An Extension of Newton's Method to ω-Continuous Semirings *

Javier Esparza, Stefan Kiefer, and Michael Luttenberger

Institute for Formal Methods in Computer Science
Universität Stuttgart, Germany
{esparza,kiefersn,luttenml}@informatik.uni-stuttgart.de

Abstract. Fixed point equations $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$ over ω-continuous semirings are a natural mathematical foundation of interprocedural program analysis. Equations over the semiring of the real numbers can be solved numerically using Newton's method. We generalize the method to any ω-continuous semiring and show that it converges faster to the least fixed point than the Kleene sequence $\mathbf{0}, \boldsymbol{F}(\mathbf{0}), \boldsymbol{F}(\boldsymbol{F}(\mathbf{0})), \ldots$ We prove that the Newton approximants in the semiring of languages coincide with finite-index approximations studied by several authors in the 1960s. Finally, we apply our results to the analysis of stochastic context-free grammars.

1 Introduction

In [2] we have argued that fixed point equations over ω-continuous semirings are a natural mathematical foundation of interprocedural program analysis. In this approach a program is mapped (in a syntax-driven way) to a system of fixed point equations over an abstract semiring. The carrier and the operations of the semiring are instantiated depending on the information about the program one wishes to compute. The information is the least solution of the system.

On ω-continuous semirings one can apply Kleene's fixed point theorem, and so the least solution of a system of equations $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$ is the supremum of the sequence $\mathbf{0}, \boldsymbol{F}(\mathbf{0}), \boldsymbol{F}^2(\mathbf{0}), \ldots$, where $\mathbf{0}$ is the vector whose components are all equal to the neutral element of $+$. If the carrier of the semiring is finite, this yields a procedure to compute the solution. However, if the carrier is infinite, the procedure rarely terminates, and its convergence can be very slow. So it is natural to look for "accelerations". Loosely speaking, an acceleration is a function $\boldsymbol{G}$ having the same least fixed point $\mu\boldsymbol{F}$ as $\boldsymbol{F}$, but such that $(\boldsymbol{G}^i(\mathbf{0}))_{i\geq 0}$ converges faster to $\mu\boldsymbol{F}$ than $(\boldsymbol{F}^i(\mathbf{0}))_{i\geq 0}$.

In [2] we presented a generic acceleration scheme for *commutative* ω-continuous semirings, which we call the *Newton scheme*. We showed that the Newton scheme generalizes two well-known but apparently disconnected acceleration schemes from the literature: Newton's method for approximating a zero of a differentiable function (this is the reason for the name of our scheme) (see for instance

* This work was partially supported by the DFG project *Algorithms for Software Model Checking*.

[12]), and the Hopkins-Kozen iteration scheme for Kleene algebras (which are very close to idempotent commutative semirings) [9].

In this paper we further generalize the Newton scheme of [2] to *arbitrary* ω-continuous semirings, commutative or not. In particular, this allows us to solve systems of fixed point equations over the language semiring having the set of languages over a given alphabet as carrier and union and concatenation of languages as sum and product, respectively. For instance, if we consider this semiring for the alphabet $\{(,)\}$, then the least solution of the equation $X = (X) + XX + 1$ is the Dyck language of well-parenthesized expressions. Clearly, the least solution of a system is a context-free language, and every context-free language is the solution of a system.

The Newton acceleration scheme approximates the least solution from below. In the case of languages, it computes a chain $L_0 \subseteq L_1 \subseteq L_2 \ldots$ of approximations of the least solution $L = \bigcup_{i \geq 0} L_i$. Our main theorem characterizes these approximations, and shows that, once again, a well-known concept from the literature "is nothing but" Newton's approximation technique.

The i-th approximation of the Newton scheme turns out to be the *index-*$(i+1)$ approximation $L_{i+1}(G)$ of $L(G)$. Recall that a terminal word w is in $L_i(G)$ if there is a derivation $S \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_r = w$ and every α_i, $0 \leq i \leq r$ contains at most i occurrences of variables [13, 8, 14, 7].

Our result allows to transfer results from language theory to numerical analysis and vice versa. We develop a way of applying finite-index approximations to stochastic context-free grammars and computing the approximation quality.

It is well-known that Newton's method for approximating the zero of a function is based on the notion of differential. Our results require to give a definition of derivative of a polynomial expressions for arbitrary ω-continuous semirings. This can be seen as a generalization of the Brzozowski's definition of derivative for regular languages and Hopkins and Kozen's definition for commutative semirings, and could have some interest of its own.

Organization and contributions of this paper. In Section 2 we define differentials for power series over ω-continuous semirings. Section 3 introduces a generalized Newton's method for approximating the least solution of fixed point equations over arbitrary ω-continuous semirings. In Section 4 (Theorem 4.1) we characterize the iterates of the Newton scheme in terms of the *tree dimension*, a concept generalized from [2]. We apply this result to context-free grammars in Section 5 and prove that the Newton iterates coincide with finite-index approximations. In Section 6 we apply the generalized Newton's method to stochastic context-free grammars. Missing proofs can be found in a technical report [1].

2 Differentials in ω-Continuous Semirings

The goal of this section is to generalize the notion of differential of a function to ω-continuous semirings. More precisely, we will only define the notion for functions that can be represented as a power series.

Recall the classical notion of differential that can be found in any elementary calculus textbook. Let V be a vector space of finite dimension over the real numbers $\mathbb{R}$. The *dual* of V is the vector space whose elements are the linear functions $V \to \mathbb{R}$, usually called *linear forms*. Given $f\colon V \to \mathbb{R}$, the *differential* Df of f (when it exists) is an application $Df\colon V \to \widetilde{V}$ that assigns to every vector $\boldsymbol{v} \in V$ a linear form $Df|_{\boldsymbol{v}}$. Loosely speaking, $Df|_{\boldsymbol{v}}$ is the best linear approximation of f at the point $\boldsymbol{v}$.

