1. Information about the carrier set

Let A denote a free abelian group on countably many generators. If it is easier, then after tensoring by $\mathbb{Q}$ we can alternatively replace A by a countable-dimensional vector space over a field (of characteristic 0, to keep things generic).

In any case, for concreteness we can think of A as the set of $\mathbb{N}$-indexed tuples (over $\mathbb{Z}$ or $\mathbb{Q}$, whichever is more convenient) with finite support. This set is generated by the basis $\mathfrak{B} = \{e_1, e_2, \ldots\}$, where e_i has 1 in the ith coordinate, and zeros elsewhere.

It will be convenient for us to have a countable list $S_0, S_1 \ldots$, where $S_i \cap S_j = \emptyset$, $|S_i| = \aleph_0$, and $\bigcup_{i \in \mathbb{N}} S_i = \mathfrak{B}$. This can be done, for example, by taking

$$S_i = \{e_j \in \mathfrak{B} : j \equiv 2^i \pmod{2^{i+1}}\}.$$

2. A generic solution to the functional equation for 1692

Equation 1692 is $x = (y \diamond x) \diamond ((y \diamond x) \diamond y)$. Setting $x \diamond y := x + f(h)$, where $h := y - x$ and f is a self-map on A, we see that we want

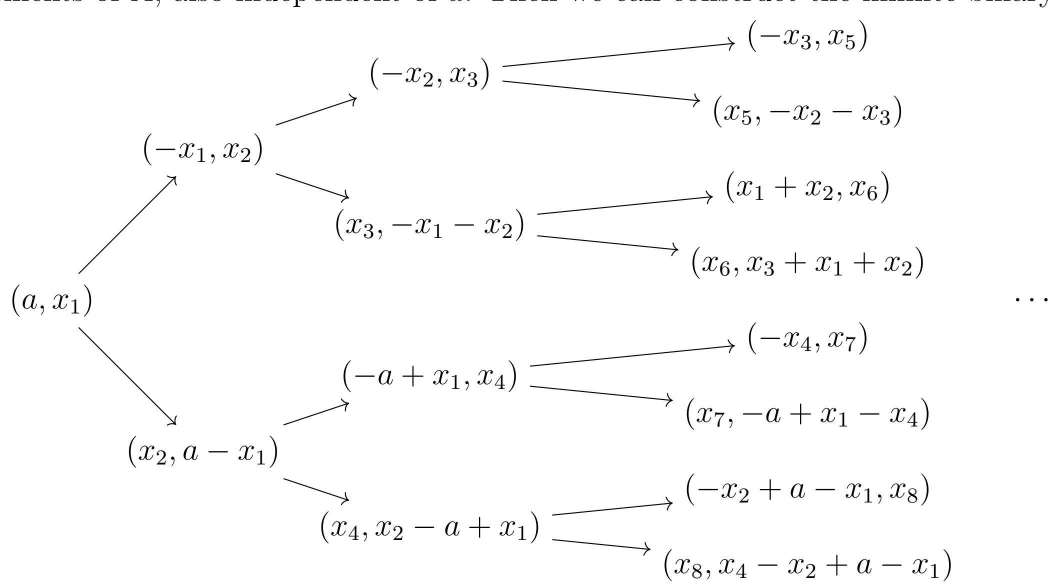
$$\begin{aligned} x &= (x + h + f(-h)) \diamond ((x + h + f(-h)) \diamond (x + h)) \\ &= (x + h + f(-h)) \diamond (x + h + f(-h) + f(-f(-h))) \\ &= x + h + f(-h) + f^2(-f(-h)). \end{aligned}$$

Rearranging, and replacing h by $-h$, we get the functional equation $f^2(-f(h)) = h - f(h)$.

Rather than showing how to extend finite partial solutions to this functional equation (which I found to be difficult, but perhaps someone else can make that work), I will instead describe how to construct a single generic solution.

The main idea is the following: If f satisfies the functional equation, then given any pair $(a, b) \in f$, there exists some $c \in A$ such that $(-b, c), (c, a - b) \in f$. So we will freely add those pairs, and then for each of the two new pairs we will add two new pairs, and so forth. This leads to to the formation of an infinite bifurcation tree, as follows.

Suppose that $a \in A$ is arbitrary, and suppose that $x_1, x_2, \ldots$ are any linearly independent elements of A, also independent of a. Then we can construct the infinite binary tree



There are some basic facts that we will need about the ordered pairs that appear in this tree. (Feel free to give better/easier proofs of these facts.)

Lemma 2.1. *For each node in the tree, the two entries in the ordered pair are linearly independent, and the second is never zero.*

Proof. Induct on the depth of the node, and use the linear independence of the x_i and the definition of the two children. □

Lemma 2.2. *In the tree above, the only entry in* $\mathrm{Span}(a)$ *is the left-most entry in the left-most ordered pair.*

Proof. It suffices to consider the case when $a = 0$, and then this follows from the previous lemma (for second coordinates), and another induction via depth (for first coordinates). □

Lemma 2.3. *The nodes in the tree form a (partial) function on A, where the functional equation is satisfied at all first coordinates.*

Proof. The second claim is clear after we show the first. I'll leave the first claim for others to justify. □

I don't know if we need the full power of the next lemma, but it helps us verify when non-implications will hold (as to be described later). It can probably be handled on a case-by-case basis to save time and effort when putting it in Lean.

Lemma 2.4. *For any finite subset $F \subseteq \{a, x_1, x_2, \ldots\}$, there are only finitely many coordinates in the ordered pairs appearing in the tree that also live in* $\mathrm{Span}(F)$*, and it is computable how far down the tree one must go before they cease to appear.*

For example, when $F = \{a, x_1\}$, there are exactly 4 nodes where at least one of the coordinates comes from $\mathrm{Span}(F)$.

Now, we construct a (total) generic function f on A, satisfying the functional equation, as follows. Order the elements of A as $a_0, a_1, \ldots$, say with $a_0 = 0$ for convenience. Build the binary tree as above, with $a = a_0$ and with $x_1, x_2, \ldots$ taken as the elements of S_0, and let f_0 be the set of ordered pairs in that tree.

Next, if a_1 occurs as a first coordinate in f_0, take $f_1 := f_0$. Otherwise, build another binary tree starting with $a = a_1$ and $x_1, x_2, \ldots$ taken from S_i where i is the smallest index such that none of its elements appear in the support of a_1, nor any coordinates in f_0. Let f_1 be f_0 unioned with the new pairs from this new tree.

Recursing this way, we are done. Note that the complete function f must be injective (for if $f(a_1) = b = f(a_2)$, and we set $c = f(-b)$, then $a_1 - b = f(c) = a_2 - b$).

3. Non-implications

Equation 23 is $x = (x \diamond x) \diamond x$. The corresponding functional equation is $f(0) + f(-f(0)) = 0$. Our generic function has $f(0) = e_1$ and $f(-e_1) = e_3$. (The first two nodes constructed on the first tree, taking $x_1 = e_1$ and $x_2 = e_3$.) This fails the new functional equation.

Equation 47 is $x = x \diamond (x \diamond (x \diamond x))$. The functional equation is $0 = f^3(0)$. By one of our lemmas, we know that 0 is never an output of our function. (Alternatively, one can trace through the paths, to find that $0 \mapsto e_1$, and $e_1 \mapsto e_7$, and e_7 is the input vector in one of the new trees, and so doesn't go to 0.)

I'll get to the other equations later if others don't get to them first (or find a much simpler argument!).