Emulating Primality with Multiset Representations of Natural Numbers

Paul Tarau[1]

[1] Department of Computer Science and Engineering
Univ of North Texas

ICTAC 2011

Motivation

- analogies (and analogies between analogies) emerge when we transport objects and operations on them
- this is a creative process - one of the most rewarding ones in terms of interesting outcomes (geometry and coordinates, Turing machines and combinators, primes and complex functions, etc.)
- Paul Erdös: It will be another million years at least, before we understand the primes $\rightarrow$ difficult open problems - e.g. the Riemann Hypothesis - unexpected connections to various fields
- to be able to encode something as something else we need isomorphisms $\rightarrow$ bijections that transport structures
- $\rightarrow$ the paper is about emulating some interesting properties of primes - using a more regular "factoring" of natural numbers

Outline

- the groupoid of data type isomorphisms
- connection between multisets, primes and Gödel's encodings
- a simple and efficient encoding of natural numbers as multisets
- the analogy between multiset decompositions and factoring - generic operations on the related monoids
- experiments with the Möbius, Mertens and "rad" functions, some interesting automorphisms of $\mathbb{N}$
- conclusion

the paper is a literate Haskell program - self contained code at `http://logic.cse.unt.edu/tarau/research/2011/mprimes.hs`

The Groupoid of Isomorphisms

```
data Iso a b = Iso (a→b) (b→a)
from (Iso f _) = f
to (Iso _ g) = g

compose :: Iso a b → Iso b c → Iso a c
compose (Iso f g) (Iso f′ g′) = Iso (f′ . f) (g . g′)

itself = Iso id id
invert (Iso f g) = Iso g f
```

Proposition

`Iso` *is a groupoid: when defined,* `compose` *is associative,* `itself` *is an identity element,* `invert` *computes the inverse of an isomorphism.*

Transporting Operations

```
borrow_from :: Encoder a → (a → a → a) →
               Encoder b → (b → b → b)
borrow_from lender op borrower x y = as borrower lender
    (op (as lender borrower x) (as lender borrower y))
```

Choosing a Hub

```
type N = Integer
type Hub = [N]
```

We can now define an *Encoder* as an isomorphism connecting an object to *Hub*

```
type Encoder a = Iso a Hub
```

the combinators *with* and *as* provide an *embedded transformation language* for routing isomorphisms through two *Encoders*:

```
with :: Encoder a→Encoder b→Iso a b
with this that = compose this (invert that)

as :: Encoder a → Encoder b → b → a
as that this = to (with that this)
```

A bijection between lists and sets of natural numbers

```
set2list xs = shift_tail pred (mset2list xs) where
  shift_tail _ [] = []
  shift_tail f (x:xs) = x:(map f xs)

list2set = (map pred) . list2mset . (map succ)

set :: Encoder [N]
set = Iso set2list list2set
```

Examples

```
*MPrimes> as set list [0,1,0,0,4]
[0,2,3,4,9]
*MPrimes> as list set [0,2,3,4,9]
[0,1,0,0,4]
```

How we do it?

```
[0, 1,0,0,4]->[0, 2,1,1,5]->[0, 2,3,4,9]
    next slide: 541=2^0+2^2+2^3+2^4+2^9
```

we map lists of natural numbers to strictly increasing sequences of natural numbers representing sets

Ackerman's bijection between $\mathbb{N}$ and sets of elements of $\mathbb{N}$

```
nat_set = Iso nat2set set2nat

nat2set n | n≥0 = nat2exps n 0 where
  nat2exps 0 _ = []
  nat2exps n x = if (even n) then xs else (x:xs) where
    xs=nat2exps (n `div` 2) (succ x)

set2nat ns = sum (map (2^) ns)
```

The resulting Encoder is:

```
nat :: Encoder N
nat = compose nat_set set
```

Examples illustrating Ackermann's bijection

We can fold a set, represented as a list of distinct natural numbers into a single natural number, reversibly, by observing that it can be seen as the list of exponents of 2 in the number's base 2 representation.

```
*MPrimes> as nat set [0, 2,3,4,9]
541
*MPrimes> as nat list [0, 1,0,0,4]
541
*MPrimes> as set nat 42
[1,3,5]
*MPrimes> borrow_from nat (+) set [1,2,9] [2,5,6,8]
[1,3,5,6,8,9]
```

Multisets and Primes

- multisets are unordered collections with repeated elements
- non-decreasing sequences provide a canonical representation for multisets of natural numbers
- a natural number as a product of primes $\rightarrow$ a multiset
- prime numbers exhibit a number of fundamental properties of natural phenomena and human artifacts in an unusually pure form (e.g "reversibility" is present as the ability to recover the operands of a product of distinct primes)
- the question we would like to explore: can alternative, computationally simpler multiset decompositions of natural numbers emulate some properties of prime numbers?