We wish to generalize the notion of differential to the case in which $\mathbb{R}$ is replaced by an arbitrary ω-continuous semiring, and f is a power series. For this, we first introduce ω-continuous semirings in Section 2.1. In Section 2.2 we generalize the notions of vector and linear form over the reals. In Section 2.3 we introduce power series, and finally in Section 2.4 the notion of differential itself.

2.1 ω-Continuous Semirings

In the following, we work with ω-continuous semirings, as defined in [11].

Definition 2.1. *A* semiring $\mathcal{S}$ *is given by* $\langle S, +, \cdot, 0, 1\rangle$*, where* S *is a set with* $0, 1 \in S$*,* $\langle S, +, 0\rangle$ *is a commutative monoid with neutral element* 0*,* $\langle S, \cdot, 1\rangle$ *is a monoid with neutral element* 1*,* 0 *is an annihilator w.r.t.* $\cdot$*, i.e.* $0 \cdot a = a \cdot 0 = 0$ *for all* $a \in S$*, and* $\cdot$ *distributes over* $+$*, i.e.* $a \cdot (b + c) = a \cdot b + a \cdot c$*, and* $(a + b) \cdot c = a \cdot c + b \cdot c$*. The natural order relation* $\sqsubseteq$ *on a semiring* $\mathcal{S}$ *is defined by* $a \sqsubseteq b \Leftrightarrow \exists d \in S : a + d = b$*. The semiring* $\mathcal{S}$ *is* naturally ordered *if* $\sqsubseteq$ *is a partial order on* S*.*

An *ω-continuous semiring* is a naturally ordered semiring extended by an infinite summation-operator $\sum$ that satisfies the following properties[1]:

- For every sequence $a : \mathbb{N} \to S$ the supremum $\sup\{\sum_{0 \le i \le k} a_i \mid k \in \mathbb{N}\}$ exists in S w.r.t. $\sqsubseteq$, and is equal to $\sum_{i \in \mathbb{N}} a_i$. As a consequence, every non-decreasing sequence $a_i \sqsubseteq a_{i+1}$ converges, i.e. $\sup\{a_i\}$ exists.
- It holds
$$\sum_{i\in\mathbb{N}}(c\cdot a_i) = c\cdot\left(\sum_{i\in\mathbb{N}} a_i\right), \quad \sum_{i\in\mathbb{N}}(a_i\cdot c) = \left(\sum_{i\in\mathbb{N}} a_i\right)\cdot c, \quad \sum_{j\in J}\left(\sum_{i\in I_j} a_j\right) = \sum_{i\in\mathbb{N}} a_i$$
for every $a : \mathbb{N} \to S$, $c \in S$, and every partition $(I_j)_{j\in J}$ of $\mathbb{N}$.

In the following we often omit the dot $\cdot$ in products.

Example 2.1. The *real semiring*, denoted by $\mathcal{S}_{\mathbb{R}}$, has $\mathbb{R}_{\ge 0} \cup \{\infty\}$ as carrier. Sum and multiplication are defined as expected (e.g. $a \cdot \infty = \infty$ for $a \neq 0$). Notice that sum is not idempotent and product is commutative.

The *language semiring* over an alphabet Σ, denoted by $\mathcal{S}_\Sigma$, has the set of all languages over Σ as carrier. Sum is union, and product is concatenation of languages. Notice that sum is idempotent and product is not commutative.

[1] [11] requires infinite summation for *any* sum, but we need only countable sums here.

2.2 Vectors and Linear Forms

We introduce the notion of vectors and linear forms over an ω-continuous semiring $\mathcal{S}$. Notice that the name vector and linear form have to be taken with a grain of salt, because for instance the set of vectors over $\mathcal{S}$ does not build a vector space (since $\mathcal{S}$ may not be a field). However, it is useful to keep the names to remember that they generalize the usual notions of vector and linear form.

Definition 2.2. *Let $\mathcal{S}$ be an ω-continuous semiring and let $\mathcal{X}$ be a finite set of variables.*

A vector *is a mapping $\boldsymbol{v}\colon \mathcal{X} \to S$. The set of all vectors is denoted by V. Given a countable set I and a vector $\boldsymbol{v}_i$ for every $i \in I$, we denote by $\sum_{i\in I} \boldsymbol{v}_i$ the vector given by $\left(\sum_{i\in I} \boldsymbol{v}_i\right)(X) = \sum_{i\in I} \boldsymbol{v}_i(X)$ for every $X \in \mathcal{X}$.*

A linear form *is a mapping $l\colon V \to S$ satisfying $l(\boldsymbol{v} + \boldsymbol{v'}) = l(\boldsymbol{v}) + l(\boldsymbol{v'})$ for every $\boldsymbol{v}, \boldsymbol{v'} \in V$ and $l(\mathbf{0}) = 0$, where $\mathbf{0}$ denotes the vector given by $\mathbf{0}(X) = 0$ for every $X \in \mathcal{X}$. Given a linear form l and $s, s' \in S$, we denote by $s \cdot l \cdot s'$ the linear form given by $(s \cdot l \cdot s')(\boldsymbol{v}) = s \cdot l(\boldsymbol{v}) \cdot s'$ for every $v \in V$. Given a countable set I and a linear form l_i for every $i \in I$, we denote by $\sum_{i\in I} l_i$ the linear form given by $\left(\sum_{i\in I} l_i\right)(\boldsymbol{v}) = \sum_{i\in I} l_i(\boldsymbol{v})$ for every $\boldsymbol{v} \in V$.*

