Rational Symmetry Function

by Sven Nilsen, 2025

In this paper I introduce a framework for studying primes that are motivated by symmetries in rational numbers. I give axioms for rational symmetry functions and show that this is a unique and well defined strongly normalization function. I show associated fix-point sequence is square free. Finally, I show that this function can be used to describe equations relevant for prime research.

A rational symmetry function \`f\` has the following axioms:

```
f(0) = 0    f(1) = 1    f(x) = f(1/x)     f(x⋅y) = f(x)⋅f(y)

f : ℚ → ℕ          f{ℕ} . f{ℕ} = f{ℕ}      ava_idem(f)
```

Notice that \`f\` returns a natural number and is idempotent over natural numbers. Furthermore, it is avatar idemponent which means that for any irreducible representation \`n\`, \`f(n) = n\`.

For example:

```
f(x²) = 1                   when `x != 0`
```

Proof:

```
∵     f(x²) = 1
∴     f(x⋅x) = 1
∴     f(x)⋅f(x) = 1           using `f(x⋅y) = f(x)⋅f(y)`
∴     f(x)⋅f(1/x) = 1         using `f(x) = f(1/x)`
∴     f(x⋅1/x) = 1            using `f(x⋅y) = f(x)⋅f(y)`
∴     f(x/x) = 1
∴     f(1) = 1                using `x/x = 1` when `x != 0`
∴     1 = 1                   using `f(1) = 1`
∴     true
```

One can use this theorem to prove the following:

```
f(x²⋅y) = f(y)              when `x != 0`
```

Proof:

```
∵     f(x²⋅y) = f(y)
∴     f(x²)⋅f(y) = f(y)
∴     1⋅f(y) = f(y)           using `f(x²) = 1` when `x != 0`
∴     f(y) = f(y)
∴     true
```

Since \`f\` is idempotent over natural numbers, it can not map natural numbers to other natural numbers than given by minimal reduction. This follows from this proof:

```
∵     f(f(x)) = f(x)          f(x) = y
∴     f(y) = y                using `f(x) = y`
```

Avatar idempotency implies that the above proof using idempotency applies to all primes, because primes are irreducible representations by the given axioms, except \`ava_idem(f)\`:

```
f(2) = 2    f(3) = 3    f(5) = 5    f(7) = 7    f(11) = 11  …
```

From the definition of primes, it follows that any composite number with unique primes, which means all prime powers by factorization equals 1, are irreducible:

```
f(x) = x    <=>   uniq_primes(x)
```

This property is the most fundamental property for this framework of studying primes that arises from the rational symmetry function. I name it `rsf` to distinguish it from other contexts where `f` is used to describe some unknown function.

```
rsf(x) = x  <=>   uniq_primes(x)
```

With other words, the fix-points of `rsf` are natural numbers with unique primes.

The `rsf` function has a natural number sequence where for any `n`, `rsf(n) = n`:

0, 1, 2, 3, 5, 6, 7, 10, 11, 13, 14, 15, 17, 19, 21, 22, 23, 26, 29, 30, 31, 33, 34, 35

Except the first number `0`, this sequence is given by A005117 in The On-line Encyclopedia of Integer Sequences (https://oeis.org/A005117).

Now, the reason the `rsf` function is interesting is because it allows equations using fix-points:

```
∵      rsf(n·(n+1)) = n·(n+1)
∴      rsf(n)·rsf(n+1) = n·(n+1)
```

There is no solution to this equation where `rsf(n) != n`.

However, instead of proving this for the particular case above, consider a more general equation:

```
∵      rsf(x·y) = x·y                  rsf(x) = x·1/z    z > 1
∴      rsf(x)·rsf(y) = x·y
∴      rsf(x)·rsf(y) = (x·1/z)·(y·z)
∴      rsf(y) =  y·z
∴      false
```

All equations of the form `rsf(x·y) = x·y` implies `rsf(x) = x` and `rsf(y) = y`, because there is nowhere else to put the counter-reduction.

