Convergence of Newton's Method over Commutative Semirings[☆,☆☆]

Michael Luttenberger[*], Maximilian Schlund[1,*]

Institut für Informatik, Technische Universität München
Boltzmannstr. 3, 85748 Garching bei München

Abstract

We give a lower bound on the speed at which Newton's method (as defined in (Esparza/Kiefer/Luttenberger, 2007)) converges over arbitrary ω-continuous commutative semirings. From this result, we deduce that Newton's method converges within a finite number of iterations over any semiring which is "collapsed at some $k \in \mathbb{N}$" (i.e. $k = k + 1$ holds) in the sense of (Bloom/Ésik, 2009). We apply these results to (1) obtain a generalization of Parikh's theorem, (2) compute the provenance of Datalog queries, and (3) analyze weighted pushdown systems. We further show how to compute Newton's method over any ω-continuous semiring by constructing a grammar unfolding w.r.t. "tree dimension". We review several concepts equivalent to tree dimension and prove a new relation to pathwidth.

Keywords: Newton's method, polynomial fixed-point equations, semirings, algebraic language theory, Horton-Strahler number

1. Introduction

Fixed-point iteration is a standard approach for solving equation systems of the form $\boldsymbol{X} = F(\boldsymbol{X})$: The naive approach is to compute the sequence $\boldsymbol{X}_{i+1} = F(\boldsymbol{X}_i)$ given some suitable initial approximation $\boldsymbol{X}_0$. In calculus, Banach's fixed-point theorem guarantees that the constructed sequence converges to a solution if F is a contraction over a complete metric space. In computer science, Kleene's fixed-point theorem[2] guarantees convergence if F is an ω-continuous map over a complete partial order. In reference to Kleene's fixed-point theorem,

[☆]This work was partially funded by the DFG project "Polynomial Systems on Semirings: Foundations, Algorithms, Applications"

[☆☆]A preliminary version of this paper was presented at LATA 2013.

[*]Corresponding author

Email addresses: `luttenbe@model.in.tum.de` (Michael Luttenberger), `schlund@model.in.tum.de` (Maximilian Schlund)

[1]Phone: +49(89)289-17232

[2]Depending on literature, this result is also attributed to Tarski [33].

we will call the naive application of fixed-point iteration "Kleene's method" in the following. It is well-known that Kleene's method converges only very slowly in general. Consider the equation $X = 1/2X^2 + 1/2$ over the reals. Kleene's method $\boldsymbol{\kappa}^{(h+1)} = 1/2(\boldsymbol{\kappa}^{(h)})^2 + 1/2$ converges from below to the only solution $x = 1$ starting from the initial approximation $\boldsymbol{\kappa}^{(0)} = 0$. However, it takes 2^{h-3} iterations to gain h bits of precision, i.e. $1 - \boldsymbol{\kappa}^{(2^{h-3})} \leq 2^{-h}$ [14].

Therefore, often approximation schemes (e.g. successive over-relaxation or Newton's method) often do not apply Kleene's method directly to F. Instead they construct from F a new map G to which fixed-point iteration is then applied: Newton's method obtains G from a nonlinear function F by linearization. In above example, $F(X) = 1/2X^2 + 1/2$ is replaced by $G(X) = 1/2X + 1/2$ yielding the sequence $\boldsymbol{\nu}^{(h+1)} = G(\boldsymbol{\nu}^{(h)}) = 1 - 2^{-h}$ for $\boldsymbol{\nu}^{(0)} = 0$, i.e. we get one bit of precision with each iteration.

A system $\boldsymbol{X} = F(\boldsymbol{X})$ where F is given in terms of polynomials over a semiring is called algebraic. In computer science, algebraic systems arise e.g. in the analysis of procedural programs where their least solution describes the set of runs of the program (possibly evaluated under a suitable abstraction). Motivated by the fast convergence of Newton's method over the reals, in [11, 12] (see [13] for an updated version) Newton's method was extended to algebraic systems over ω-continuous semirings: It was shown there that Newton's method always converges monotonically from below to the least solution at least as fast as Kleene's method. In particular, there are semirings where Newton's method converges within a finite number of iterations while Kleene's method does not. This extension of Newton's method found several applications in verification (see e.g. [13, 10, 17]). Independent of the mentioned work, the same extension of Newton's method has been proposed in [27] in the setting of combinatorics which led to new efficient algorithms for random generation of objects.

In this article we give a lower bound on the speed at which Newton's method converges over arbitrary *commutative* ω-continuous semirings. We measure the speed by looking at the number of terms evaluated by Newton's method. To make this more precise, consider the equation $X = aX^2 + c$ in the formal parameters a, c (e.g. over the semiring of formal power series). Its least solution is the series

$$\begin{aligned} B &= \textstyle\sum_{n\in\mathbb{N}} C_n a^n c^{n+1} \\ &= 1c + 1ac^2 + 2a^2c^3 + 5a^3c^4 + 14a^4c^5 + 42a^5c^6 + 132a^6c^7 + 429a^7c^8 + \ldots \end{aligned}$$

with $C_n = \frac{1}{n+1}\binom{2n}{n}$ the n-th Catalan number.

The Kleene approximations $\boldsymbol{\kappa}^{(h+1)} := a\boldsymbol{\kappa}^{(h)}\boldsymbol{\kappa}^{(h)} + c$ of B (modulo commutativity) are always polynomials and one can show that the number of coefficients computed correctly increases by one in each iteration, e.g. the third Kleene approximation has converged in exactly the first three coefficients:

$$\boldsymbol{\kappa}^{(3)} = \mathbf{1}c + \mathbf{1}ac^2 + \mathbf{2}a^2c^3 + 1a^3c^4.$$

By contrast, the Newton approximations $\boldsymbol{\nu}^{(h)}$ are (infinite) power series. Applying the results of this article (see also Example 3.2) we have (again modulo commutativity)

$$\begin{aligned}\boldsymbol{\nu}^{(3)} &= (2a((2ac)^*ac^2+c))^*a((2ac)^*ac^2)^2 \\ &= \mathbf{1}c+\mathbf{1}ac^2+\mathbf{2}a^2c^3+\mathbf{5}a^3c^4+\mathbf{14}a^4c^5+\mathbf{42}a^5c^6+\mathbf{132}a^6c^7+428a^7c^8+\dots\end{aligned}$$

That is, the third Newton approximation has already converged in the first seven coefficients. It follows easily from the characterization [11] of the Newton approximations by "tree-dimension" (see Sec. 3), that the coefficient of $a^n c^{n+1}$ in $\boldsymbol{\nu}^{(h)}$ has converged to C_n if and only if $n+1 < 2^h$, i.e. the number of coefficients which have converged is now roughly doubled in each iteration. In [27] this property is called *quadratic convergence* (see also Ex. 3.2) and is used there to argue that Newton's method allows to efficiently compute a finite number of coefficients of the formal power series representing a generating function.

In program analysis, monomials correspond to runs of a program and for verifying properties it is not sufficient to consider a finite number of runs. Hence we are in general interested in the coefficients of all monomials. We show in Theorem 4.1 for *any* monomial m that either its coefficient in $\boldsymbol{\nu}^{(n+k+1)}$ has already converged or it is bounded from below by 2^{1+2^k} (where n is the number of variables of the given algebraic system). In particular, if the coefficient of the monomial m w.r.t. the power series $\boldsymbol{\nu}^{(n+k+1)}$ is less than 2^{1+2^k}, then we know that it has converged. Using this theorem, we extend Parikh's theorem[3] to multiplicities bounded by a given $k \in \mathbb{N}$ (see Sec. 5.1). From this it follows that the set of monomials whose coefficients have converged in the h-th Newton approximation is Presburger definable. In Sec. 5.2 we apply these results to the problem of computing the provenance of a Datalog query improving on the algorithms proposed in [19]. As a further application of our results, we show in Sec. 5.3 how Newton's method by virtue of Theorem 4.1 can be used to speed up the computation of predecessors and successors in weighted pushdown-systems [28] which has applications e.g. in the analysis of procedural programs or generalized authorization problems in SPKI/SDSI. As a side result, we also show how to compute Newton's method for algebraic systems over arbitrary, also noncommutative, ω-continuous semirings (Sec. 3, Definition 3.3). Finally we remark that the notion of *tree-dimension* has been re-discovered a number of times under various names in different fields during the last 60 years. In Sec. 6 we first survey these notions and then prove a new relation between the dimension and the pathwidth of a tree.

2. Preliminaries

$\mathbb{N}$ denotes the nonnegative integers (natural numbers) with the natural addition, multiplication, and partial order $\leq$. Furthermore, we write $\mathbb{N}_\infty$ for the natural numbers extended by a greatest element ∞. For $k \in \mathbb{N}$ let $\mathbb{N}_k = \{0, 1, \dots, k\}$.