2.3 Polynomials and Power Series

Definition 2.3. *Let $\mathcal{S}$ be an ω-continuous semiring and $\mathcal{X}$ be a finite set of variables. A* monomial *is a finite expression*

$$a_1 X_1 a_2 \cdots a_k X_k a_{k+1}$$

where $k \geq 0$, $a_1, \ldots, a_{k+1} \in S$ and $X_1, \ldots, X_k \in \mathcal{X}$. A polynomial *is an expression of the form $m_1 + \ldots + m_k$ where $k \geq 0$ and $m_1, \ldots, m_k$ are monomials. We let $\mathcal{S}[\mathcal{X}]$ denote the set of polynomials w.r.t. $\mathcal{S}$ and $\mathcal{X}$. Similarly, a* power series *is an expression of the form $\sum_{i\in I} m_i$, where I is a countable set and m_i is a monomial for every $i \in I$. We use $\mathcal{S}[\![\mathcal{X}]\!]$ to denote this set.*

Definition 2.4. *Let $f = \alpha_1 X_1 \alpha_2 X_2 \alpha_3 \ldots \alpha_k X_k \alpha_{k+1} \in \mathcal{S}[\![\mathcal{X}]\!]$ be a monomial and let $\boldsymbol{v}$ be a vector. We define $f(\boldsymbol{v})$, the* evaluation of f at $\boldsymbol{v}$, *as*

$$f(\boldsymbol{v}) = \alpha_1 \boldsymbol{v}(X_1) \alpha_2 \boldsymbol{v}(X_2) \alpha_3 \cdots \alpha_k \boldsymbol{v}(X_k) \alpha_{k+1}.$$

We extend this to any power series $f = \sum_{i\in I} f_i \in \mathcal{S}[\![\mathcal{X}]\!]$ by $f(\boldsymbol{v}) = \sum_{i\in I} f_i(\boldsymbol{v})$.

Finally, we can also define the product of polynomials and linear forms as follows:

Definition 2.5. *Let $I \subseteq \mathbb{N}$, let $f, g \in \mathcal{S}[\mathcal{X}]$ be polynomials, and let l be a linear form. The expression flg denotes the mapping $T\colon V \to V \to S$ given by*

$$T(\boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{u}) l(\boldsymbol{v}) g(\boldsymbol{u})\,.$$

We denote by $T|_{\boldsymbol{u}}\colon V \to S$ the linear form given by $T|_{\boldsymbol{u}}(\boldsymbol{v}) = T(\boldsymbol{u}, \boldsymbol{v})$.

2.4 Differential of a Power Series

Recall that in the real case the differential of a function $f\colon V \to \mathbb{R}$ is a mapping $Df\colon V \to \widetilde{V}$ that assigns to every vector $\boldsymbol{v} \in V$ a linear form $Df|_{\boldsymbol{v}}$, the best linear approximation of f at the point $\boldsymbol{v}$. Given the basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ of unit vectors of V, the dual basis $\{\mathrm{dX}_1, \ldots, \mathrm{dX}_n\}$ of $\widetilde{V}$ is defined by $\mathrm{dX}_i(a_1\boldsymbol{e}_1 + \ldots + a_n\boldsymbol{e}_n) = a_i$ for every $a_1, \ldots, a_n \in \mathbb{R}$. Since $\{\mathrm{dX}_1, \ldots, \mathrm{dX}_n\}$ is a basis of $\widetilde{V}$ there are functions $\lambda_1, \ldots, \lambda_n\colon V \to \mathbb{R}$ such that

$$Df|_{\boldsymbol{v}} = \lambda_1(\boldsymbol{v})\,\mathrm{dX}_1 + \cdots + \lambda_n(\boldsymbol{v})\,\mathrm{dX}_n$$

for every $\boldsymbol{v} \in V$ (here λ_i is the partial derivative of f w.r.t. X_i). If for every variable X_i we define $D_{X_i}f\colon V \to \widetilde{V}$ as the mapping that assigns to every vector $\boldsymbol{v}$ the linear form $D_{X_i}f|_{\boldsymbol{v}} = \lambda_i(\boldsymbol{v})\,\mathrm{dX}_i$, then we have $Df = D_{X_1}f + \ldots + D_{X_n}f$.

Definition 2.7 below generalizes the linear forms $D_{X_i}f|_{\boldsymbol{v}}$ to the case in which $\mathbb{R}$ is replaced by an ω-continuous semiring. We start by generalizing the dX_i:

Definition 2.6. *For every $X \in \mathcal{X}$, we denote by* dX *the linear form defined by* $\mathrm{dX}(\boldsymbol{v}) = \boldsymbol{v}(X)$ *for every $\boldsymbol{v} \in V$.*

Definition 2.7. *Let f be a power series and let $X \in \mathcal{X}$ be a variable. The* differential of f w.r.t. X *is the mapping $D_Xf\colon V \to V \to S$ that assigns to every vector $\boldsymbol{v}$ the linear form $D_Xf|_{\boldsymbol{v}}\colon V \to S$ inductively defined as follows:*

$$D_Xf|_{\boldsymbol{v}} = \begin{cases} 0 & \text{if } f \in S \text{ or } f \in \mathcal{X} \setminus \{X\} \\ \mathrm{dX} & \text{if } f = X \\ D_X g|_{\boldsymbol{v}} \cdot h + g \cdot D_X h|_{\boldsymbol{v}} & \text{if } f = g \cdot h \\ \sum_{i \in I} D_X f_i|_{\boldsymbol{v}} & \text{if } f = \sum_{i \in I} f_i. \end{cases}$$

Further, we define the differential of f *as the linear form*

$$Df := \sum_{X \in \mathcal{X}} D_X f.$$

In the real case the differential is used to approximate the value of a differentiable function $f(\boldsymbol{v} + \boldsymbol{u})$ in terms of $f(\boldsymbol{v})$ and $Df|_{\boldsymbol{v}}(\boldsymbol{u})$. The following lemma goes in the same direction.