Very often, the `rsf` function can simplify problems about primes that are composed together to produce new primes, such as primbixes:

```
∵      rsf(2·r·s·(2·r·s+1)) = 2·r·s·(2·r·s+1)          where `r, s` are primes
∴      rsf(2·r·s)·rsf(2·r·s+1) = 2·r·s·(2·r·s+1)
∴      rsf(2)·rfs(r)·rfs(s)·rsf(2·r·s+1) = 2·r·s·(2·r·s+1)
∴      2·r·s·rsf(2·r·s+1) = 2·r·s·(2·r·s+1)
∴      rsf(2·r·s+1) = 2·r·s+1                    dividing by primes on both sides
```

To have a primbix requires ` 2·r·s+1` to be a prime, but here it merely has unique primes. A generalized version of primbixes is that one has the following equation without contraints:

```
rsf(r·s·(2·r·s+1)) = r·s·(2·r·s+1)
```

Here, `r` and `s` can not have a common divisor, so they both need unique non-overlapping primes. One such example is `r = 2` and `s = 15`, which produces ` 2·r·s+1 =  2·2·15+1 = 61`. Since `61` is a prime and hence has unique primes, this satisfies the equation above. It is not a primbix, though.

I can use `rsf` to derive a fast filter algorithm for primbix candidates:

$$\because \quad y = 2 \cdot r \cdot s+1$$
$$\therefore \quad y-1 = 2 \cdot r \cdot s$$
$$\therefore \quad (y-1)/2 = r \cdot s$$

$$\therefore \quad \mathrm{rsf}(r \cdot s \cdot (2 \cdot r \cdot s+1)) = r \cdot s \cdot (2 \cdot r \cdot s+1)$$
$$\therefore \quad \mathrm{rsf}((y-1)/2 \cdot y) = (y-1)/2 \cdot y$$

This might seem like cheating, but it is not. For any `y : ℕ ∧ (> 1) ∧ odd`, because `(y-1)/2` needs to be a whole number, I can calculate `(y-1)/2·y` and look for any square divisor. Notice that this does not need to be a square prime divisor, which is slow. Since any reduction requires some square, I can check directly whether `rsf` would reduce and hence skip numbers that fail this check.

One thing I have not talked about yet is the following problem:

$$\mathrm{rsf}(2 \cdot r \cdot s) = 2 \cdot r \cdot s$$

Naturally, neither `r` nor `s` can contain `2`. However, one can transform this into:

$$\mathrm{rsf}(r \cdot s) = r \cdot s$$

Here, either `r` or `s` can contain `2`. What is going on here?

Previously, I made a generalized version of primbixes:

$$\mathrm{rsf}(r \cdot s \cdot (2 \cdot r \cdot s+1)) = s \cdot (2 \cdot r \cdot s+1)$$

On the other hand, if I wrote this:

$$\mathrm{rsf}(2 \cdot r \cdot s \cdot (2 \cdot r \cdot s+1)) = 2 \cdot r \cdot s \cdot (2 \cdot r \cdot s+1)$$

Then this would exclude some primbixes such as `13`. If you knew this and payed close attention, then you might notice that I did this in a previous proof. However, this was just as an example from which I did not make further conclusions. I leave it as a warning for you who might repeat the error.

In some applications, such as minimum primbix value search, it is important to have a greater filter than the set of primbixes, or else you might miss some primbixes. Now you know how easy it is to make a mistake by accident when transforming equations using the `rsf` function. Be careful.

Now, do you think that the generalized version of primbixes cover all primbixes? Just because I avoided making one mistake, does not imply that I did not made any others. Take a look at this:

$$2 \cdot 5 \cdot 5+1 = 51$$

Do you see it? Is is a symmetric primbix. For symmetric primbixes… `r·s = r·r` which is a square. This means, if I am searching for some minimum primbix value and there happens to be some minimum primbix value that is symmetric, then I would have a bug in the algorithm. Are there any symmetric primbixes in the minimum primbix sequence? Not very likely. However, who knows?

This is my second warning to you. Using the `rsf` function is really tricky. I leave the errors here in the paper so people can learn from them. The alternative is to make these mistakes in production code, particular in primbix search, which are algorithms that people might run for years at a time. It would not be as fun, if it was not hard. Just, do not be like the knot-theorists (40 year errors).