[3] Parikh's theorem states that the commutative image of a context-free grammar is a semi-linear set, i.e. definable by a Presburger formula

A^* ($\mathsf{A}^\oplus$) denotes the free (commutative) monoid generated by A. Elements of A^* are of course written as words over the alphabet A; elements of $\mathsf{A}^\oplus$ are usually written as monomials (in the variables A). $\mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle$ denotes the set of all total functions from A^* to $\mathbb{N}_\infty$.These functions are commonly represented as a formal power series in noncommuting variables A and coefficients in $\mathbb{N}_\infty$. Similarly, elements of $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ are viewed as formal power series in commuting variables A and coefficients in $\mathbb{N}_\infty$. Analogously, for $\mathbb{N}_k\langle\langle \mathsf{A}^* \rangle\rangle$ and $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$.

Semirings. A *semiring* $\langle S, +, \cdot, 0, 1\rangle$ consists of a commutative additively-written monoid $\langle S, +, 0\rangle$ and a multiplicatively-written monoid $\langle S, \cdot, 1\rangle$; both monoids are connected via (1) distributivity of multiplication over addition from both left and right, and (2) 0 as annihilator ($\forall a \in S\colon 0 \cdot a = a$). If addition and multiplication are given by the context we simply write S for $\langle S, +, \cdot, 0, 1\rangle$. A *commutative* semiring is a semiring whose multiplicative monoid is also commutative. $\langle \mathbb{N}, +, \cdot, 0, 1\rangle$ with the canonical addition and multiplication is a commutative semiring. Extending addition and multiplication on $\mathbb{N}$ to $\mathbb{N}_\infty$ so that both remain monotonic and 0 is still an annihilator (i.e. $a + \infty = \infty$, $a \cdot \infty = \infty$ if $a \neq 0$, and $0 \cdot \infty = 0$) yields also a commutative semiring $\langle \mathbb{N}_\infty, +, \cdot, 0, 1\rangle$. For $k \in \mathbb{N}$ define $h_k\colon \mathbb{N}_\infty \to \mathbb{N}_k$ by $h_k(x) := x$ if $x < k$, otherwise $h_k(x) := k$; i.e. h_k "collapses" $\mathbb{N}_\infty$ at k (cf. [2]). We define addition and multiplication on $\mathbb{N}_k$ by $a + b := h_k(a + b)$ and $a \cdot b := h_k(a \cdot b)$ so that h_k becomes a semiring homomorphism. Then $\langle \mathbb{N}_k, +, \cdot, 0, 1\rangle$ is also a commutative semiring. Addition on $\mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle$ resp. $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ is defined pointwise; multiplication is for both defined via the Cauchy product. Then $\langle \mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle, +, \cdot, 0, 1\rangle$ is a (non-commutative) semiring and $\langle \mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle, +, \cdot, 0, 1\rangle$ is a commutative semiring. Analogously for $\mathbb{N}_k\langle\langle \mathsf{A}^* \rangle\rangle$ and $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$.

Recall that a partially ordered set $\langle D, \sqsubseteq, \bot\rangle$ with least element $\bot$ is ω-complete if any ω-chain, i.e. monotonically $\sqsubseteq$-increasing, countable sequence $(a_i)_{i\in\mathbb{N}}$ in D has a least upper bound $\sup_{i\in\mathbb{N}} a_i$ in D w.r.t. $\sqsubseteq$; a function $f\colon D \to D$ is then ω-continuous if $f(\sup_{i\in\mathbb{N}} a_i) = \sup_{i\in\mathbb{N}} f(a_i)$ for any ω-chain $(a_i)_{i\in\mathbb{N}}$ in D. A semiring $\langle S, +, \cdot, 0, 1\rangle$ is ω-continuous if: (1) the *natural order* $a \sqsubseteq b :\Leftrightarrow \exists d\colon a + d = b$ is a partial order on S (with least element 0); (2) $\langle S, \sqsubseteq, 0\rangle$ is ω-complete; and (3) addition and multiplication are ω-continuous in every argument. For every ω-continuous semiring countable summation is well-defined by

$$\sum_{i\in\mathbb{N}} a_i := \sup_{k\in\mathbb{N}} a_1 + \ldots + a_k$$

and behaves as absolutely convergent series over the reals do. In particular we can define the *Kleene star* for any semiring element a in this case by:

$$a^* := \sum_{i\in\mathbb{N}} a^i$$

From the semirings mentioned so far only $\mathbb{N}$ is not ω-continuous. In examples we will also use the ω-continuous semiring over the extended nonnegative reals $\langle [0, \infty], +, \cdot, 0, 1\rangle$ which is obtained by extending the canonical addition and multiplication on $\mathbb{R}$ just as $\langle \mathbb{N}_\infty, +, \cdot, 0, 1\rangle$ was obtained from $\langle \mathbb{N}, +, \cdot, 0, 1\rangle$.

Context-free grammars. A context-free grammar is a triple $G = (\mathcal{X}, \mathsf{A}, R)$ with variables (nonterminals) $\mathcal{X}$, alphabet (formal parameters) A, and (rewrite) rules R. We do not assume a specific start symbol. G is nonexpansive if no variable $X \in \mathcal{X}$ can be rewritten into a sentential form in which X occurs at least twice (see e.g. [29]). G is in quadratic normal form if any rule $X \to u_0 X_1 u_1 \ldots u_{r-1} X_r u_r$ of G satisfies $u_0 u_1 \ldots u_r \in \mathsf{A}^+$, $X_1 X_2 \ldots X_r \in \mathcal{X}^+$, and $r \in \{0, 2\}$.

We slightly deviate from the standard representation of derivation trees: We label the nodes of a derivation tree directly by the corresponding rule (see Example 2.1). For $X \in \mathcal{X}$ a derivation tree of G is an X-tree if its root is labeled by a rule rewriting X. The word represented by a derivation tree is called its yield. The ambiguity of a context-free grammar G w.r.t. to $X \in \mathcal{X}$ is the map $\mathsf{amb}_X \in \mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle$ which assigns to a word $w \in \mathsf{A}^*$ the number of X-trees of G which yield w. Analogously we define the *commutative* ambiguity $\mathsf{camb}_X \in \mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ which assigns to each monomial $m \in \mathsf{A}^\oplus$ the number of X-trees of G which yield a permutation of m. G is unambiguous w.r.t. X if every word has a unique X-tree, i.e. if amb_X takes only values in $\{0, 1\}$.

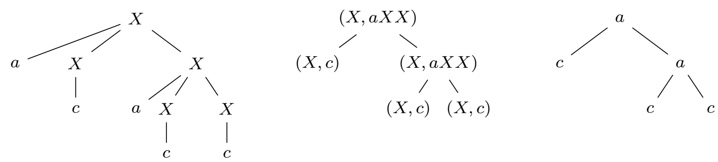
Algebraic systems. With any context-free grammar G we associate the algebraic system $\boldsymbol{X} = F_G(\boldsymbol{X})$ over $\mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle$ (resp. $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ modulo commutativity) consisting of the equations $X = \sum_{(X,\gamma) \in P} \gamma$ (for $X \in \mathcal{X}$). ($\boldsymbol{X}$ is denotes a vector representation of $\mathcal{X}$ w.r.t. some order on $\mathcal{X}$ so that F_G can be viewed as a map on $\mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle^n$ for $n = |\mathcal{X}|$.) As is well-known, the equation system $\boldsymbol{X} = F_G(\boldsymbol{X})$ has always a unique least solution over the most general ω-continuous semiring $\mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle$. This solution is given by the vector $\mathsf{amb} = (\mathsf{amb}_X \mid X \in \mathcal{X})$ (resp. $\mathsf{camb} = (\mathsf{camb}_X \mid X \in \mathcal{X})$ modulo commutativity). amb has the following property [4, 13]: Let $\iota \colon \mathsf{A} \to S$ be any valuation mapping the alphabet into some ω-continuous semiring S. Identify ι with its unique extension to an ω-continuous semiring homomorphism from $\mathbb{N}_\infty\langle\langle \mathsf{A}^* \rangle\rangle$ to S. Finally, let F_G^ι denote the algebraic system over S which we obtain from F_G by substituting every occurrence of $a \in \mathsf{A}$ by $\iota(a)$. Then $\iota(\mathsf{amb})$ is the least solution of $\boldsymbol{X} = F_G^\iota(\boldsymbol{X})$. If S is a commutative semiring the analogous result holds for camb.