Factoring as a multiset representation of a natural number

```
nat2pmset 1 = []
nat2pmset n = to_prime_positions n
```

Proposition

p is prime if and only if its decomposition in a multiset given by `nat2pmset` *is a singleton*

a function `pmset2nat` maps back a multiset of positions of primes to the result of the product of the corresponding primes

```
pmset2nat [] = 1
pmset2nat ns = product (map (from_pos_in primes . pred) ns)
```

The Encoder **pmset**

```
pmset :: Encoder [N]
pmset = compose (Iso pmset2nat nat2pmset) nat
```

working as follows:

```
*MPrimes> as pmset nat 2010
[1,2,3,19]
*MPrimes> as nat pmset [1,2,3,19]
2010
```

As the factoring of 2010 is $2 * 3 * 5 * 67$, the list [1,2,3,19] contains the positions of the factors, starting from 1, in the sequence of primes.

An alternative bijection between finite multisets and $\mathbb{N}$

- a multiset like $[4,4,1,3,3,3]$ could be represented canonically as sequence by first ordering it as $[1,3,3,3,4,4]$
- computing the differences between consecutive elements i.e. $[x_0, x_1 \ldots x_i, x_{i+1} \ldots] \rightarrow [x_0, x_1 - x_0, \ldots x_{i+1} - x_i \ldots]$ gives $[1,2,0,0,1,0]$
- $\rightarrow$ the first element 1 followed by the increments $[2,0,0,1,0]$ maps multisets to finite lists of $\mathbb{N}$ $\rightarrow$ which are in bijection with $\mathbb{N}$

The Encoder **mset**

We will need one small change to convert this into a mapping on $\mathbb{N}^+$.

```
nat2mset1 n = map succ (as mset0 nat (pred n))
mset2nat1 ns = succ (as nat mset0 (map pred ns))

mset :: Encoder [N]
mset = compose (Iso mset2nat1 nat2mset1) nat
```

The resulting mapping, like `pmset`, now works on $\mathbb{N}^+$.

```
*MPrimes> as mset nat 2012
[1,1,2,2,3,3,3,3,3]
*MPrimes> as nat mset it
2012
*MPrimes> map (as mset nat) [1..7]
[[],[1],[2],[1,1],[3],[1,2],[2,2]]
```

A multiset analog to multiplication

```
mprod = borrow_from mset sortedConcat nat
```

Proposition

$\langle N^+, mprod, 1\rangle$ is a commutative monoid i.e. `mprod` *is defined for all pairs of natural numbers and it is associative, commutative and has 1 as an identity element.*

Proof.

rewrite the definition of `mprod` as the equivalent:

```
mprod_alt n m = as nat mset
  (sortedConcat (as mset nat n) (as mset nat m))
```

follows from the associativity of the concatenation operation □ □

Proprieties of **mprod**: examples

`mprod` has properties similar to ordinary multiplication:

```
*MPrimes> mprod 41 (mprod 33 38) == mprod (mprod 41 33) 38
True
*MPrimes> mprod 33 46 == mprod 46 33
True
*MPrimes> mprod 1 712 == 712
True
```

Similar definition - **mprod** - same as *

```
mprod = borrow_from mset sortedConcat nat
```

Multiset analogues for div, gcd and lcd: definitions

```
mgcd :: N → N → N
mgcd = borrow_from mset msetInter nat

mlcm :: N → N → N
mlcm = borrow_from mset msetUnion nat

mdivisible :: N→N→Bool
mdivisible n m = mgcd n m==m

mexactdiv :: N → N → N
mexactdiv n m | mdivisible n m = mdiv n m
```

Multiset analogues for div, gcd and lcd: properties

$$mprod\,(mgcd\ x\ y)\,(mlcm\ x\ y) \equiv mprod\ x\ y \tag{1}$$

$$mexactdiv\,(mprod\ x\ y)\ y \equiv x \tag{2}$$

$$mexactdiv\,(mprod\ x\ y)\ x \equiv y \tag{3}$$

Multiset primes

Definition

We say that $p > 1$ is a multiset-prime (or **mprime***), if its decomposition as a multiset is a singleton.*

The following holds

Proposition

$p > 1$ is a multiset prime if and only if it is not mdivisible by any number in $[2..p-1]$.

Proof.

By observing that singleton multisets are the first to contain a given number as the multiset [a,b] corresponds to a number strictly larger than the numbers corresponding to multisets [a] and [b]. $\square$ $\square$

There's an infinite number of multiset primes

```
*MPrimes> take 10 mprimes
[2,3,5,9,17,33,65,129,257,513]
```

suggesting the following proposition:

Proposition

There's an infinite number of multiset primes *and they are exactly the numbers of the form* $2^n + 1$*.*

Proof.