Lemma 2.1. *Let f be a power series and let $\boldsymbol{v}, \boldsymbol{u}$ be two vectors. We have*

$$f(\boldsymbol{v}) + Df|_{\boldsymbol{v}}(\boldsymbol{u}) \sqsubseteq f(\boldsymbol{v} + \boldsymbol{u}) \sqsubseteq f(\boldsymbol{v}) + Df|_{\boldsymbol{v}+\boldsymbol{u}}(\boldsymbol{u}).$$

3 Solving Systems of Fixed Point Equations

The partial order $\sqsubseteq$ on the semiring $\mathcal{S}$ can be lifted to an order on vectors, also denoted by $\sqsubseteq$, given by $\boldsymbol{v} \sqsubseteq \boldsymbol{v}'$ iff $\boldsymbol{v}(X) \sqsubseteq \boldsymbol{v}'(X)$ for every $X \in \mathcal{X}$.

In the following, let $\boldsymbol{F}$ be a *vector of power series*, i.e., a mapping that assigns to each variable $X \in \mathcal{X}$ a power series $\boldsymbol{F}(X)$. For convenience we denote $\boldsymbol{F}(X)$

by $\boldsymbol{F}_X$. Given a vector $\boldsymbol{v}$, we define $\boldsymbol{F}(\boldsymbol{v})$ as the vector satisfying $(\boldsymbol{F}(\boldsymbol{v}))(X) = \boldsymbol{F}_X(\boldsymbol{v})$ for every $X \in \mathcal{X}$, i.e., $\boldsymbol{F}(\boldsymbol{v})$ is the vector that assigns to X the result of evaluating the power series $\boldsymbol{F}_X$ at $\boldsymbol{v}$. So, $\boldsymbol{F}$ can be seen as a mapping $\boldsymbol{F}\colon V \to V$.

Given a vector of power series $\boldsymbol{F}$, we are interested in the least fixed point of $\boldsymbol{F}$, i.e., the least vector $\boldsymbol{v}$ w.r.t. $\sqsubseteq$ satisfying $\boldsymbol{v} = \boldsymbol{F}(\boldsymbol{v})$.

3.1 Kleene's Iteration Scheme

Recall that a mapping $f\colon \mathcal{S} \to \mathcal{S}$ is *monotone* if $a \sqsubseteq b$ implies $f(a) \sqsubseteq f(b)$, and *ω-continuous* if for any infinite chain $a_0 \sqsubseteq a_1 \sqsubseteq a_2 \sqsubseteq \ldots$ we have $\sup\{f(a_i)\} = f(\sup\{a_i\})$. The definition can be extended to mappings $\boldsymbol{F}\colon V \to V$ from vectors to vectors in the obvious way (componentwise). Then we may formulate the following proposition (cf. [11]).

Proposition 3.1. *Let $\boldsymbol{F}$ be a vector of power series. The mapping induced by $\boldsymbol{F}$ is monotone and continuous. Hence, by Kleene's theorem, $\boldsymbol{F}$ has a unique least fixed point $\mu\boldsymbol{F}$. Further, $\mu\boldsymbol{F}$ is the supremum (w.r.t. $\sqsubseteq$) of the* Kleene sequence *given by $\boldsymbol{\kappa}^{(0)} = \boldsymbol{F}(\boldsymbol{0})$, and $\boldsymbol{\kappa}^{(i+1)} = \boldsymbol{F}(\boldsymbol{\kappa}^{(i)})$.*[2]

Kleene's iteration scheme converges very slowly. Consider for instance the equation $X = aXb + 1$ over the semiring of languages over $\{a, b\}$ (where $0 = \emptyset$ and $1 = \{\lambda\}$). The i-th iteration $\boldsymbol{\kappa}^{(i)}$ is the language $\{a^j b^j \mid j \leq i\}$, so the scheme needs an infinite number of iterations to reach μF. Newton's iteration scheme, introduced below, can be seen as an "acceleration" of Kleene's scheme.

3.2 Newton's Iteration Scheme

Let $\boldsymbol{F}$ be a vector of power series, and $\boldsymbol{v}$ any vector. Then $D\boldsymbol{F}|_{\boldsymbol{v}}$ denotes the mapping $V \to V$ with $\big(D\boldsymbol{F}|_{\boldsymbol{v}}(\boldsymbol{u})\big)_X = D\boldsymbol{F}_X|_{\boldsymbol{v}}(\boldsymbol{u})$. So $D\boldsymbol{F}|_{\boldsymbol{v}}$ can be seen as the evaluation of a mapping $D\boldsymbol{F} : V \to V \to V$ at $\boldsymbol{v}$ (cf. Definition 2.7). Lemma 2.1 then becomes $\boldsymbol{F}(\boldsymbol{v}) + D\boldsymbol{F}|_{\boldsymbol{v}}(\boldsymbol{u}) \sqsubseteq \boldsymbol{F}(\boldsymbol{v} + \boldsymbol{u}) \sqsubseteq \boldsymbol{F}(\boldsymbol{v}) + D\boldsymbol{F}|_{\boldsymbol{v}+\boldsymbol{u}}(\boldsymbol{u})$.

Newton's scheme uses $D\boldsymbol{F}$ to obtain a sequence that converges more quickly to the least fixed point than Kleene's sequence. In order to introduce it we first define the Kleene star of an arbitrary mapping $V \to V$:

Definition 3.1. *Let $\boldsymbol{F}\colon V \to V$ be an arbitrary mapping. The mapping $\boldsymbol{F}^i\colon V \to V$ is inductively defined by $\boldsymbol{F}^0(\boldsymbol{v}) = \boldsymbol{v}$ and $\boldsymbol{F}^{i+1}(\boldsymbol{v}) = \boldsymbol{F}(\boldsymbol{F}^i(\boldsymbol{v}))$. The* Kleene star *of $\boldsymbol{F}$, denoted by $\boldsymbol{F}^*$, is the mapping $\boldsymbol{F}^*\colon V \to V$ given by $\boldsymbol{F}^*(\boldsymbol{v}) = \sum_{i\geq 0} \boldsymbol{F}^i(\boldsymbol{v})$.*

Now we can define Newton's scheme.