Because of this fact any approximation scheme for amb (resp. camb in case of commutativity) translates to an approximation scheme for $\iota(\mathsf{amb})$ over S. It is not hard to see that with any algebraic system $\boldsymbol{X} = F(\boldsymbol{X})$ over some ω-continuous semiring $\langle S, +, \cdot \rangle$ we can associate a context-free grammar and an interpretation ι such that F_G^ι and F have the same least solution over $\langle S, +, \cdot \rangle$. Hence it suffices to study how to approximate amb (resp. camb) for context-free grammars.

By introducing auxiliary variables we can further assume that F only has monomials of degree at most two. Using further the Bekic identity (see e.g. [7]) we can even remove linear terms so that G can always be assumed to be in quadratic normal form (see the example below).

Example 2.1. *Consider the equation $X = 1/4X^2 + 1/2X + 1/4$ over the extended nonnegative reals. We can remove the linear terms without introducing* *auxiliary variables by subtracting* $1/2X$ *on both sides and multiplying afterwards by* $(1-1/2)^{-1} = (1/2)^*$ *which leads to* $X = 1/2X^2 + 1/2$. *(This transformation is possible over any* ω*-continuous semiring as can be shown using the Bekic identity [7].) The simplified equation leads us to the grammar* $G_L\colon X \to aXX \mid c$ *in quadratic normal form with the interpretation* $\iota(a) = \iota(c) = 1/2$.

The language $L(G_L)$ *generated by* G_L *is known as Lukasiewicz language of all proper*[4] *binary trees with binary nodes labeled by* a *and leaves labeled by* c *represented as a word using Polish notation. Below on the left the common depiction of the derivation tree of* $acacc$ *is shown; the middle tree is the representation used in the following which is isomorphic to the binary tree represented by* $acacc$ *shown on the right:*



As G_L *is unambiguous,* amb *enumerates all proper binary trees.* camb *on the other hand is the generating function of proper binary trees, i.e.* $\mathsf{camb}(a^n c^{n+1})$ *is the* n*-th Catalan number* C_n.

$$\mathsf{camb} = c + ac^2 + 2a^2c^3 + 5a^3c^4 + 14a^4c^5 + 42a^5c^6 + 132a^6c^7 + 429a^7c^8 + 1430a^8c^9 + \ldots$$

In particular we have that $\iota(\mathsf{amb}) = \iota(\mathsf{camb}) = 1$ *which is the least (in fact, unique) solution of the original equation* $X = 1/4X^2 + 1/2X + 1/4$.

3. Newton's Method for Context-Free Grammars

In this section, we will first recall Newton's method for context-free grammars resp. algebraic systems over ω-continuous semirings and how the Newton approximations can be characterized by the *dimension* of a tree (resp. term) (Subsection 3.1). We then show an alternative definition of it by unfolding the context-free grammar w.r.t. the dimension (Subsection 3.2, Definition 3.3 and Lemma 3.2). This leads to a description of Newton's method which yields immediately an effective algorithm for computing the Newton approximations over any ω-continuous semiring whose Kleene star is computable.

3.1. Newton's Method and the Dimension of a Tree

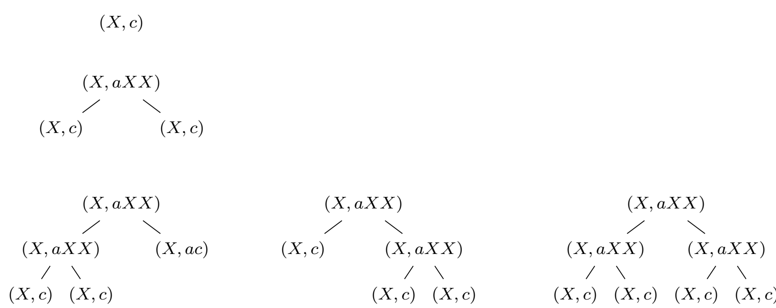
Let us first recall by means of an example that the Kleene approximation $\boldsymbol{\kappa}^{(h)}$ of amb, i.e. $\boldsymbol{\kappa}^{(h+1)} = F_G(\boldsymbol{\kappa}^{(h)})$ with $\boldsymbol{\kappa}^{(0)} = 0$, can be characterized as the sum of the yield of all derivation trees of *height less than h* (see e.g. [11] for a proof).

[4] A binary tree is proper if every node is either binary or nullary.

Example 3.1. *By our convention, we identify the context-free Grammar* $G_L\colon X \to aXX \mid c$ *with the algebraic system* $X = F_{G_L}(X) := aXX + c$ *over* $\mathbb{N}_\infty\langle\langle\{a,c\}^*\rangle\rangle$*. Applying Kleene's method we obtain as the first three approximations (starting with* $\boldsymbol{\kappa}^{(0)} = 0$*) of* amb*:*

$$\begin{aligned}\boldsymbol{\kappa}^{(1)} &= c\\ \boldsymbol{\kappa}^{(2)} &= acc + c\\ \boldsymbol{\kappa}^{(3)} &= aaccacc + aaccc + acacc + acc + c\end{aligned}$$

As G_L *is unambiguous, every word (summand) correspond uniquely to a derivation tree. This correspondence is in this case straightforward as every word can be read as a term in Polish notation (with* a *of arity two, and* c *of arity zero) which encodes a proper binary tree that essentially is the derivation tree (see Example 2.1). For comparison, the derivations trees w.r.t.* G_L *of height less than* 3*:*



In case of a context-free grammar, the same characterization holds but of course we can have multiple derivation trees for the same word so that words will be weighted by elements of $\mathbb{N}_\infty$*.*

In [12, 11] the *dimension* of a derivation tree was introduced in order to give a similar characterization of the approximations obtained by Newton's method via derivation trees (resp. the terms evaluated by Newton's method). As mentioned in the introduction this notion has appeared under different names in various areas, see Sec. 6 for a survey.

A tree $T = (V, E)$ is as always an undirected, acyclic, connected graph with nodes V and edges E. Given a tree $T = (V, E)$ and a node $r \in V$, we write (T, r) for the rooted tree which we obtain by orientating all edges such that they point away from the root r. Any derivation tree is thought of as a rooted tree in the natural way.

Definition 3.1. *Let* (T, r) *be a rooted tree and* $r_1, \ldots, r_s$ *the children of* r*. If* r *has no children (*$s = 0$*), then the* dimension *of* (T, r) *is* $\mathsf{dim}(T, r) := 0$*; otherwise let* $(T_1, r_1), \ldots, (T_s, r_s)$ *be the subtrees we obtain by removing* r*: if there is a unique subtree of maximal dimension* d*, then* $\mathsf{dim}(T, r) := d$*; else* $\mathsf{dim}(T, r) := d + 1$*.*

Set $\mathsf{mindim}(T) := \min_{r \in V} \mathsf{dim}(T, r)$*.*

Note that many other tree invariants (like pathwidth – discussed in Sec. 6) are defined over non-rooted trees. To compare these to the dimension we use $\mathsf{mindim}(T)$ instead. The following lemma shows that the choice of the root has little impact on the dimension.

Lemma 3.1. *Let $T = (V, E)$ be a tree. For any $r \in V$ we have* $\mathsf{dim}(T, r) \leq \mathsf{mindim}(T) + 1$.

Proof. Choose $r' \in V$ such that $\mathsf{dim}(T, r') = \mathsf{mindim}(T)$. Consider then the simple path π connecting r' and r in T. Let $r_1, \ldots, r_k$ be the nodes which do not lie on π but are connected by an edge to some node along this path. Denote by (T_j, r_j) the induced subtrees w.r.t. (T, r'). Then $\mathsf{dim}(T_j, r_j) \leq \mathsf{dim}(T, r') = \mathsf{mindim}(T)$. Changing the root from r' to r, the subtrees (T_j, r_j) and their respective dimensions remain unchanged. Thus $\mathsf{dim}(T, r) \leq 1 + \max_{j \in [k]} \mathsf{dim}(T_j, r_j) \leq 1 + \mathsf{mindim}(T)$. □

Let us call a rooted binary tree where each inner node has exactly two children and all leaves have the same distance to the root a *perfect binary tree*. It is not hard to see that the dimension of a tree can be defined more succinctly as follows (see e.g. [15]):

> Let (T, r) be a rooted tree. Then $\mathsf{dim}(T, r)$ is the height of the largest perfect binary tree which is a minor[5] of (T, r).

An immediate consequence is that the dimension of a tree is always bounded by its height.

