Walnut with Ostrowski Numeration Systems

Aseem Baranwal
School of Computer Science, University of Waterloo, Canada, ON
May 10, 2020

1 INTRODUCTION

We assume that the readers are familiar with the basic usage of Walnut; for a quick reference on installation and basic functionality, see Hamoon Mousavi's manuscript on arXiv[1].

This article serves as a reference manual for using Walnut in any Ostrowski numeration system based on a quadratic irrational number α. In Section 2, we talk about the theoretical definition of the Ostrowski numeration system, and in Section 4, we provide the syntax for the commands that enable Walnut to work in any Ostrowski numeration system. Walnut is an open source software package available for free. To work with the version of Walnut equipped with this feature, please download it from GitHub.

2 OSTROWSKI NUMERATION SYSTEM

Named after the mathematician Alexander Markowich Ostrowski[2], the Ostrowski numeration system is a representation system for integers, based on the continued fraction of an irrational number α. It is known that the continued fraction of an irrational number is infinite.

Definition 1. Let $\alpha = [d_0; d_1, d_2, \ldots]$, and let $(p_i)_{i\geq 0}$ and $(q_i)_{i\geq 0}$ be the sequences denoting respectively, the numerator and denominator of the convergents of α. Then, the *Ostrowski-α numeration system* is defined as a representation system for non-negative integers, such that any $N \geq 0$ can be uniquely represented in MSD-first notation as

$$N = [a_{n-1}a_{n-2}\cdots a_0]_\alpha = \sum_{0\leq i<n} a_i q_i,$$

where

1. $0 \leq a_0 < d_1$,
2. $0 \leq a_i \leq d_{i+1}$, for $i \geq 1$, and
3. for all $i \geq 1$, if $a_i = d_{i+1}$ then $a_{i-1} = 0$.

We call such a unique representation a *canonical* representation of N in the corresponding Ostrowski-α numeration system and write it as

$$N = [a_{k-1}a_{k-2}\cdots a_0]_\alpha.$$

We observe that d_0 plays no role in the representation. Therefore, for all practical purposes we can assume $d_0 = 0$, and hence $0 < \alpha < 1$. Furthermore, we observe that if $d_1 = 1$ then $q_0 = q_1 = 1$, implying that for every non-negative integer N, the least significant digit in the representation, a_0, is always 0. Thus, we can further assume that $d_1 \geq 2$, so that $0 < \alpha < 1/2$.

Example 1. Consider $\alpha = 1/\phi^2 = [0; 2, \overline{1}]$, where ϕ is the golden ratio. The corresponding Ostrowski numeration system is the Zeckendorf representation, defined by the Fibonacci sequence $(q_n)_{n\geq 0} = 1, 2, 3, 5, 8, \ldots$. For example, the representation for $N = 23$ is $[1000010]_\alpha$.

Example 2. Consider $\beta = 1/\delta_S = \sqrt{2} - 1 = [0; \overline{2}]$, where $\delta_S = \sqrt{2} + 1$ is the silver ratio. The corresponding Ostrowski numeration system is based on the Pell sequence $(q_n)_{n\geq 0} = 1, 2, 5, 12, \ldots$. For example, the representation for $N = 27$ is $[2011]_\beta$.

3 INSTALLATION

On most Unix based systems, the following steps will perform a one-time setup for Walnut 2.0.

1. From GitHub, download Walnut's master branch as a zip file [direct link].
2. Extract the archive's contents in a directory.
3. On the command-line, change directory to '`Walnut-master`', the top-level directory.
4. Make sure that JDK and JRE are installed using:
   ```
   javac -version
   java -version
   ```
5. Execute the following:
   ```
   source build.sh
   ```
6. Now Walnut is compiled, and can be run by navigating to the '`bin/`' directory and executing:
   ```
   java Main.Prover
   ```

4 SYNTAX

To enable Walnut to work in an arbitrary Ostrowski numeration system associated with the irrational number α, we use the "`ost`" command as follows.

```
ost <name> <preperiod> <period>;
```

The command takes three parameters:

1. `name`, the name we want to give to the new numeration system;
2. `<preperiod>`, the non-repeating initial part of the continued fraction of α; and
3. `<period>`, the repeating tail of the continued fraction.

For example, the command for $\alpha = (5 + \sqrt{13})/6 = [0; 3, 1, \overline{1, 2}]$, we type in the following command in Walnut:

```
ost numsys [0 3 1] [1 2];
```

This command will create two new files in the directory "`/Walnut/Custom Bases/`" with the names:
"`msd_numsys.txt`", and
"`msd_numsys_addition.txt`".

These files contain the representation and the adder automaton that Walnut created for this numeration system. We can now use this numeration system like usual in any predicate. For example, to create an automaton that accepts the Ostrowski-α representations of all even integers n we input the following predicate in Walnut:

```
eval isEven "?msd_numsys Ey x=2*y";
```

REFERENCES

1. H. Mousavi. Automatic theorem proving in Walnut. *arXiv preprint arXiv:1603.06017*, 2016.

2. A. Ostrowski. Bemerkungen zur Theorie der Diophantischen Approximationen. *Abh. Math. Sem. Hamburg*, 1:77–98, 250–251, 1922. Reprinted in *Collected Mathematical Papers*, Vol. 3, pp. 57–80.