Definition 3.2. *Let $\boldsymbol{F}\colon V \to V$ be a vector of power series. We define the* Newton sequence *$(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ as follows:*

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{F}(\boldsymbol{0}) \quad \text{and} \quad \boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + D\boldsymbol{F}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}),$$

where $\boldsymbol{\delta}^{(i)}$ has to satisfy $\boldsymbol{F}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$.

In words, $\boldsymbol{\nu}^{(i+1)}$ is obtained by adding to $\boldsymbol{\nu}^{(i)}$ the result of evaluating the Kleene star of $D\boldsymbol{F}|_{\boldsymbol{\nu}^{(i)}}$ at the point $\boldsymbol{\delta}^{(i)}$.

[2] In [2] we define $\boldsymbol{\kappa}^{(0)} = \boldsymbol{0}$, but $\boldsymbol{\kappa}^{(0)} = \boldsymbol{F}(\boldsymbol{0})$ is slightly more convenient for this paper.

The name "Newton's method" is justified as follows: Consider a univariate equation $X = F(X)$ over $\mathcal{S}_{\mathbb{R}}$ with $F'|_x \in (-1, 1)$ for $x \in [0, \mu F)$. Applying Newton's method as described above to $X = F(X)$, yields $\nu^{(i+1)} = \nu^{(i)} + F'|^*_{\nu^{(i)}}(F(\nu^{(i)}) - \nu^{(i)}) = \nu^{(i)} + \sum_{k\in\mathbb{N}} F'|^k_{\nu^{(i)}}(F(\nu^{(i)}) - \nu^{(i)}) = \nu^{(i)} + \frac{F(\nu^{(i)})-\nu^{(i)}}{1-F'|_{\nu^{(i)}}} = \nu^{(i)} - \frac{G(\nu^{(i)})}{G'|_{\nu^{(i)}}}$. But this is exactly Newton's method for finding a zero of $G(X) = F(X) - X$. This can be generalized to equation systems (see [2, 5]).

The following theorem summarizes the properties of the Newton sequence.

Theorem 3.1. *Let $\boldsymbol{F} \in \mathcal{S}[\![\mathcal{X}]\!]^{\mathcal{X}}$ be a vector of power series. For every $i \in \mathbb{N}$: $\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{F}(\boldsymbol{\nu}^{(i)}) \sqsubseteq \mu\boldsymbol{F} = \sup_j \boldsymbol{\kappa}^{(j)}$.*

In particular, this theorem ensures the existence of a suitable $\boldsymbol{\delta}^{(i)}$ (because $\boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{F}(\boldsymbol{\nu}^{(i)})$, and the convergence of the Newton sequence to the same value as the Kleene sequence. Moreover, since $\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)}$, the Newton sequence converges "at least as fast" as the Kleene sequence.

Example 3.1. In the following examples we set $\mathcal{X} = \{X\}$. Since in this case vectors only have one component, given an element s of a semiring we also use s to denote the vector $\boldsymbol{v}$ given by $\boldsymbol{v}(X) = s$.

Consider the language semiring $\mathcal{S}_{\{a,b\}}$ over the alphabet $\{a, b\}$. One can show that by taking $\boldsymbol{\delta}^{(i)} = \boldsymbol{F}(\boldsymbol{\nu}^{(i)})$ Newton's sequence can be simplified to

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{F}(\boldsymbol{0}) \quad \text{and} \quad \boldsymbol{\nu}^{(i+1)} = D\boldsymbol{F}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{F}(\boldsymbol{\nu}^{(i)})).$$

(1) Consider again the polynomial $f(X) = aXb+1$. As already mentioned above, the Kleene sequence needs ω iterations to reach the fixed point $\{a^n b^n \mid n \geq 0\}$. As a warm-up we show that the Newton sequence converges after one step.

We have $Df|_{\boldsymbol{v}} = a\,\mathrm{dX}\,b$ for every $\boldsymbol{v} \in V$, and so

$$\boldsymbol{\nu}^{(1)} = (a\,\mathrm{dX}\,b)^*(1) = \sum_{j\geq 0} a^j\,\mathrm{dX}(1)b^j = \{a^j b^j \mid j \geq 0\}.$$

The next example shows a more interesting case.

(2) Consider the polynomial $f(X) = aXX + b$. We have:

$$\begin{aligned}
Df|_{\boldsymbol{v}} &= a\boldsymbol{v}(X)\,\mathrm{dX} + a\,\mathrm{dX}\,\boldsymbol{v}(X)\\
\boldsymbol{\nu}^{(0)} &= b\\
\boldsymbol{\nu}^{(1)} &= Df|^*_b(abb + b) = (ab\,\mathrm{dX} + a\,\mathrm{dX}\,b)^*(abb + b)\\
&= L(X \to abX \mid aXb \mid abb \mid b)\\
\boldsymbol{\nu}^{(i+1)} &= Df|^*_{\boldsymbol{\nu}^{(i)}}(f(\boldsymbol{\nu}^{(i)})) = (a\boldsymbol{\nu}^{(i)}\,\mathrm{dX} + a\,\mathrm{dX}\,\boldsymbol{\nu}^{(i)})^*(f(\boldsymbol{\nu}^{(i)}))
\end{aligned}$$

In this case the Newton sequence also needs ω iterations. We shall see in Section 5 that $\boldsymbol{\nu}^{(i)}$ contains the words generated by the grammar $X \to aXX,\ X \to b$ via derivations of index at most $i+1$, i.e., derivations in which no intermediate word contains more than $i + 1$ occurrences of variables.