Next, let us recall how Newton's method and the tree dimension are related to each other: The original definition of Newton's method for algebraic systems over ω-continuous systems given in [12, 11] is a word-to-word translation of Newton's method over the reals: Over the reals, given a nonlinear map $G \colon \mathbb{R}^n \to \mathbb{R}^n$ and an initial approximation of a zero $\vec{x} \in \mathbb{R}^n$ of G the next approximation is defined to be

$$\vec{x}' := \vec{x} - J_G\big|_{\vec{x}}^{-1} G(\vec{x}).$$

where $J_G\big|_{\vec{x}}$ denotes the Jacobian of G evaluated at $\vec{x}$. Considering the special case that we want to compute a fixed point $X = F(X)$ of a nonlinear map $F \colon \mathbb{R}^n \to \mathbb{R}^n$, we may set $G(X) := F(X) - X$ so that a single Newton step becomes:

$$\vec{x}' := \vec{x} - (J_F\big|_{\vec{x}} - \mathrm{Id})^{-1}(F(\vec{x}) - \vec{x}) = \vec{x} + (\mathrm{Id} - J_F\big|_{\vec{x}})^{-1}(F(\vec{x}) - \vec{x}).$$

As is well-known, sometimes we may write $(\mathrm{Id} - J_F\big|_{\vec{x}})^{-1}$ also as the series $\sum_{k \geq 0} J_F\big|_{\vec{x}}^{k}$ (von Neumann series) which is commonly denoted also as $J_F\big|_{\vec{x}}^{*}$. In order to obtain a definition of Newton's method which also applies to algebraic

[5] A tree T' is a minor of a tree T if T' can be obtained from T by contracting edges and accordingly identifying vertices.

systems over ω-continuous semirings one needs to: (1) generalize the notion of derivative to polynomials over non-commutative semirings, and (2) find a suitable semiring element which represents the difference $F(\vec{x}) - \vec{x}$ since in [12, 11] it is only shown that such a difference always exists but not how to compute it in general. We will not present any further details here as in [12, 11] it was also shown that for any algebraic system over an ω-continuous semiring the terms evaluated by the k-th Newton approximation correspond exactly to the derivation trees of dimension less than k w.r.t. the associated context-free grammar. We therefore can use this result instead as definition of Newton's method for context-free grammars (and thus for algebraic systems).

Definition 3.2. *Let G be a context-free grammar and X a nonterminal. The k-th Newton approximation of amb_X (resp. camb_X) is the formal power series of the yield of exactly those X-trees whose dimension is less than k.*

3.2. Newton's Method as Grammar Unfolding

As stated in the preceding subsection, we can define Newton's method for context-free grammars by using the dimension. In this subsection we will use this result to unfold any context-free grammar G w.r.t. to the dimension into a new context-free grammar $G^{[h]}$ so that the (commutative) ambiguity of $G^{[h]}$ is exactly the h-th Newton approximation of the (commutative) ambiguity of G. One advantage of this new definition is that it allows to effectively compute Newton's method over any ω-continuous semiring for which we can compute the semiring operations and the Kleene star. By contrast, the algebraic definition in [12, 11] requires the user to find in every iteration step a certain semiring element (see the preceding subsection). There, only for particular semirings, e.g. when addition is idempotent, it was shown how to construct these elements.

To unfold general context-free grammars we annotate nonterminals with the superscript (d) resp. $[d]$ to denote that only derivation trees of dimension *exactly* d resp. of dimension *less than* d can be obtained from these nonterminals. In case of a rule whose right-hand side consists of more than two nonterminals we have to distinguish all possible cases which give rise to an increase of dimension:

Consider for simplicity a derivation tree t whose root is labeled by the rule $X \to UVW$ and let t_1, t_2, t_3 be the direct subtrees describing the derivation of U, V, W, respectively – we will abbreviate this by $t = (X, UVW)t_1t_2t_3$ in the following. Further let $d := \mathsf{dim}(t)$. We then have to distinguish the cases

1. $\exists i\colon d = \mathsf{dim}(t_i) \wedge \forall j \neq i\colon \mathsf{dim}(t_i) < d$
2. $\exists i \neq j\colon d - 1 = \mathsf{dim}(t_i) = \mathsf{dim}(t_j) \wedge \forall k \not\in \{i, j\}\colon \mathsf{dim}(t_i) < d - 1$
3. $d - 1 = \mathsf{dim}(t_1) = \mathsf{dim}(t_2) = \mathsf{dim}(t_3)$

when unfolding the corresponding grammar. Generalizing this case distinction to rules whose right-hand side contains more than three nonterminals leads to the following unfolding for general context-free grammars (for $r \in \mathbb{N}$ let $[r] = \{1, 2, \ldots, r\}$):

Definition 3.3. *Let G be a context-free grammar $G = (\mathcal{X}, \mathsf{A}, R)$. Set $\mathcal{X}^\nu := \{X^{[d]}, X^{(d)} \mid X \in \mathcal{X}, d \in \mathbb{N}\} - \{X^{[0]}\}$. The unfolding $G^\nu = (\mathcal{X}^\nu, \mathsf{A}, R^\nu)$ of G is (for $u_1, \ldots, u_r \in \mathsf{A}^*$):*

1. *$X^{[d]} \to X^{(e)} \in R^\nu$ for $0 \le e < d$.*
2. *If $X \to u_0 \in R$, then $X^{(0)} \to u_0 \in R^\nu$.*
3. *If $X \to u_0 X_1 u_1 \in R$, then $X^{(d)} \to u_0 X_1^{(d)} u_1 \in R^\nu$ for every $d \ge 0$.*
4. *If $X \to u_0 X_1 u_1 \ldots u_{r-1} X_r u_r \in R$ with $r > 1$:*
   (a) *For $d = 1$:*
   $$X^{(1)} \to u_0 X_0^{(0)} u_1 \ldots\ldots u_{r-1} X_{r-1}^{(0)} u_r \in R^\nu$$
   (b) *For every $d \ge 1$, and every $J \subseteq [r]$ with $|J| \ge 2$: Set $Z_i := X_i^{(d-1)}$ for all $i \in J$, and $Z_i := X_i^{[d-1]}$ for all $i \in [r] - J$. Then:*
   $$X^{(d)} \to u_0 Z_0 u_1 \ldots\ldots u_{r-1} Z_{r-1} u_r \in R^\nu$$
   (c) *For every $d > 0$, and every $j \in [r]$: Set $Z_j := X_i^{(d)}$ and $Z_i := X_i^{[d]}$ for all $i \in [r] - \{j\}$. Then:*
   $$X^{(d)} \to u_0 Z_1 u_1 \ldots u_{r-1} Z_r u_r \in R^\nu$$

For any given $h \in \mathbb{N}$ let $G^{[h]} = (\mathcal{X}^{[h]}, \mathsf{A}, R^{[h]})$ be the context-free grammar induced by the variables $\{X^{[h]} \mid X \in \mathcal{X}\}$.

Lemma 3.2. *Every $X^{(d)}$-tree ($X^{[d]}$-tree) has dimension exactly (less than) d. There is a yield-preserving bijection between the $X^{(d)}$-trees ($X^{[d]}$-trees) and the X-trees of dimension exactly (less than) d, i.e. the h-th Newton approximation $\boldsymbol{\nu}_X^{(h)}$ of the (commutative) ambiguity of G w.r.t. X is the (commutative) ambiguity of $G^{[h]}$ w.r.t. $X^{[h]}$.*

Proof. Throughout the proof, given a derivation tree t we will write $t = \sigma t_1 \ldots t_r$ to denote that the root of t is labeled by the rule σ and has exactly r children which give rise (from left to right) to the derivation trees $t_1, \ldots, t_r$, i.e. the right-hand side of the rule σ contains exactly r nonterminals where the i-th (from the left) is derived accordingly to t_i. For any node v of t, we write $t|_v$ for the derivation tree we obtain from t by removing all nodes not reachable from v.

Let t be a derivation tree of dimension $\mathsf{dim}(t) = d$. Then $t = \sigma t_1 \ldots t_r$ has at most one child t_c ($c \in [r]$) with $\mathsf{dim}(t) = \mathsf{dim}(t_c)$ by definition of dim. Hence, there is a unique maximal path $v_0 \ldots v_l$ starting at the root v_0 of t. such that (i) $\mathsf{dim}(t) = \mathsf{dim}(t|_{v_l})$ and (ii) either v_l is a leaf of t or every proper subtree of v_l has dimension less than d. Let $\mathsf{dlen}(t) = l$ denote the length of this unique path. Further, we use $\mathsf{dchar}(t) = \{(i, \mathsf{dim}(t'_i)) \mid i \in [r'] \text{ for } t|_{v_l} = \sigma' t'_1 \ldots t'_{r'}\}$ to remember the dimensions of the children of $t|_{v_l}$. ($\mathsf{dchar}(t) = \emptyset$ if v_l is a leaf of t.)

