A Formal Verification of Reversible Primitive Permutations

Giacomo Maletto

1. The definition

1.1 Reversible computing

Reversible computing is a model of computation in which every process can be run backwards. Simply put, in a reversible setting any program takes inputs and gives outputs (like usual), but can also go the other way around: provided the output it can reconstruct the input. In a mathematical sense, every function is expected to be invertible.

Why do we care about such a thing?

Firstly, having a programming language in which every function (or even a subset of functions) is reversible could lead to interesting and practical applications.

But we can also imagine reversible computers, in which the underlying architecture is inherently reversible: Toffoli gates provides a way to do so. The opposite of reversibility is loss of information, which (for thermodynamic reasons) leads to loss of energy and heat dissipation. This means that a non-reversible gate dissipates energy each time information is discarded, while in principle a reversible computer wouldn't.

Lastly, reversible computing is directly related to quantum computing, as each operation in a quantum computer must be reversible.

1.2 Reversible Primitive Permutations

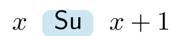
In the article I decided to formalize, the authors focus on providing a functional model of reversible computation. They develop an inductively defined set of functions, called **Reversible Primitive Permutations** or **RPP**, which are expressive enough to represent all Primitive Recursive Functions (we talk about what this means in section ?). Here is the definition that we will use:

Definition 1.2.1 (Reversible Primitive Permutations)**.**
The class of **Reversible Primitive Permutations** or RPP is the smallest subset of functions $\mathbb{Z}^n \to \mathbb{Z}^n$ satisfying the following conditions:

- The **identity** $\mathsf{Id}(x) = x$ belongs to RPP.

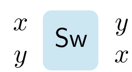
- The **sign-change** $\mathsf{Ne}(x) = -x$ belongs to RPP.

- The **successor function** $\mathsf{Su}(x) = x + 1$ belongs to RPP.



- The **predecessor function** $\mathsf{Pr}(x) = x - 1$ belongs to RPP.

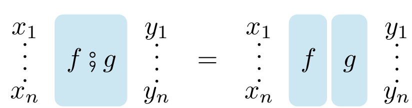
- The **swap** $\mathsf{Sw}(x, y) = (y, x)$ belongs to RPP.



- If $f : \mathbb{Z}^n \to \mathbb{Z}^n$ and $g : \mathbb{Z}^n \to \mathbb{Z}^n$ belongs to RPP, then the **series composition** $(f \fatsemi g) : \mathbb{Z}^n \to \mathbb{Z}^n$ belongs to RPP and is such that:

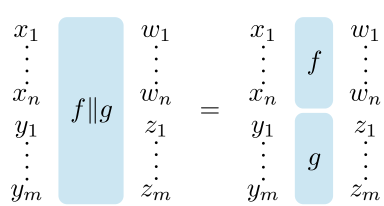
$$(f \fatsemi g)(x_1, \ldots, x_n) = g(f(x_1, \ldots, x_n)) = (g \circ f)(x_1, \ldots, x_n)).$$

  We remark that $f \fatsemi g$ means that f is applied first, and then g, in opposition to the standard functional composition (denoted by $\circ$).



- If $f : \mathbb{Z}^n \to \mathbb{Z}^n$ and $g : \mathbb{Z}^m \to \mathbb{Z}^m$ belongs to RPP, then the **parallel composition** $(f \| g) : \mathbb{Z}^{n+m} \to \mathbb{Z}^{n+m}$ belongs to RPP and is such that:

$$(f \| g)(x_1, \ldots, x_n, y_1, \ldots, y_m) = (f(x_1, \ldots, x_n), g(y_1, \ldots, y_m)).$$



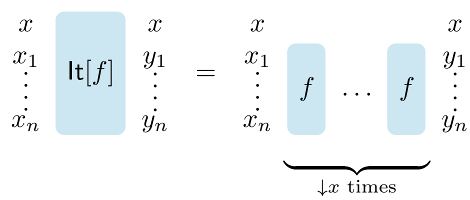
- If $f : \mathbb{Z}^n \to \mathbb{Z}^n$ belongs to RPP, then then **finite iteration** $\mathsf{It}[f] : \mathbb{Z}^{n+1} \to \mathbb{Z}^{n+1}$ belongs to RPP and is such that:

$$\mathsf{It}[f](x, x_1, \ldots, x_n) = (x, (\overbrace{f \circ \cdots \circ f}^{\downarrow x \text{ times}})(x_1, \ldots, x_n))$$

where $\downarrow (\cdot) : \mathbb{Z} \to \mathbb{N}$ is defined as

$$\downarrow x = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}.$$

This means that the function f is applied $\downarrow x$ times to $(x_1, \ldots, x_n)$.



- If $f, g, h : \mathbb{Z}^n \to \mathbb{Z}^n$ belongs to RPP, then the **selection** $\mathsf{If}[f, g, h] : \mathbb{Z}^{n+1} \to \mathbb{Z}^{n+1}$ belongs to RPP and is such that:

$$\mathsf{If}[f, g, h](x, x_1, \ldots, x_n) = \begin{cases} (x, f(x_1, \ldots, x_n)), & \text{if } x > 0 \\ (x, g(x_1, \ldots, x_n)), & \text{if } x = 0 \\ (x, h(x_1, \ldots, x_n)), & \text{if } x < 0 \end{cases}.$$

We remark that the argument x which determines which among f, g and h must be used cannot be among the arguments of f, g and h, as that would break reversibility.