The proof follows immediately by observing that the first value of `as mset nat n` that contains k, is $n = 2^k + 1$ and that numbers of that form are exactly the numbers resulting in singleton multisets. □ □

Examples

```
*MPrimes> map (as mset nat) [1..9]
[[],[1],[2],[1,1],[3],[1,2],[2,2],[1,1,1],[4]]
         ^^^        ^^^                        ^^^
         2+1        4+1                        8+1
```

→ faster versions of `mprimes` and `is_mprime`:

```
mprimes' = map (λx→2^x+1) [0..]

is_mprime' p | p>1 = p==
  last (takeWhile (λx→x≤p) mprimes')
```

An analog to the "rad" function

Definition

n is square-free if each prime on its list of factors occurs exactly once

The `rad(n)` function (`A007947` at EOIS) is defined as follows:

Definition

rad(n) is the largest square-free number that divides n

can be computed by factoring and trimming multiple occurrences

```
rad n = product (nub (to_primes n))
```

“rad” for primes and mprimes

```
prad n =  as nat pmset  (pfactors n)

mrad n =  as nat mset  (mfactors n)

*MPrimes> map rad [2..16]
[2,3,2,5,6,7,2,3,10,11,6,13,14,15,2]
*MPrimes> map prad [2..16]
[2,3,2,5,6,7,2,3,10,11,6,13,14,15,2]
*MPrimes> map mrad [2..16]
[2,3,2,5,6,3,2,9,10,11,6,5,6,3,2]
```

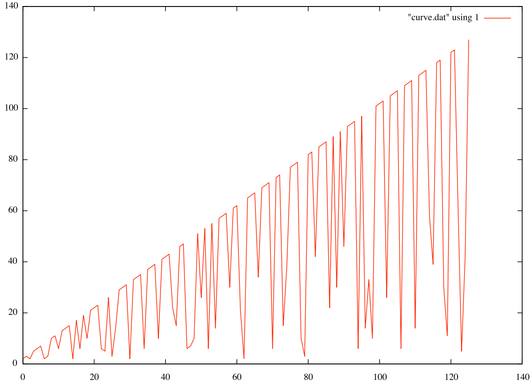

Figure: rad(n) on $[2..2^7 - 1]$

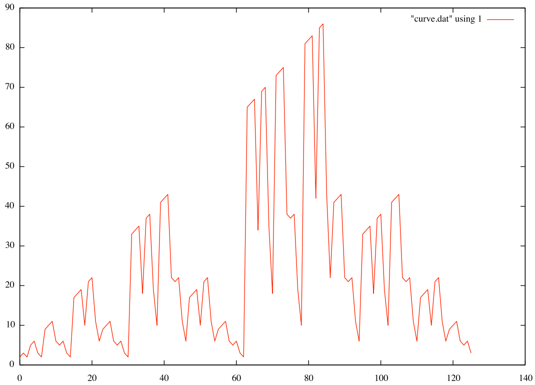

Figure: mrad(n) on $[2..2^7 - 1]$

Emulating the Möbius function

the Möbius function

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } p^2 \textit{ divides } n \text{ for some prime } p \\ (-1)^r & \text{if } n \text{ has } r \text{ distinct prime factors} \end{cases}$$

we parameterize it by the type `t` of a multiset encoding

```
mobius t n = if nub ns == ns then f ns else 0 where
  ns = as t nat n
  f ns = if even (genericLength ns) then 1 else -1
```

- `t=pmset` → *primes* (sequence A008683 in EOIS)
- `t=mset` → *mprimes* (sequence A132971 in EOIS)

```
*MPrimes> map (mobius pmset) [1..16]
[1,-1,-1,0,-1,1,-1,0,0,1,-1,0,-1,1,1,0]
*MPrimes> map (mobius mset) [1..16]
[1,-1,-1,0,-1,1,0,0,-1,1,1,0,0,0,0,0]
```