We first show that every $X^{(d)}$-tree has dimension d resp. every $X^{[d]}$-tree has dimension less than d. We proceed by induction on d. In case of $X^{(d)}$-trees, we start from $d = 0$, in case of $X^{[d]}$-trees from $d = 1$:

- First, consider an $X^{(0)}$-tree t. The only rules rewriting $X^{(0)}$ are of the form $X^{(0)} \to u$ or $X^{(0)} \to uY^{(0)}v$ (for $u, v \in \mathsf{A}^*$ and $Y \in \mathcal{X}$). Thus, t is a chain with $0 = \mathsf{dim}(t)$.
- Next, consider an $X^{[1]}$-tree t. By definition of the unfolding, $X^{[1]}$ can only be rewritten to $X^{(0)}$. So $t = (X^{[1]}, X^{(0)})t_1$ for t_1 an $X^{(0)}$-tree, and thus also $\mathsf{dim}(t) = \mathsf{dim}(t_1) = 0$.

For the induction step, we again distinguish the two cases of an $X^{[d]}$-tree and an $X^{(d)}$-tree:

- Let t be an $X^{(d)}$-tree for $d > 0$ where

  $$t = (X^{(d)}, u_0 Z_1 u_1 \dots u_{r-1} Z_r u_r) t_1 \dots t_r$$

  for some $r > 0$. By construction, there is a rule

  $$X \to u_0 X_1 u_1 \dots u_{r-1} X_r u_r \in R (X_i \in \mathcal{X}, u_i \in \mathsf{A}^*)$$

  such that either

  $$\exists i \in [r] \colon Z_i = X_i^{(d)} \wedge \forall j \in [r] - \{i\} \colon Z_j = X_j^{[d]}$$

  or

  $$\exists I \subseteq [r] \colon\ |I| \geq 2 \wedge \forall i \in I \colon Z_i = X_i^{(d-1)} \wedge \forall j \in [r] - I \colon Z_j = X_j^{[d-1]}$$

  Assume first the latter case, and let I be the uniquely determined set of indices with $Z_i = X_i^{(d-1)}$.

  By induction, we already know that every $X_i^{(d-1)}$-tree has dimension exactly $d-1$, and every $X_i^{[d-1]}$-tree has dimension less than $d-1$. As $|I| \geq 2$, we immediately obtain that $\mathsf{dim}(t) = d$ and $\mathsf{dlen}(t) = 0$.

  In the first case, there is a unique index i such that $Z_i = X_i^{(d)}$ while $Z_j = X_j^{[d]}$ for all $j \in [r] - \{i\}$. We thus have $\mathsf{dim}(t) = \mathsf{dim}(t_i)$ and $\mathsf{dlen}(t) = 1 + \mathsf{dlen}(t_i)$. We therefore obtain via induction on $\mathsf{dlen}(t)$ that also $\mathsf{dim}(t) = d$.
- Consider now an $X^{[d]}$-tree t ($d > 1$). Again, by the definition of the unfolding we have $t = (X^{[d]}, X^{(e)})t_1$ for some $e < d$. By induction, we have that the $X^{(e)}$-tree t_1 has dimension exactly e, so that $\mathsf{dim}(t) = \mathsf{dim}(t_1) = e < d$ immediately follows.

We now construct a mapping $\hat{\cdot}$ from the derivation trees of the unfolded grammar to the original grammar G. Informally, $\hat{\cdot}$ contracts edges induced by rules $X^{[d]} \to X^{(e)}$ which choose a concrete dimension $e < d$ (these rules do not occur in the original grammar); then it removes any superscripts from the labels of the tree (so that the resulting tree is a derivation w.r.t. the original grammar). Formally:

- If $t = (X^{[d]}, X^{(e)})t_1$, then $\widehat{t} := \widehat{t_1}$.
- If
$$t = (X^{(d)}, u_0 Z_1 u_1 \dots u_{r-1} Z_r u_r) t_1 \dots t_r,$$
then
$$\widehat{t} := (X, u_0 X_1 u_1 \dots u_{r-1} X_r u_r) \widehat{t_1} \dots \widehat{t_r}$$
where $X_i \in \mathcal{X}$ is the variable from which $Z_i \in \mathcal{X}^{[k]}$ was derived.

Note that $X^{[d]}$ can only be rewritten to $X^{(e)}$ for some $e < d$. So, contracting the corresponding edges can therefore neither change dim nor dchar, but it can decrease dlen. By definition, the rules of $G^{[k]}$ which rewrite the variable $X^{(d)}$ are obtained from the rules of G which rewrite the variable X by only adding superscripts. Hence, by removing the superscripts again any $X^{[d]}$-tree resp. any $X^{(d)}$-tree t is mapped by $\hat{\cdot}$ to an X-tree which has the same yield as t.

We claim that for any $d \geq 0$, $\hat{\cdot}$ is a bijection between the $X^{(d)}$-trees and the X-trees of dimension exactly d. Similarly, we claim that for any $d > 0$, $\hat{\cdot}$ is a bijection between the $X^{[d]}$-trees and X-trees of dimension less than d.

To this end, we show that every X-tree t' (w.r.t. the original grammar) of dimension k there is both a unique $X^{(k)}$-tree t resp. for every $d > k$, a unique $X^{[d]}$-tree t such that $\hat{t} = t'$.

We proceed by induction on the structure of t'.

- Let $t' = (X, u_0)$.

  Then $t = (X^{(0)}, u_0)$ is the unique $X^{(0)}$-tree with $\hat{t} = t$ as we only contract edges corresponding to rules of the form $(X^{[i]}, X^{(j)})$ for $0 \leq j < i$.

  Analogously, $(X^{[d]}, X^{(0)})(X^{(0)}, u_0)$ is the unique $X^{(d)}$-tree which is mapped by $\hat{\cdot}$ onto t': by construction, we need to rewrite $X^{(d)}$ to some $X^{(i)}$ with $i < d$; on the other hand, we already know that all $X^{(i)}$-trees have dimension i, so $i = 0$ has to hold; but as just noted $(X^{(0)}, u_0)$ is the unique $X^{(0)}$-tree which mapped by $\hat{\cdot}$ to t'.

- Let $t' = (X, u_0 X_1 u_1) t'_1$. Further set $k := \mathsf{dim}(t) = \mathsf{dim}(t_1)$.

  In order to construct an $X^{(k)}$-tree t with $\hat{t} = t'$, we need to have $t = (X^{(k)}, u_0 X_1^{(k)} u_1) t_1$ for some $X_1^{(k)}$-tree t_1 with $\hat{t_1} = t'_1$. But by induction t_1 is unique and, thus, so has to be t.

  Analogously, to construct an $X^{[d]}$-tree t for $d > k$, we must have $t = (X^{[d]}, X^{(k)}) t_1$ for some $X^{(k)}$-tree t_1 with $\hat{t_1} = t'$. As just seen this t_1 is unique.

- Let $t' = (X, u_0 X_1 u_1 \dots u_{r-1} X_r u_r) t'_1 t'_2 \dots t'_r$ and $k = \mathsf{dim}(t)$.

  We proceed by induction on $\mathsf{dlen}(t)$. If $\mathsf{dlen}(t) = 0$, i.e. the dimension of t is larger than the dimensions of its subtrees, let $I = \{i \in [r] \mid \mathsf{dim}(t_i) = k-1\}$ be the indices of the subtrees of maximal dimension $k-1$; in particular, t has no subtree of dimension k. Hence, in order to construct an $X^{(k)}$-tree t with $\hat{t} = t'$ we must start with the rule $(X^{(k)}, u_0 Z_1 u_1 \dots u_{r-1} Z_r u_r)$ where

$Z_i = X_i^{(k-1)}$ for $i \in I$, and $Z_j = X_j^{[k-1]}$ for $j \in [r] - I$. By induction, for every X_i-tree t'_i there is a unique $X^{(k-1)}$-tree (if $i \in I$) resp. $X^{[k-1]}$-tree t_i (if $i \in [r] - I$) with $\hat{t_i} = t'_i$. So,

$$t = (X^{(k)}, u_0 Z_1 u_1 \dots u_{r-1} Z_r u_r) t_1 t_2 \dots t_r$$

is the unique $X^{(k)}$-tree with $\hat{t} = t'$.

If $\mathsf{dlen}(t) > 0$, then there is a unique $i \in [r]$ such that $\mathsf{dim}(t) = \mathsf{dim}(t_i) = k$, while $\mathsf{dim}(t_j) < k - 1$ for all $j \in [r] - \{i\}$. Here, it follows analogously that we need to label the root of t by the unique rule $X^{(k)} \to u_0 Z_1 u_1 \dots u_{r-1} Z_r u_r$ where $Z_i = X_i^{(k)}$ and $Z_j = X_j^{[k]}$. Hence, the choice of the subtrees is already uniquely determined again, and so t itself is unique.

Similarly, if we want to construct an $X^{[d]}$-tree t ($d > k$) with $\hat{t} = t'$, we need to start with the rule $X^{[d]} \to X^{(k)}$. Uniqueness then follows from the uniqueness of the $X^{(k)}$-tree.