(3) Consider the same polynomial as in (2), but over a semiring where product is *commutative* (and + is still idempotent). In this case we have:

$$Df|_{\boldsymbol{v}} = a\boldsymbol{v}(X)\,\mathrm{dX}$$

$$\begin{aligned}
\boldsymbol{\nu}^{(0)} &= b \\
\boldsymbol{\nu}^{(1)} &= Df|_b^*(ab^2+b) = (ab\,\mathrm{dX})^*(ab^2+b) = (ab)^*(ab^2+b) = (ab)^*b \\
\boldsymbol{\nu}^{(2)} &= Df|_{\boldsymbol{\nu}^{(1)}}^*(f(\boldsymbol{\nu}^{(1)})) = (a\boldsymbol{\nu}^{(1)})^*(f(\boldsymbol{\nu}^{(1)})) = a(ab)^*b(a((ab)^*b)^2+b) = (ab)^*b
\end{aligned}$$

So the Newton sequence reaches the fixed point at $\boldsymbol{\nu}^{(1)}$. The language $(ab)^*b$ is a regular language having the same Parikh mapping as the context-free language generated by $X \to aXX,\ X \to b$.

4 Derivation Trees and the Newton Iterates

In language theory, given a grammar G, one associates derivations and derivation trees with G. The language $L(G)$ can be seen as the set of all words that can be derived by a derivation tree of G. On the other hand, if G is context-free, then $L(G)$ is the least solution of a fixed point equation $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$ over a language semiring, where the equation $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$ is essentially the production set of G.

In this section we extend the notion of derivation trees to fixed point equations $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$ over any ω-continuous semiring. It will be easy to see that the Kleene iterates $\boldsymbol{\kappa}^{(i)}$ correspond to the derivation trees of height at most i. We will show that the Newton iterates $\boldsymbol{\nu}^{(i)}$ correspond to the derivation trees of *dimension* at most i, generalizing the concept of dimension introduced in [2]. This gives valuable insight into the generalized Newton's method from the previous section and establishes a strong link between two apparently disconnected concepts, one from language theory (finite-index languages, see Section 5) and the other from numerical mathematics (Newton's method).

Definition 4.1 (derivation tree). *Let $\boldsymbol{F}$ be a vector of power series. A derivation tree of $\boldsymbol{F}$ is defined inductively as follows. Let $a_1X_1a_2\cdots X_sa_{s+1}$ be a summand of $\boldsymbol{F}_X$ ($X \in \mathcal{X}$) and let v be a node labelled by*

$$\lambda(v) = (\lambda_1(v), \lambda_2(v)) = (X, a_1X_1a_2\cdots X_sa_{s+1}).$$

Let $t_1, \ldots, t_s$ be derivation trees with $\lambda_1(t_r) = X_r$ ($1 \le r \le s$). Then the tree whose root is v and whose (ordered) children are $t_1, \ldots, t_s$ is a derivation tree.

We identify a derivation tree and its root from now on and often simply write *tree* when we mean *derivation tree*.

Remark to multiplicities. Let $\boldsymbol{F}$ be a system of power series. If, for a variable $X \in \mathcal{X}$, the same monomial m occurs more than once as a summand of $\boldsymbol{F}_X$, and there is a node v in a tree of $\boldsymbol{F}$ s.t. $\lambda_1(v) = X$ and $\lambda_2(v) = m$, then it is not clear "which" summand m of $\boldsymbol{F}_X$ was used at v. But we assume in the following that $\lambda_2(v)$ is a particular *occurrence* of m in $\boldsymbol{F}_X$. Hence, two trees which are equal up to different occurrences are regarded as *different* in the following. However, we do not make that explicit in our definition to avoid notational clutter.

Definition 4.2 (height, yield). *The height $h(t)$ of a derivation tree t is the length of a longest path from the root to a leaf of t. The yield $Y(t)$ of a derivation tree t with $\lambda_2(t) = a_1 X_1 a_2 \cdots X_s a_{s+1}$ is inductively defined to be*

$$Y(t) = a_1 Y(t_1) a_2 \cdots Y(t_s) a_{s+1}.$$

We can characterize the Kleene sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ using the height as follows.

Proposition 4.1. *For all $i \in \mathbb{N}$ and $X \in \mathcal{X}$, we have that $\left(\boldsymbol{\kappa}^{(i)}\right)_X$ is the sum of yields of all derivation trees with $\lambda_1(t) = X$ and $h(t) \leq i$.*

Now we aim at characterizing the Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ in terms of derivation trees. To this end we need use another property of a tree, the tree *dimension*. As does the height, it depends only on the graph structure of a tree.

Definition 4.3 (dimension). *For a tree t, define $dl(t) = (d(t), l(t))$ as follows.*

1. *If $h(t) = 0$, then $dl(t) = (0, 0)$.*
2. *If $h(t) > 0$, let $\{t_1, \ldots, t_s\}$ be the children of t where $d(t_1) \geq \ldots \geq d(t_s)$. Let $d_1 = d(t_1)$. If $s > 1$, let $d_2 = d(t_2)$, otherwise let $d_2 = 0$. Then*

$$dl(t) = \begin{cases} (d_1 + 1, 0) & \text{if } d_1 = d_2 \\ (d_1, l(t_1) + 1) & \text{if } d_1 > d_2. \end{cases}$$

We call $d(t)$ the dimension *of the tree t.*

The following Theorem 4.1 defines a concrete Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ which allows for the desired tree characterization of $\boldsymbol{\nu}^{(i)}$ (cf. Prop. 4.1).

Theorem 4.1. *Let $\boldsymbol{F}$ be a vector of power series. Define the sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ as follows:*

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{F}(\boldsymbol{0}) \quad \text{and} \quad \boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + D\boldsymbol{F}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}),$$

where $\boldsymbol{\delta}^{(i)}_X$ is the sum of yields of all derivation trees t with $\lambda_1(t) = X$ and $dl(t) = (i + 1, 0)$. Then for all $i \geq 0$:

(1) $\boldsymbol{F}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$, so $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ is a Newton sequence as defined in Def. 3.2;
(2) $\boldsymbol{\nu}^{(i)}_X$ is the sum of yields of all derivation trees t with $\lambda_1(t) = X$ and $d(t) \leq i$.