An analogue of the Mertens function

generalization of the Mertens function (A002321 in EOIS)

$$M(x) = \sum_{n \leq x} \mu(n)$$

that accumulates values of the Möbius function up to n:

```
mertens t n = sum (map (mobius t) [1..n])
```

```
*MPrimes> map (mertens pmset) [1..16]
[1,0,-1,-1,-2,-1,-2,-2,-2,-1,-2,-2,-3,-2,-1,-1]
```

```
*MPrimes> map (mertens mset) [1..16]
[1,0,-1,-1,-2,-1,-1,-1,-2,-1,0,0,0,0,0,0]
```

the Mertens conjecture (disproved by Odlyzko and te Riele)

$$|M(n)| < \sqrt{n}, \ for\ n > 1$$

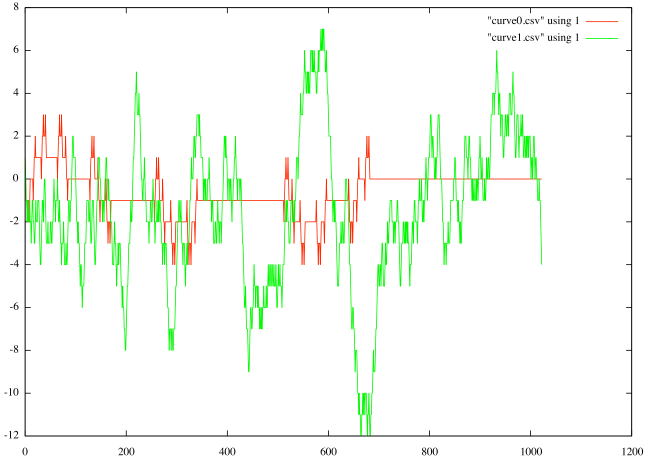

Figure: Mertens functions for mset and pmset

Exploring the Riemann Hypothesis

A connection between the Riemann Hypothesis, originating from a representation of the inverse of the Riemann ζ function as

$$\frac{1}{\zeta(s)} = \sum_{n=1}^{\infty} \frac{\mu(n)}{n^s}$$

has lead to an equivalent elementary formulation (attributed to Littlewood) of the Riemann Hypothesis

$$M(x) = O(x^{1/2+\varepsilon}) \ \forall \varepsilon > 0 \tag{4}$$

An emulation of an equivalent of the Riemann Hypothesis for multiset primes

By instantiating the previous statement to a Mertens function parameterized by a simple multiset representation like `mset` one obtains an analogue of the Riemann Hypothesis in much simpler and possibly more tractable context. A possibly interesting **a conjecture**:

> *The inequality* **??** *holds for the the instance of $M(x)$ derived from* `mset` *i.e. computed by the function* `mertens mset`*.*

This leads to speculating that, for instance, connecting values of ε between the emulation (derived from `mset`) and the original Martens function (derived from `pmset`) could provide interesting insight on the Riemann Hypothesis as such.

Deriving automorphisms of $\mathbb{N}$

Definition

an automorphism *is an isomorphism for which the source and target are the same*

```
auto_m2p 0 = 0
auto_m2p n = as nat pmset (as mset nat n)

auto_p2m 0 = 0
auto_p2m n = as nat mset (as pmset nat n)

*MPrimes> map auto_m2p [0..31]
[0,1,2,3,4,5,6,9,8,7,10,15,12,25,18,27,16,11,14,
 21,20,35,30,45,24,49,50,75,36,125,54,81]
```

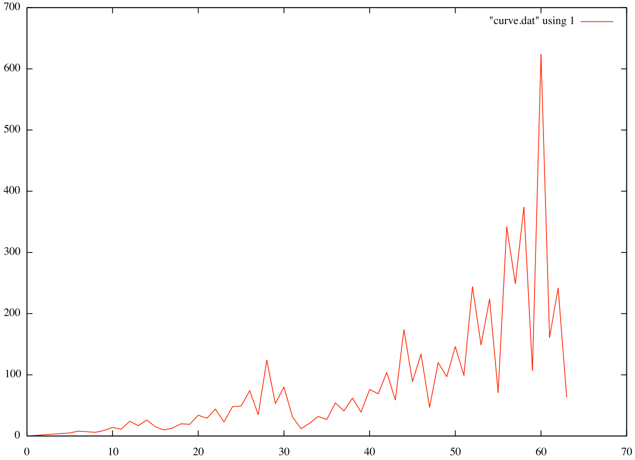

Figure: The automorphism auto_m2p

Future work

- lifting our Haskell implementation to a generic type class based which allows experimenting with instances parameterized by arbitrary bijections between N and [N]
- multiset decompositions of a natural number in O(log(log(n))) factors, similar to the $\omega(x)$ and $\Omega(x)$ (functions counting the distinct and non-distinct prime factors of x) to mimic more closely the distribution of primes
- open problem: can we find a matching additive operation for some multiset of factors induced commutative monoid?

Conclusion

- we have explored some computational analogies between multisets, natural number encodings and prime numbers
- emulating more difficult number theoretic phenomena through simpler isomorphic representations reveals interesting shared behaviors
- like in the case of *abstract interpretation*, we use a simpler domain to approximate properties of a more complex one

Acknowledgment: we thank NSF for support (research grant 1018172)