□

We remark that Definition 3.3 should not be taken literally for implementing Newton's method: For instance, in case 4.(b), the set J explicitly determines the variables which are derived to a term of dimension exactly $d-1$. Thus, when taking literally, the unfolding grows exponentially with the degree of grammar resp. algebraic system, i.e. the maximal degree of any monomial of the given algebraic system. This problem can be avoided by either transforming the given grammar into quadratic normal form before unfolding, or by carefully refining the unfolding by introducing auxiliary variables which allow to combine and re-use shared subexpressions. Both can be implemented in such a way that the size of the unfolded grammar only grows by a constant factor. But as the former is much more practical from our point of view, we refer the reader to [22] for further details on the latter.

Newton's method is closely related to nonexpansive grammars and related notions like quasi-rational languages:

Theorem 3.3. *Let $G = (\mathcal{X}, \mathsf{A}, R)$ be a context-free grammar.*

1. *All Newton approximations of* camb *are rational in* $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$.
2. *Newton's method converges to* amb *(*camb*) of G within a finite number of iterations if and only if G is nonexpansive. If G is nonexpansive, then Newton's method converges within $|\mathcal{X}|$ iterations.*

Proof. The first claim that $\mathsf{camb}_{X^{[k]}}$ is expressible by a weighted rational expression follows directly from the structure of the unfolding of $G^{[k]}$. With $G^{[k]}$ we associate an algebraic system over $\mathbb{N}_\infty\langle\langle \mathbb{N}^{\mathsf{A}} \rangle\rangle$ defined by the equations $X = \sum_{X \to \gamma} \gamma$. The least solution of this system is exactly camb. For $k = 0$ we have only rules which contain at most one variable on the right-hand side. So, the associated algebraic system is linear, in particular right-linear because of commutativity and thus the least solution is expressible as a rational expression. For $k > 0$, solving the associated algebraic system bottom up, we have already determined rational expressions for the variables of the form $X^{[d]}$ and $X^{(d)}$ for $d < k$. By the structure of unfolding, the system is again right-linear w.r.t. to the remaining variables $X^{[k]}$ and $X^{(k)}$. So the claim follows.

For the second claim, assume first that G is expansive. Then there is a derivation of the form $Y \Rightarrow w_0 Y w_1 Y w_2$ for some $Y \in \mathcal{X}$. Obviously, we can use this derivation to construct Y-trees of arbitrary dimension. Hence, $\mathsf{camb}_{Y^{[k]}} < \mathsf{camb}_Y$ for all $k \in \mathbb{N}$. Assume now that G is nonexpansive. The definition of "nonexpansive" can be restated as: In any X-tree $t = \sigma t_1 t_2 \dots t_r$, at most one child contains a node which is labeled by a rule rewriting X. Let $l(t)$ be number of distinct variables Y for which there is at least one node of t which is labeled by a rule rewriting Y. Obviously, $l(t) \leq |\mathcal{X}|$. Induction on $l(t)$ shows that every derivation tree t satisfying this property has dimension less than $l(t)$: For $l(t) = 1$ a tree with this property cannot contain any nodes of arity two or more. Hence, its dimension is trivially zero. For $l(t) > 1$ given such an X-tree $t = \sigma t_1 \dots t_r$ we can find a simple path π leading from the root of t to a leaf which visits all nodes of t which are labeled by a rule rewriting X. Removing π from t we obtain a forest of subtrees each labeled by at most $l(t) - 1$ distinct variables, and each still having above property. Hence, by induction each of these subtrees has dimension less than $l(t) - 1$, and, thus, t has dimension less than $l(t)$. □

If G is expansive, not much can be said regarding convergence speed in the noncommutative setting as illustrated by any unambiguous grammar G: For a given $w \in L(G)$, the least h with $\boldsymbol{\nu}_X^{(h)}(w) = \mathsf{amb}_X(w)$ is simply the dimension of the unique X-tree yielding w. Thus, in the following section we focus on the commutative setting and study the speed at which Newton's method converges to camb by giving a lower bound on all coefficients which have not yet converged.

Example 3.2. *Unfolding G_L (see Ex. 3.1) w.r.t. the dimension gives us $X^{(0)} \to c$, $X^{[1]} \to X^{(0)}$ and for $d > 0$*

$$\begin{array}{rcl} X^{[d]} & \to & X^{(0)} \mid X^{(1)} \mid \dots \mid X^{(d-1)} \\ X^{(d)} & \to & aX^{[d]}X^{(d)} \mid aX^{(d)}X^{[d]} \mid aX^{(d-1)}X^{(d-1)} \end{array}$$

Modulo commutativity, we can deduce from this the following rational expressions for the first few approximations of camb*: $\boldsymbol{\nu}^{(0)} = 0$, $\boldsymbol{\nu}^{(1)} = c$,*

$$\begin{array}{rcl} \boldsymbol{\nu}^{(2)} & = & (2ac)^* ac^2 + c \\ & = & c + ac^2 + 2a^2c^3 + 4a^3c^4 + \dots \\ \boldsymbol{\nu}^{(3)} & = & (2a((2ac)^*ac^2 + c))^* a((2ac)^*ac^2)^2 \\ & = & c + ac^2 + 2a^2c^3 + 5a^3c^4 + 14a^4c^5 + 42a^5c^6 + 132a^6c^7 + 428a^7c^8 + \dots \end{array}$$

We have expanded the series until the first coefficient which differs from camb *(see Ex. 2.1) to exemplify the notion of quadratic convergence introduced in [27]: $\boldsymbol{\nu}^{(h)}$ differs from* camb *in the coefficient of $a^n c^{n+1}$ if and only if $n + 1 \geq 2^h$ as* *any tree with less than 2^h leaves can only have dimension at most $h-1$. This also shows that Newton's method cannot converge faster than quadratic in this sense. Note that although Newton's method converges quadratically w.r.t.* camb, *it only converges linearly over the reals: Consider G_L interpreted as an algebraic system over $\mathbb{R}$ with $\iota(a) = \iota(c) = 1/2$ yielding $X = 1/2X^2 + 1/2$. By also reading the unfolded grammar as an algebraic system and interpreting the alphabet by the same ι we recover the Newton approximations over $\mathbb{R}$: $X^{[0]} = 0$, $X^{(0)} = 1/2$, and for $d > 0$:*

$$X^{[d]} = X^{[d-1]} + X^{(d-1)} \text{ and } X^{(d)} = (1 - X^{[d]})^{-1} \cdot 1/2 \left(X^{(d-1)}\right)^2$$

Induction shows that indeed $\iota(\boldsymbol{\nu}^{(h)}) = X^{[h]} = 1 - 2^{-h}$.

The next example illustrates how the general unfolding of a grammar yields the Newton approximations (i.e. if G not in quadratic normal form).

Example 3.3. *Let G be defined by the productions*

$$X \to aXXXXXX \mid bXXXXX \mid c.$$

The abstract algebraic system associated with this grammar is

$$X = aX^6 + bX^5 + c.$$

Using the interpretation $\iota(a) = 1/6$, $\iota(b) = 1/2$, $\iota(c) = 1/3$, we interpret this abstract system as the concrete system

$$X = 1/6X^6 + 1/2X^5 + 1/3$$

over the ω-continuous semiring $\langle [0,\infty], +, \cdot, 0, 1\rangle$. The least solution μ of this system, i.e. the least nonnegative root of $1/6X^6 + 1/2X^5 - X + 1/3$, can be shown to be neither rational nor expressible using radicals. We may approximate μ by evaluating $\mathsf{camb}_{X^{[k]}}$ under ι. Up to commutativity, the grammar $G^{[k]}$ corresponds to the following algebraic system:

$$\begin{aligned} X^{(0)} &= c & X^{[1]} &= c \\ &\vdots & &\vdots \\ X^{(k)} &= \left(\binom{6}{1} a (X^{[k]})^5 + \binom{5}{1} b (X^{[k]})^4\right) X^{(k)} & X^{[k]} &= \textstyle\sum_{e=0}^{k-1} X^{(e)} \\ &+ \textstyle\sum_{j=2}^{6} \binom{6}{j} a (X^{[k-1]})^{6-j} (X^{(k-1)})^j \\ &+ \textstyle\sum_{j=2}^{5} \binom{5}{j} b (X^{[k-1]})^{5-j} (X^{(k-1)})^j. \end{aligned}$$

