Example Lazy Reductions in Haskell

Given Definitions (Equations)

Note: the simplify the example reductions, the definition of the index operator has been simplified compared to the standard version.

```
ones = 1:ones                       -- the infinite list [1,1,1...

nats = natsFrom 0                   -- the infinite list [0,1,2,...

natsFrom n = n:(natsFrom (n+1))     -- note: outer parens not requ'd

(x:xs) !! 0 = x                     -- (!!).1
(x:xs) !! n = xs !! (n-1)           -- (!!).2
```

Example 1

Lazy Strategy: prioritize the leftmost outermost step that is possible (where the subexpression matches an equation LHS)

```
        ones !! 2

=       (1:ones) !! 2                by ones

=       ones !! (2-1)                by (!!).2 with [x=1,xs=ones, n=2]

=       (1:ones) !! (2-1)            by ones

=       (1:ones) !! 1                by (-)

=       ones !! (1-1)                by (!!).2 with [x=1,xs=ones, n=1]

=       (1:ones) !! (1-1)            by ones

=       (1:ones) !! 0                by (-)

=       1                            by (!!).1 with [x=1,xs=ones]
```

Example 2

Note: at some steps, sharing results in a smaller expression at some steps than is shown here (it’s an expression graph, not an expression tree). For example, the “0+1” that appears at multiple places in the same expression starting at line 5 is actually represented internally by a single structure.

```
       nats !! 2
=      (natsFrom 0) !! 2                              by nats
=      (0:(natsFrom (0+1))) !! 2                      by natsFrom with [n=0]
=      (natsFrom (0+1)) !! (2-1)                      by (!!).2 with [x=0,xs=natsFrom (0+1),n=2]
=      ((0+1):(natsFrom ((0+1)+1))) !! (2-1)          by natsFrom with [n=0+1]
=      ((0+1):(natsFrom ((0+1)+1))) !! 1              by (-)
=      (natsFrom ((0+1)+1)) !! (1-1)                  by (!!).2 with [x=0+1,xs=natsFrom ((0+1)+1),n=1]
=      (((0+1)+1):(natsFrom (((0+1)+1)+1))) !! (1-1)  by natsFrom with [n=(0+1)+1]
=      (((0+1)+1):(natsFrom (((0+1)+1)+1))) !! 0      by (-)
=      (0+1)+1                                        by (!!).1 with [x=(0+1)+1,xs=natsFrom (((0+1)+1)+1)]
=      1+1                                            by (+)
=      2                                              by (+)
```