5 Languages with Finite Index

In this section we study fixed point equations $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$ over language semirings. Let $\mathcal{S}_\Sigma$ be the language semiring over a finite alphabet Σ. Let $\boldsymbol{F}$ be a vector of polynomials over $\mathcal{X}$ whose coefficients are elements of Σ. Then, for each $X_0 \in \mathcal{X}$, there is a naturally associated context-free grammar $G_{\boldsymbol{F},X_0} = (\mathcal{X}, \Sigma, P, X_0)$, where the set of productions is $P = \{(X_i \to \alpha) \mid \alpha \text{ is a summand of } \boldsymbol{F}_{X_i}\}$. It is well-known that $L(G_{\boldsymbol{F},X_0}) = \left(\mu\boldsymbol{F}\right)_{X_0}$ (see e.g. [11]). Analogously, each grammar is naturally associated with a vector of polynomials. In the following we use grammars and vectors of polynomials interchangeably.

We show in this section that the approximations $\boldsymbol{\nu}^{(i)}$ obtained from our generalized Newton's method are strongly linked with the *finite-index* approximations of $L(G)$. Finite-index languages have been extensively investigated under different names [13, 8, 14, 7, 6] (see [6] for historical background).

Definition 5.1. *Let G be a grammar, and let D be a derivation $X_0 = \alpha_0 \Rightarrow \cdots \Rightarrow \alpha_r = w$ of $w \in L(G)$, and for every $i \in \{0, \ldots, r\}$ let β_r be the projection of α_r onto the variables of G. The* index *of D is the maximum of $\{|\beta_0|, \ldots, |\beta_r|\}$. The* index-$i$ approximation *of $L(G)$, denoted by $L_i(G)$, contains the words derivable by some derivation of G of index at most i.*

We show that for a context-free grammar G in Chomsky normal form (CNF), the Newton approximations to $L(G)$ coincide with the finite-index approximations.

Theorem 5.1. *Let $G = (\mathcal{X}, \Sigma, P, X_0)$ be a context-free grammar in CNF and let $(\boldsymbol{\nu}^{(i)})_{i \in \mathbb{N}}$ be the Newton sequence associated with G. Then $\big(\boldsymbol{\nu}^{(i)}\big)_{X_0} = L_{i+1}(G)$ for every $i \geq 0$.*

In particular, it follows from Theorem 5.1 that the (first component of the) Newton sequence for a context-free grammar G converges in finitely many steps if and only if $L(G) = L_i(G)$ for some $i \in \mathbb{N}$.

6 Stochastic Context-Free Grammars

In this section we show how the link between finite-index approximations and (the classical version of) Newton's method can be exploited for the analysis of *stochastic context-free grammars (SCFGs)*, a model that combines the language semiring and the real semiring.

A SCFG is a CFG where every production is assigned a probability. SCFGs are widely used in natural language processing and in bioinformatics (see also the example at the end of the section). We use the grammar G^{ex} with productions $X \xrightarrow{1/6} X^6, X \xrightarrow{1/2} X^5, X \xrightarrow{1/3} a$ as running example.

SCFGs can be seen as systems of polynomials over the *direct product* of the semiring $\mathcal{S}_\Sigma$ of languages over Σ and the semiring of non-negative reals $(\mathbb{R}_{\geq 0} \cup \{\infty\}, +, \cdot, 0, 1)$. The system for the grammar G has one polynomial, namely $F_X = (\lambda, \frac{1}{6})X^6 + (\lambda, \frac{1}{2})X^5 + (a, \frac{1}{3})$.

Given an SCFG it is often important to compute the *termination probability* $T(X)$ of a given variable X (see [4, 3] for applications to program verification). $T(X)$ is the probability that a derivation starting at X "terminates", i.e., generates some word. For G^{ex} we have $0.3357037075 < T(X) < 0.3357037076$. It is easy to see that the termination probabilities are given by (the real part of) the least fixed point of the corresponding system of polynomials [4, 5], and that they may be irrational and not representable by radicals (in fact, G^{ex} is an example). Therefore, they must be numerically approximated. This raises the problem that whenever other parameters are calculated from the termination probabilities it is necessary to conduct an error propagation analysis.

A solution to this problem is to replace the original SCFG G by another one, say G', generating "almost" the same language, and for which all termination probabilities are 1. "Almost" means that (1) the probability of the derivations of G that are not derivations of G' is below a given bound, and (2) that the quotient of the probabilities of any two derivations that appear in both G and G' is the same in G and G'. G' can be obtained as follows. Since $\boldsymbol{\delta}^{(i)}$ is uniquely determined by the equation $\boldsymbol{F}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$ over $\mathcal{S}_{\mathbb{R}}$, we know that $\boldsymbol{\nu}^{(K)}$ is the probability of the derivation trees of dimension at most K. Hence, we can approximate the terminating runs by choosing $G' = G_K$, where G_K generates the derivation trees of G of dimension at most K. Assuming that G is in Chomsky normal form, G_K has variables $X_0, X_1, X_{\leq 1}, \ldots, X_K, X_{\leq K}$ for every variable X of G, with $X_{\leq K}$ as axiom. Its productions are constructed so that X_i ($X_{\leq i}$) generates the derivation trees of G of dimension i (at most i) with X as root. For this, G_K has: (i) a production $X_0 \to a$ for every production $X \to a$ of G, (ii) productions[3] $X_i \to Y_{i-1}Z_{i-1}$, $X_i \to Y_i Z_{\leq i-1}$ and $X_i \to Y_{\leq i-1}Z_i$ for every production $X \to YZ$ of G and every $i \in \{1, \ldots, K\}$, and (iii) productions $X_{\leq i} \to X_i$ and $X_{\leq i} \to X_{\leq i-1}$ for every variable X and every $i \in \{1, \ldots, K\}$. It remains to define the probabilities of the productions so that the probability that a tree t is derived from $X_{\leq K}$ in G_K is equal to the conditional probability that t is derived from X in G under the condition that a tree of dimension at most K is derived from X. For this we set $p(X_0 \to a) = \frac{p(X \to a)}{\boldsymbol{\nu}_X^{(0)}}$. For $K > 0$, an induction over the tree dimension shows that we have to choose the remaining probabilities as follows (we omit some symmetric cases):

