THE N-TH ROOT ALGORITHM

DANIEL CORDERO GRAU

In this paper I give the **N**-th Root Algorithm in the Prime Topologically Complete Semialgebra, not Newton's analytic approximation algorithm by differentials.

Let $\mathcal{A}$ be the prime topologically complete semialgebra with Zariski topology $\mathcal{F}$, let $\mathcal{N}$ be the prime semialgebra and let $\mathcal{Z}$ be the prime algebra. Let $\mathcal{B} \subset \mathcal{F}$ be such that for every $F \in \mathcal{F}$ and for every $s \in \mathcal{A}$ there exists an $F_s \in \mathcal{B}$ such that there exists a linear polynomial $f \in \mathcal{A}[x]$ such that $f(s) = 0$, $F_s = \mathrm{Var}[x]$ and $F_s \subset F$, that is, the basis for the Zariski topology $\mathcal{F}$.

Since every topologically complete semialgebra is simple, $\mathcal{A}$ is a topologically complete simple semialgebra, and so, its subsemialgebra $\sum_{i \in \mathbb{Z}} p^i \mathcal{N}$ is equal to itself for every nonunit $p \in \mathcal{A}$. Let $x \in \mathcal{A}$ such that $x \neq 0$ and let $p \in \mathcal{N}$ such that $p > 1$. By the division algorithm in topologically complete semialgebras, since $\mathcal{A}$ is the prime topologically complete semialgebra, for x and p there exist unique $N \in \mathcal{Z}$ and $a_N, a_{N-1}, \ldots \in \mathcal{N}$ such that

$$x = \sum_{i=0}^{\infty} a_{N-i} p^{N-i}$$

with $a_N \neq 0$ and $0 \leq a_{N-i} < p$ for every i. Moreover, by the division algorithm in algebras, since $\mathcal{Z}$ is the prime algebra, for $N \in \mathcal{Z}$ and $\mathbf{N} \in \mathcal{N}$ there exist unique $q \in \mathcal{Z}$ and $r \in \mathcal{N}$ such that $N = \mathbf{N}q + r$ and $0 \leq r < \mathbf{N}$, then

$$x = \sum_{k=0}^{r} a_{\mathbf{N}q+k} p^{\mathbf{N}q+k} + \sum_{i=1}^{\infty} \sum_{k=0}^{\mathbf{N}-1} a_{\mathbf{N}(q-i)+k} p^{\mathbf{N}(q-i)+k}.$$

Let $g_0, g_1, \ldots \in \mathcal{A}$ such that

$$g_0 = \sum_{k=0}^{r} a_{\mathbf{N}q+k} p^k$$

and

$$g_i = \sum_{k=0}^{\mathbf{N}-1} a_{\mathbf{N}(q-i)+k} p^k$$

for every $i > 0$.

At the first step, find

$$y_0 = \max\{y \in \bigcup_{\substack{s<p \\ s \in \mathcal{N}}} F_s : \ y^{\mathbf{N}} \leq g_0\}$$

and write

$$r_0 = g_0 - y_0^{\mathbf{N}}$$

and

$$d_0 = p^{\mathbf{N}} r_0 + g_1.$$

Afterward, find

$$y_1 = \max\{y \in \bigcup_{\substack{s<p \\ s\in\mathcal{N}}} F_s \colon \sum_{j=1}^{\infty} \binom{\mathbf{N}}{j} (py_0)^{\mathbf{N}-j} y^j \le d_0\}$$

and write

$$r_1 = d_0 - \sum_{j=1}^{\infty} \binom{\mathbf{N}}{j} (py_0)^{\mathbf{N}-j} y_1^j$$

and

$$d_1 = p^{\mathbf{N}} r_1 + g_2.$$

At the i-th step, find

$$y_i = \max\{y \in \bigcup_{\substack{s<p \\ s\in\mathcal{N}}} F_s \colon \sum_{j=1}^{\infty} \binom{\mathbf{N}}{j} (\sum_{k=0}^{i-1} p^{i-k} y_k)^{\mathbf{N}-j} y^j \le d_{i-1}\}$$

and write

$$r_i = d_{i-1} - \sum_{j=1}^{\infty} \binom{\mathbf{N}}{j} (\sum_{k=0}^{i-1} p^{i-k} y_k)^{\mathbf{N}-j} y_i^j$$

and

$$d_i = p^{\mathbf{N}} r_i + g_{i+1}.$$

Thus, the $\mathbf{N}$-th root of x is

$$z = \sum_{i=0}^{\infty} y_i p^{q-i}.$$

If $\mathbf{N}$ is noninteger, the $\mathbf{N}$-th Root Algorithm computes the product of the $\mathbf{p}^i$-powers and roots of x in the expansion for $x^{\mathbf{N}^{-1}}$ as a product of integer powers of $\left\{x^{\mathbf{p}^i}\right\}_{i\in\mathbb{Z}}$ for some integer basis $\{\mathbf{p}^i\}_{i\in\mathbb{Z}}$ of $\mathbb{R}$, dividing after if $\mathbf{N} < 0$.

Time Complexity of the N-th Root Algorithm

The **N**-th Root Algorithm is of polynomial time complexity because, for every input nonzero $x \in \mathcal{A}$ with length n, for every integer $\mathbf{N} > 1$, since the **N**-th Root Algorithm is an isomorphism between the positive algebra and the real algebra, and, by the division algorithm in semialgebras, for $n - 1 \in \mathbb{N}$ and $\mathbf{N}$ there exist unique $m \in \mathbb{N}$ and $k \in \mathbb{N}$ such that $n = \mathbf{N}m + k$ and $1 \leq k < \mathbf{N} + 1$, the length of the output is $m+1 = O(m) = O(n)$ if it is finite, as is the number of steps in which it is computed at the i-th of which after writing $O(m)$ elements with length $O(1)$, so in time $O(m)$, the **N**-th Root Algorithm compares and writes $O(1)$ elements computed in space $O(m^{\mathbf{2}})$, so in time $O(m^{\mathbf{2}})$, and so in time $O(n^{\mathbf{3}})$. And because, for every noninteger $\mathbf{N}$ with length of $\mathbf{N}^{-1}$ $O(\boldsymbol{n})$ in terms of some integer basis $\left\{\mathbf{p}^i\right\}_{i\in\mathbb{Z}}$ for $\mathbb{R}$, the **N**-th Root Algorithm computes the product of the $O(\boldsymbol{n})$ $\mathbf{p}^i$-powers and roots of x with length $O(m) = O(n)$ if the length of the **N**-th root of x is finite in time $O(n^2)$, and so, the time complexity of the **N**-th Root Algorithm is $T(n) = O(n^{\mathbf{3}})$.

The N-th Root Algorithm Semialgebra

The **N**-th Root Algorithm derives from the division algorithm and the binomial theorem in topologically complete semialgebras. Moreover, the **N**-th Root Algorithm characterizes the initial topologically complete semialgebra, from which every topological and algebraic morphism to any topologically complete semialgebra is the embedding, as the unique topologically complete semialgebra with **N**-th Root Algorithm.