Calculating the Sieve of Eratosthenes

Lambert Meertens
Kestrel Institute & Utrecht University
The Sieve, informally

- Write down the successive "plurals": 2, 3, 4, ...

- Repeat:
  - Take the first number that is not circled or crossed out
  - Circle it
  - Cross out its proper multiples
Shown in action . . .

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>⨳</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>...</td>
</tr>
</tbody>
</table>
Folklore Functional Program

There is a well-known “folklore” functional program for the Sieve.

How to derive that program?

By calculation, of course!
The Essence of Sieve-hood

The Sieve produces a stream of primes, and that stream is used while it is being produced to filter itself.
Preliminaries: Streams

Always an infinite list
Codomain of final coalgebra
Corresponding anamorphism:

\[ h \times = f \times : h(g \times) \]

Notation:

\[ h = [f \triangle g] \]
Particular case

$\left[ f \triangle (+1) \right]$, in which $(+1)$ is the successor function on naturals

Claim:

$\left[ f \triangle (+1) \right] n = map f \ [n..]$
Proof:

\[
\begin{align*}
\text{map } f \ [n..] \\
= & \quad \text{\{definition of } \ldots \text{\}} \\
\text{map } f \ (n : [n+1..]) \\
= & \quad \text{\{definition of } map\text{\}} \\
\text{f n : map } f \ [n+1..]
\end{align*}
\]
Preliminaries: Primes

If $\text{prime}_0 = 2$, $\text{prime}_1 = 3$, $\text{prime}_2 = 5$, etc.

$$\text{primes} = \text{map prime } [0..]$$

Needs characterization of function $\text{prime}$
Being Prime

A prime is a plural not divisible by a smaller prime

So prime\( n \) is the head of the stream remaining after removing from [2..] the multiples of prime 0, prime 1, ..., up to but not including prime\( n \)
In Haskell

\[
prime n = \text{head}\ (\text{remvto}\ n\ [2 \ldots])
\]

where

\[
\text{remvto } 0 = id
\]

\[
\text{remvto } (n+1) = \text{filter}\ (\text{notdiv}\ (prime\ n)) \cdot \text{remvto}\ n
\]

\[
\text{notdiv}\ d\ n = n \mod d \neq 0
\]

This is actually an effective definition (11)
Strengthening

\[
\text{head}(\text{remvto } n \ [2..]) = \text{prime } n \\
\text{tail} \ (\text{remvto } n \ [2..]) = \\
\text{remvto } n \ [(\text{prime } n) + 1 ..]
\]

So

\[
\text{remvto } n \ [2..] = \\
\text{prime } n : \text{remvto } n \ [(\text{prime } n) + 1 ..]
\]
Generalize

\[ \textit{primes} = pp\ 0 \]

where

\[ pp\ n = \text{map}\ \textit{prime}\ [n..] \]

So

\[ pp\ n = \textit{prime}\ n : pp\ (n+1) \]
Calculating the solution

We want a solution in “sieve” form:

\[ pp\, n = sieve\,(remvto\, n\, [2\ldots]) \]

for some function \( sieve \)

Derive \( sieve \) by matching to the anamorphism pattern

Abbreviate \( prime\, n \) to \( p \) throughout
Left-hand side

\[ pp \ n = \ \{\text{sieve form}\} \]
\[ \text{sieve} (\text{remvto} \ n \ [2 \ldots]) = \ \{\text{property of remvto}\} \]
\[ \text{sieve} (p : \text{remvto} \ n \ [p+1 \ldots]) = \ \{\text{abbreviating to ‘ns’}\} \]
\[ \text{sieve} (p : \text{ns}) \]
Right-hand side

\[ p : pp (n+1) \]
\[ = \quad \text{sieve form} \]
\[ p : \text{sieve} \left( \text{remvto} \ (n+1) \ [2\ldots] \right) \]
\[ = \quad \text{definitions} \]
\[ p : \text{sieve} \left( \text{filter} \ (\text{notdiv} \ p) \ (\text{remvto} \ n \ [2\ldots]) \right) \]
\[ = \quad \text{property of remvto} \]
\[ p : \text{sieve} \left( \text{filter} \ (\text{notdiv} \ p) \ (p : \text{remvto} \ n \ [p+1\ldots]) \right) \]
\[ = \quad \text{abbreviating as before} \]
\[ p : \text{sieve} \left( \text{filter} \ (\text{notdiv} \ p) \ (p : \text{ns}) \right) \]
\[ = \quad \text{notdiv} \ p \ p = \ False, \ definition \ of \ filter \}
\[ p : \text{sieve} \left( \text{filter} \ (\text{notdiv} \ p) \ \text{ns} \right) \]
The Solution for *sieve*

Any definition of *sieve* equating the final expressions of the last two calculations:

\[
\text{sieve} (p : ns) \equiv p : \text{sieve} (\text{filter} (\text{notdiv} \ p) \ ns)
\]

will do

If we forget that \( p \) and \( ns \) denote abbreviations, this is a fine definition
So for primes ... 

primes
= {definition}
map prime [0..]
= {definition}
pp 0
= {sieve form}
sieve (remvto 0 [2..])
= {definitions}
sieve [2..]
Wrapping it up:

\[ \text{primes} = \text{sieve} \ [2..] \]

where

\[ \text{sieve} \ (p : ns) = p : \text{sieve} \ (\text{filter} \ (\text{notdiv} \ p) \ ns) \]