Announcements

"A computer is like a violin. You can imagine a novice trying first a phonograph and then a violin. The latter, he says, sounds terrible. That is the argument we have heard from our humanists and most of our computer scientists. Computer programs are good, they say, for particular purposes, but they aren't flexible. Neither is a violin, or a typewriter, until you learn how to use it."

– *Marvin Minsky, "Why Programming Is a Good Medium for Expressing Poorly-Understood and Sloppily-Formulated Ideas", 1967*

List Definitions (foldr and friends) Lists3.hs

Define:

- $sum\ [a1, a2, ..an] = a1 + a2 + ... + an + 0$
- $product\ [a1, a2, ..an] = a1 * a2 * ... * an * 1$
- $or\ [a1, a2, ..an] = a1 || a2 || ... || an\ || False$
- $append\ [a1, a2, ..an]\ b = a1 : a2 : ... : an : b$
- generalized to
  $foldr\ \oplus\ v\ [a1, a2, ..an] = a1 \oplus (a2 \oplus (... \oplus (an \oplus v)))$

-
  ```
  foldr f v []     = v
  foldr f v (x:xs) = f x (foldr f v xs)
  ```

Clicker Question

myfoldr is defined by

```
-- myfoldr op v [a1,a2,..an]
--        = a1 op (a2 op (... op (an  op v)))
myfoldr f v []  = v
myfoldr f v (x:xs) = f x (myfoldr f v xs)
```

What is the type of `myfoldr (&&)`
(Recall that `&& :: Bool -> Bool` is logical "and")

- A `myfoldr (&&) :: [Bool] -> Bool`
- B `myfoldr (&&) :: [Bool] -> [Bool] -> Bool`
- C `myfoldr (&&) :: Bool -> [Bool] -> Bool`
- D `myfoldr (&&) :: Bool -> Bool`
- E `myfoldr (&&) :: Bool -> [Bool] -> [Bool]`

Clicker Question

myfoldr is defined by

```
-- myfoldr op v [a1,a2,..an]
--        = a1 op (a2 op (... op (an  op v)))
myfoldr f v []  = v
myfoldr f v (x:xs) = f x (myfoldr f v xs)
```

What is the value of

```
myfoldr (\ x y -> 10*x : y) [] [1,2,3,4]
```

- A `100`
- B `[40,30,20,10]`
- C `[1,2,3,4]`
- D `[10,20,30,40]`
- E `4321`

Clicker Question

myfoldr is defined by

```
-- myfoldr op v [a1,a2,..an]
--        = a1 op (a2 op (... op (an  op v)))
myfoldr f v []  = v
myfoldr f v (x:xs) = f x (myfoldr f v xs)
```

What is the value of

```
myfoldr (\ x y -> 10*x + y) 0 [1,2,3,4]
```

- A `100`
- B `[4,3,2,1]`
- C `[10,20,30,40]`
- D `1234`
- E `4321`

Clicker Question

myfoldr is defined by

```
-- myfoldr op v [a1,a2,..an]
--        = a1 op (a2 op (... op (an  op v)))
myfoldr f v []  = v
myfoldr f v (x:xs) = f x (myfoldr f v xs)
```

What is the value of

```
myfoldr (\ x y -> x + 10*y) 0 [1,2,3,4]
```

A 20

B [4,3,2,1]

C [1,2,3,4]

D 1234

E 4321

List Definitions (foldl and friends) Lists4.hs

- Define sum using accumulators (tail recursion).
- Define $rev2\ lst1\ lst2 =$ reverse of list lst2 followed by lst1
- Define $rev\ \ lst\ =$ the reverse of lst
- $foldl\ \ \oplus\ \ v\ [a1, a2, ..an] = (((v \oplus a1) \oplus a2) \oplus ...) \oplus an$
- How can *sum* be defined using *foldl*?
- How can *product* be defined using *foldl*?
- How can reverse be defined in terms of *foldl*?
- How can we define str2int that creates an integer from a string?