From this, rational expressions for $\mathsf{camb}_{X^{[k]}}$ *can easily be obtained:*

$$\begin{aligned}
\mathsf{camb}_{X^{(0)}} &= c \\
\mathsf{camb}_{X^{(1)}} &= (6ac^5 + 5bc^4)^*(ac^6 + bc^5) \\
&\vdots \\
\mathsf{camb}_{X^{(k)}} &= \left(\tbinom{6}{1} a\, \mathsf{camb}^5_{X^{[k]}} + \tbinom{5}{1} b\, \mathsf{camb}^4_{X^{[k]}}\right)^* \\
&+ \textstyle\sum_{j=2}^{6} \tbinom{6}{j} a\, \mathsf{camb}^{6-j}_{X^{[k-1]}}\, \mathsf{camb}^{j}_{X^{(k-1)}} \\
&+ \textstyle\sum_{j=2}^{5} \tbinom{5}{j} b\, \mathsf{camb}^{5-j}_{X^{[k-1]}}\, \mathsf{camb}^{j}_{X^{(k-1)}} .
\end{aligned}$$

where $\mathsf{camb}_{X^{[d]}} = \sum_{k=0}^{d-1} \mathsf{camb}_{X^{(k)}}$ *for all* $d > 0$*.*

Evaluating the first three expressions for $\mathsf{camb}_{X^{[k]}}$ *under* h *we obtain the following approximations of* μ*:*

$$\begin{aligned}
\iota(\mathsf{camb}_{G^{[k]},X^{[0]}}) &= 1/3 \\
\iota(\mathsf{camb}_{G^{[k]},X^{[1]}}) &= 1/3 + (6^{-1}3^{-6} + 2^{-1}3^{-5})(1 - 6 \cdot 6^{-1}3^{-5} - 5 \cdot 2^{-1}3^{-4})^{-1} \\
&= \tfrac{1417}{4221} \approx 0.335702 \\
\iota(\mathsf{camb}_{G^{[k]},X^{[2]}}) &= \tfrac{10981709605561545700033}{32712506178044757018129} \approx 0.335704
\end{aligned}$$

It can be shown by induction on k *that* $\iota(\mathsf{camb}_{X^{[k]}})$ *is exactly the* k*-th approximation obtained by applying Newton's method to* $1/6X^6 + 1/2X^5 - X + 1/3$ *starting at* $X = 0$*.*

4. Rate of Convergence Modulo Commutativity

Let $G = (\mathcal{X}, \mathsf{A}, R)$ be a context-free grammar. In the following n denotes $|\mathcal{X}|$ and $\boldsymbol{\nu}^{(h)}$ denotes the h-th Newton approximation of camb of G, i.e. $\boldsymbol{\nu}^{(h)}_X = \mathsf{camb}_{X^{[h]}}$. We say that two X-trees (w.r.t. G) are *Parikh-equivalent* if they yield the same word up to commutativity. We show that after $n+k+1$ iterations all coefficients which have not converged yet are bounded from below by 2^{2^k+1}.

Theorem 4.1. *For all* $k > 0$ *and* $\boldsymbol{v} \in \mathsf{A}^{\oplus}$*:* $\boldsymbol{\nu}^{(n+k+1)}_X(\boldsymbol{v}) \geq \min(\mathsf{camb}_X(\boldsymbol{v}), 2^{1+2^k})$*.*

Proof. Recall that we labeled the nodes of derivation trees by rules of G. A variable Y is a label of t if there is at least one node which is labeled by a rule rewriting Y. Then by $l(t)$ we denote the number of variables labeling t.

For the proof we need a small lemma from [12]:

Lemma 4.2 ([12])**.** *For every* X*-tree* t *there is a Parikh-equivalent tree* $\tilde{t}$ *of dimension at most* $l(t)$*.*

Assume there is $\boldsymbol{v} \in \mathsf{A}^{\oplus}$ with $\boldsymbol{\nu}^{(n+k+1)}_X(\boldsymbol{v}) < \mathsf{camb}_X(\boldsymbol{v})$. This means there exists some derivation tree t with dimension $\mathsf{dim}(t) \geq n+k+1$ and yield $\boldsymbol{v}$

modulo commutativity. Essentially we show that t witnesses the existence of at least 2^{1+2^k} different, but Parikh-equivalent trees of lower dimension.

We prove the following slightly stronger statement by induction on the number of vertices of t:

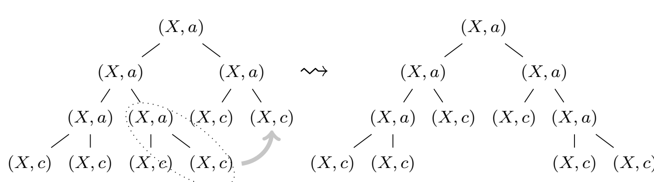
> If $\mathsf{dim}(t) \geq l(t) + k + 1$, then there exist at least 2^{1+2^k} Parikh-equivalent trees of dimension at most $l(t) + k$.

If $|V(t)| = 1$, then $\mathsf{dim}(t) = 0$ whereas $l(t) + k + 1 = k + 2 > 0$, so the claim trivially holds.

If t has a subtree of dimension at least $l(t) + k + 1$ we can apply the inductive hypothesis to every such subtree and thus obtain altogether at least 2^{1+2^k} Parikh-equivalent trees of dimension lower than $\mathsf{dim}(t)$. Therefore, we can restrict ourselves to the case where $\mathsf{dim}(t) = l(t) + k + 1$ and all subtrees have dimension at most $l(t) + k$. Note, that in this case t must have (at least) two subtrees t_1, t_2 of dimension exactly $l(t) + k$. We distinguish two cases:

- Case $l(t_1) < l(t)$ or $l(t_2) < l(t)$: Suppose w.l.o.g. $l(t_1) < l(t)$. We can apply the inductive hypothesis to t_1, since $\mathsf{dim}(t_1) = l(t) + k \geq l(t_1) + k + 1$ and obtain at least 2^{1+2^k} Parikh-equivalent trees of dimension at most $l(t_1) + k$. Then we apply Lemma 4.2 to every *other* subtree of t reducing the dimension of t to at most $l(t) + k$.

- Case $l(t_1) = l(t_2) = l(t)$: Since t_1 has dimension $l(t) + k$ it contains a perfect binary tree of height $l(t) + k$ as a minor. The set of nodes of this minor on level k define 2^k (independent) subtrees of t_1. Each of these 2^k subtrees has height at least $l(t)$, and thus by the Pigeonhole principle contains a path with two variables repeating. We call the partial derivation tree defined by these two repeating variables a *pump-tree*. We relocate any *subset* of these 2^k pump-trees to t_2 which is possible since $l(t_2) = l(t) = l(t_1)$. This changes the subtrees t_1, t_2 into $\tilde{t_1}, \tilde{t_2}$. See the following picture for an illustration of the relocation process (we have two choices for the pump-tree on the left, yielding four possible "remainders").

  Each of these 2^{2^k} choices produces a different tree $\tilde{t}$—the trees differ in the subtree $\tilde{t_1}$. As in the previous case we now apply Lemma 4.2 to every subtree of $\tilde{t}$ except $\tilde{t_1}$ to reduce the dimension of $\tilde{t}$ to at most $\mathsf{dim}(\tilde{t_1}) = l(t) + k$. From this we get at least 2^{2^k} different Parikh-equivalent trees of dimension at most $\mathsf{dim}(\tilde{t_1}) = l(t) + k$. As we can also choose t_2 as the source and t_1 as the destination of the relocation process and apply the same reasoning again, we obtain our desired lower bound of 2^{1+2^k}

□

Remark 4.3. *Although a non-uniform global bound on the coefficients* $\boldsymbol{\nu}^{(n+1+k)}(v)$ *would be desirable (i.e. some bound that depends on* k *and* $|v|$*), the following grammar* H *shows that this cannot be done without taking into account the structure of the grammar:* $H : Y \to BY \mid BX, B \to b, X \to aXX \mid c$. *This grammar contains* G_L*, but any word produced by* Y *can have an arbitrarily long prefix of* b*'s and each such prefix has a unique derivation. Thus* $\mathsf{camb}_Y(b^m a^n c^{n+1}) = \mathsf{camb}_X(a^n c^{n+1}) = C_n$.

We say that an ω-continuous semiring S is *collapsed* at some positive integer k if in S the identity $k = k+1$ holds (see e.g. [2]). For instance, the semirings $\mathbb{N}_k\langle\langle \mathsf{A}^* \rangle\rangle$ and $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$ are collapsed at k. For $k = 1$ the semiring is idempotent.

Corollary 4.4. *Newton's method converges within* $n + \log\log k$ *iterations for any algebraic system with* n *variables over a commutative semiring collapsed at* k.

5. Applications

5.1. Parikh's Theorem for Bounded Multiplicities

Petre [25] defines a hierarchy of power series over $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ and showed that this hierarchy is strict. In particular he shows that Parikh's Theorem does not hold if multiplicities are considered. Here we combine our convergence result and some identities for weighted rational expressions over commutative k-collapsed semirings to show that moving from $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ to $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$ allows us to prove a Parikh-like theorem, i.e. we give a semilinear characterization of camb_G.

