x : \rho \circ F.x \circ out) \tag{31}$$
$$[\!|\, \sigma \,|\!] = in \circ F.[\!|\, \sigma \,|\!] \circ \sigma \tag{32}$$
$$= fix(\lambda\, x : in \circ F.x \circ \sigma) \tag{33}$$
$$giterate.h.f = [\!|\, \hookleftarrow \circ (h \vartriangle f) \,|\!] \tag{34}$$
$$iterate = giterate.\mathsf{id} \tag{35}$$
$$\tag{36}$$

```
foldr :: (a -> b -> b) -> b -> [a] -> b

import Data.List (unfoldr)

unfoldr :: (b -> Maybe (a, b)) -> b -> [a]

iterate :: (a -> a) -> a -> [a]
```

1.

$$nats = [\!|\, \hookleftarrow \circ (\mathsf{id} \vartriangle (+1)) \,|\!].0 \tag{37}$$
$$= iterate.(+1).0 \tag{38}$$

```
nats = unfoldr (\n -> Just (n, n + 1)) 0

nats = iterate (+1) 0
```

2.

$$map.f = (\!|\, in \circ (\mathsf{id} + f \times \mathsf{id}) \,|\!) \tag{39}$$
$$= (\!|\, [\,]^{\bullet} \triangledown \vdash \circ (f \times \mathsf{id}) \,|\!) \tag{40}$$
$$\tag{41}$$
$$map.f = [\!|\, (\_^{\bullet} + f \vartriangle \mathsf{id}) \circ (= [\,])? \,|\!] \tag{42}$$

```
mymap f = foldr (\a mfas -> f a : mfas) []

mymap f = unfoldr (\xs -> if null xs then Nothing
                          else Just (f (head xs), tail xs))
```

3.

$$ftake = (\!|\, []^\bullet \triangledown \oplus \,|\!) \quad \text{where} \tag{43}$$

$$a \oplus ftas = (\lambda n : \text{if } n = 0 \text{ then } [] \text{ else } a \vdash ftas.(n-1)) \tag{44}$$

$$= ([]^\bullet \triangledown \vdash \circ (a^\bullet \vartriangle ftas \circ (-1))) \circ (=0)? \tag{45}$$

N.B. $uncurry.ftake = uftake$ can be defined as a list *ana*morphism:

$$uftake \in \mathbb{L}.A \times \mathbb{N} \to \mathbb{L}.A \tag{46}$$

$$uftake.(xs, n) = \text{if } xs = [] \vee n = 0 \text{ then } [] \text{ else } head.xs \vdash uftake.(tail.xs, n-1 \tag{47}$$

$$uftake = [\!(\, \_^\bullet + head \circ \ll \vartriangle (tail \times (-1))) \circ (\vee \circ ((=[]) \times (=0)))?\,]\!] \tag{48}$$

```
ftake = foldr (\a tfas n ->
                 if n == 0 then []
                 else a : ftas (n - 1)
               )
               (const [])

uftake = unfoldr (\ (xs, n) -> if xs == [] || n == 0 then Nothing
                               else Just (head xs, (tail xs, n - 1)))
```

5 Things to try

- $ftake.nats.5$
- $map.(+1).(ftake.nats.5)$
- $ftake.(map.(+1).nats).5$