Free bands

Michael A. Daas

A *semigroup* is an algebraic structure that can be thought of as a group without inverses, and without an identity element. The following definition makes this more precise.

Definition 1. A *semigroup* is a set S equipped with an associative binary operation $\bullet$. This means that $(a \bullet b) \bullet c = a \bullet (b \bullet c)$ for all $a, b, c \in S$.

We are interested in very particular semigroups, called *bands*, also known as *idempotent semigroups*.

Definition 2. A *band* B is a semigroup with the property that $b \bullet b = b$ for all $b \in B$.

A band is called *free* if it is generated by a certain *alphabet*, in which case its elements consist of words of finite length in this alphabet with the operation $\bullet$ given by concatenation of words. Given a positive integer $n \geq 1$, let FB_n denote the band freely generated by n letters.

Example 3. Let $n = 1$. Then all words in FB_1 are generated by words containing only one letter, say x. Therefore, $\mathrm{FB}_1 = \{x, xx, xxx, \dots\} = \{x\}$, where we used that $xx = x$. Therefore, $\#\mathrm{FB}_1 = 1$.

Example 4. The free band FB_2 uses two letters x, y, and it is a fun exercise to check that

$$\mathrm{FB}_2 = \{x, y, xy, yx, xyx, yxy\} \implies \#\mathrm{FB}_2 = 6.$$

No longer words are possible; for example $xyx \bullet x = xyxx = xyx$, and $xyx \bullet y = xyxy = xy$.

For $n > 2$ generators, it is not as easy to see whether or not FB_n is finite. Yet, the following is known.

Theorem 5 (Green-Rees)**.** *The size of FB_n is finite, and in fact, it is given by the formula*

$$\#\mathrm{FB}_n = \sum_{r=1}^{n} \binom{n}{r} \prod_{i=1}^{r} (r - i + 1)^{2^i}.$$

This is quite remarkable, as sometimes words can only be made shorter after first making them longer:

$$xyzyxyz = xyzyx(yz)(yz) = (xyzy)(xyzy)z = x(yz)(yz) = xyz.$$

The theory of bands is therefore rather subtle. The main goals for the project would be as follows:

- Understand the original proof by Green and Rees of Theorem 5 and write it up in your own way.
- Use their methods to answer some related questions. For example, what is the longest word in FB_n? What are the asymptotics for the size of FB_n? How far can we compute these numbers exactly, and can we give some reasonable upper and lower bounds in general?
- Write some computer code that lists all words in FB_3, and if possible, also FB_4.
- Can you come up with other fast ways to test the equality between two elements in FB_n?