$$p(X_{\leq K} \to X_K) = \frac{\boldsymbol{\Delta}_X^{(K)}}{\boldsymbol{\nu}_X^{(K)}} \qquad p(X_K \to Y_K Z_{\leq K-1}) = \frac{p(X \to YZ)}{\boldsymbol{\Delta}_X^{(K)}} \boldsymbol{\Delta}_Y^{(K)} \boldsymbol{\nu}_Z^{(K-1)}$$

$$p(X_{\leq K} \to X_{\leq K-1}) = \frac{\boldsymbol{\nu}_X^{(K-1)}}{\boldsymbol{\nu}_X^{(K)}} \qquad p(X_K \to Y_{K-1} Z_{K-1}) = \frac{p(X \to YZ)}{\boldsymbol{\Delta}_X^{(K)}} \boldsymbol{\Delta}_Y^{(K-1)} \boldsymbol{\Delta}_Z^{(K-1)}$$

with $\boldsymbol{\Delta}^{(k)} = \boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}$ for $k > 0$, and $\boldsymbol{\Delta}^{(0)} = \boldsymbol{\nu}^{(0)}$.

The first iterations of the Newton sequence for our running example G^{ex} are

$$\boldsymbol{\nu}^{(0)} = 1/3, \quad \boldsymbol{\nu}^{(1)} = 0.3357024402, \quad \boldsymbol{\nu}^{(2)} = 0.3357037075, \quad \boldsymbol{\nu}^{(3)} = 0.3357037075$$

(up to machine accuracy). In this case we could replace G^{ex} by G_2^{ex} or even G_1^{ex}.

We finish the section with another example. The following SCFG, taken from [10], is used to describe the secondary structure in RNA:

$$L \xrightarrow{0.869} CL \quad L \xrightarrow{0.131} C \quad S \xrightarrow{0.788} pSp \quad S \xrightarrow{0.212} CL \quad C \xrightarrow{0.895} s \quad C \xrightarrow{0.105} pSp.$$

The following table shows the first iterates of the Newton and Kleene sequences for the corresponding system of polynomials.

i	$(\ \boldsymbol{\nu}_L^{(i)}, \quad \boldsymbol{\nu}_S^{(i)}, \quad \boldsymbol{\nu}_C^{(i)}\ )$	$(\ \boldsymbol{\kappa}_L^{(i)}, \quad \boldsymbol{\kappa}_S^{(i)}, \quad \boldsymbol{\kappa}_C^{(i)}\ )$
1	(0.5585, 0.4998, 0.9475)	(0.1172, 0, 0.895)
3	(0.9250, 0.9150, 0.9911)	(0.2793, 0.0571, 0.8973)
5	(0.9972, 0.9968, 0.9997)	(0.3806, 0.1414, 0.9053)

[3] where $X_{\leq 0}$ is identified with X_0.

As we can see, the contribution of trees of dimension larger than 5 is negligible. Here, the Newton sequence converges much faster than the Kleene sequence.

7 Conclusions

We have generalized Newton's method for numerically computing a zero of a differentiable function to a method for approximating the least fixed point of a system of power series over an arbitrary ω-continuous semiring. We have characterized the iterates of the Newton sequence in terms of derivation trees: the i-th iterate corresponds to the trees of dimension at most i. Perhaps surprisingly, in the language semiring the Newton iterates turn out to coincide with the classical notion of finite-index approximations. Finally, we have sketched how our approach can help to analyze stochastic context-free grammars.

References

1. J. Esparza, S. Kiefer, and M. Luttenberger. An extension of Newton's method to ω-continuous semirings. Technical report, Universität Stuttgart, 2007.
2. J. Esparza, S. Kiefer, and M. Luttenberger. On fixed point equations over commutative semirings. In *Proceedings of STACS*, LNCS 4397, pages 296–307, 2007.
3. J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proceedings of LICS 2005*, pages 117–126. IEEE Computer Society Press, 2005.
4. J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *LICS 2004*. IEEE Computer Society, 2004.
5. K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. In *STACS*, pages 340–352, 2005.
6. H. Fernau and M. Holzer. Conditional context-free languages of finite index. In *New Trends in Formal Languages*, pages 10–26, 1997.
7. S. Ginsburg and E. Spanier. Derivation-bounded languages. *Journal of Computer and System Sciences*, 2:228–250, 1968.
8. J. Gruska. A few remarks on the index of context-free grammars and languages. *Information and Control*, 19:216–223, 1971.
9. M. W. Hopkins and D. Kozen. Parikh's theorem in commutative Kleene algebra. In *Logic in Computer Science*, pages 394–401, 1999.
10. B. Knudsen and J. Hein. RNA secondary structure predicition using stochastic context-free grammars and evolutionary history. *Oxford Journals - Bioinformatics*, 15(6):446–454, 1999.
11. W. Kuich. *Handbook of Formal Languages*, volume 1, chapter 9: Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata, pages 609 – 677. Springer, 1997.
12. J.M. Ortega. *Numerical Analysis: A Second Course*. Academic Press, New York, 1972.
13. A. Salomaa. On the index of a context-free grammar and language. *Information and Control*, 14:474–477, 1969.
14. M.K. Yntema. Inclusion relations among families of context-free languages. *Information and Control*, 10:572–597, 1967.