In the following, let k denote a fixed positive integer. By Theorem 3.3 and Corollary 4.4 we know that camb_G is rational modulo $k = k+1$. In the idempotent setting ($k = 1$), see e.g. [26] the identities (i) $(x^*)^* = x^*$, (ii) $(x+y)^* = x^*y^*$, and (iii) $(xy^*)^* = 1 + xx^*y^*$ can be used to transform any regular expression into a regular expression in "semilinear normal form" $\sum_{i=1}^{r} w_{i,0} w_{i,1}^* \dots w_{i,l_r}^*$ with $w_{i,j} \in \mathsf{A}^*$. It is not hard to deduce the following identities over $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$ where $x^{<r}$ abbreviates the sum $\sum_{i=0}^{r-1} x^i$. By $\mathsf{supp}(x)$ we denote the characteristic series of the support of x:

Lemma 5.1. *The following identities hold over* $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$*:*

$$
\begin{array}{llcl}
\text{(I1)} & kx & = & k\,\mathsf{supp}(x) \\
\text{(I2)} & (\gamma x)^* & = & (\gamma x)^{<\lceil \log_\gamma k \rceil} + kx^{\lceil \log_\gamma k \rceil} x^* \\
\text{(I3)} & (x^*)^* & = & kx^* \\
\text{(I4)} & (x+y)^* & = & (x+y)^{<k} + x^k x^* + y^k y^* + kxy(x+y)^{\max(k-2,0)} x^* y^* \\
\text{(I5)} & (xy^*)^* & = & 1 + xy^* + x^2 x^* + x^2 y \sum_{0 \le m,j < k-2} \binom{2+m+j}{1+j} x^m y^j \\
 & & + & kx^2 y (x^{\max(k-2,0)} + y^{\max(k-2,0)}) x^* y^*
\end{array}
$$

for γ *any integer greater than one.*

Proof. The proofs are straightforward, and essentially only require to unroll and cut off the power series underlying the Kleene star using the ω-continuity of the Kleene star and the assumption that $k = k+1$. We several times make use of the trivial bound $\binom{a}{b} \geq a$ for $0 < b < a$ on the binomial coefficient.

(I1) $kx = k\,\mathsf{supp}(x)$ is obviously true modulo $k = k+1$.

(I2) $(\gamma x)^* = (\gamma x)^{<\lceil \log_\gamma k \rceil} + k \cdot x^{\lceil \log_\gamma k \rceil} x^*$

This follows from the ω-continuity of the star $(\gamma x)^* = \sum_{n\in\mathbb{N}} (\gamma x)^n$ and the first identity.

(I3) $(x^*)^* = kx^*$

Choose any $w \in \mathsf{supp}((x^*)^*)$. Then w can be factorized into $w = u_1 \dots u_l$ with $u_i \in \mathsf{supp}(x^*)$, i.e., $w \in \mathsf{supp}((x^*)^l)$. Obviously, we then can also find a factorization of w into $l+i$ words for any $i > 0$ as we may add an arbitrary number of neutral elements ε into this factorization. Hence, $w \in \mathsf{supp}((x^*)^{l+i})$ for all $i \geq 0$. So, the coefficient of w in $(x^*)^*$ is $\infty = k$ modulo $k = k+1$.

(I4) $(x+y)^* = (x+y)^{<k} + x^k x^* + y^k y^* + kxy(x+y)^{\max(k-2,0)} x^* y^*$

Proof:

$$
\begin{array}{lcl}
 & & (x+y)^* \\
 & = & (x+y)^{<k} + \sum_{n\geq k} (x+y)^n \\
(xy = yx) & = & (x+y)^{<k} \\
 & + & \sum_{n\geq k} \sum_{j=0}^{n} \binom{n}{j} x^j y^{n-j} \\
 & = & (x+y)^{<k} + \sum_{n\geq k} x^n + y^n \\
 & + & \sum_{j=1}^{n-1} \binom{n}{j} x^j y^{n-j} \\
 & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & \sum_{n\geq k} \sum_{j=1}^{n-1} \binom{n}{j} x^j y^{n-j} \\
(j = i+1, n = m+2) & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & \sum_{m \geq \max(k-2,0)} \sum_{i=0}^{m} \binom{m+2}{i+1} x^{i+1} y^{m-i+1} \\
(\binom{m+2}{i+1} \geq k, (\mathsf{I1})) & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & kxy \sum_{m\geq \max(k-2,0)} \sum_{i=0}^{m} \binom{m}{i} x^i y^{m-i} \\
 & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & kxy(x+y)^{\max(k-2,0)} (x+y)^* \\
((\mathsf{I1})) & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & kxy(x+y)^{\max(k-2,0)}\, \mathsf{supp}((x+y)^*) \\
(\mathsf{supp}((x+y)^*) = \mathsf{supp}(x^*y^*)) & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & kxy(x+y)^{\max(k-2,0)}\, \mathsf{supp}(x^*y^*) \\
((\mathsf{I1})) & = & (x+y)^{<k} + x^k x^* + y^k y^* \\
 & + & kxy(x+y)^{\max(k-2,0)} x^* y^*
\end{array}
$$

(I5)

$$\begin{aligned}(xy^*)^* &= 1 + xy^* + x^2x^* + x^2y\sum_{0\le m,j<k-2}\binom{2+m+j}{1+j}x^my^j \\ &+ kx^2yx^{\max(k-2,0)}x^*y^* + kx^2yx^*y^{\max(k-2,0)}y^*\end{aligned}$$

Proof:

$$\begin{aligned}
& & & (xy^*)^* \\
&(xy^* = y^*x) &=\;& \textstyle\sum_{n\in\mathbb{N}} x^n(y^*)^n \\
& &=\;& 1 + xy^* \\
& &+\;& \textstyle\sum_{n\ge 2} x^n \sum_{l\ge 0}\binom{n+l-1}{l}y^l \\
& &=\;& 1 + xy^* + x^2x^* \\
& &+\;& \textstyle\sum_{n\ge 2, l\ge 1}\binom{n+l-1}{l}x^ny^l \\
&(n = m+2, l = j+1, xy = yx) &=\;& 1 + xy^* + x^2x^* \\
& &+\;& \textstyle x^2y\sum_{m\ge 0, j\ge 0}\binom{2+m+j}{1+j}x^my^j \\
& &=\;& 1 + xy^* + x^2x^* \\
& &+\;& \textstyle x^2y\sum_{\substack{m,j\ge 0\\ m\ge k-2 \vee j\ge k-2}}\binom{2+m+j}{1+j}x^my^j \\
& &+\;& \textstyle x^2y\sum_{0\le m,j<k-2}\binom{2+m+j}{1+j}x^my^j \\
&(k = k+1) &=\;& 1 + xy^* + x^2x^* \\
& &+\;& \textstyle kx^2y\sum_{\substack{m,j\ge 0\\ m\ge k-2 \vee j\ge k-2}} x^my^j \\
& &+\;& \textstyle x^2y\sum_{0\le m,j<k-2}\binom{2+m+j}{1+j}x^my^j \\
&(I1) &=\;& 1 + xy^* + x^2x^* \\
& &+\;& kx^2yx^{\max(k-2,0)}x^*y^* \\
& &+\;& kx^2yx^*y^{\max(k-2,0)}y^* \\
& &+\;& \textstyle x^2y\sum_{0\le m,j<k-2}\binom{2+m+j}{1+j}x^my^j
\end{aligned}$$

□

Consider a rational series $\mathfrak{r} \in \mathbb{N}_k\langle\langle \mathsf{A}^\oplus\rangle\rangle$ represented by the rational expression ρ. The above identities, where (I3), (I4), (I5) generalizes (i), (ii), (iii), respectively, allow us to reduce the star height of ρ to at most one by distributing the Kleene stars over sums and products yielding a rational expression ρ' of the form $\rho' = \sum_{i=1}^{s}\gamma_i w_{i,0}w_{i,1}^*\dots w_{i,l_i}^*$ $(w_{i,j} \in \mathsf{A}^*, \gamma_i \in \mathbb{N}_k)$ which still represents $\mathfrak{r}$ over $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus\rangle\rangle$. By (I1) we know that, if $\gamma_{i,0} = k$, we may replace $w_{i,0}w_{i,1}^*\dots w_{i,l_i}^*$ by its support which is a linear set in $\mathbb{N}^{\mathsf{A}}$.

Theorem 5.2. *Every rational* $\mathfrak{r} \in \mathbb{N}_k\langle\langle \mathsf{A}^\oplus\rangle\rangle$ *can be represented as a finite sum of weighted linear sets, i.e.* $\mathfrak{r} = \sum_{i\in[s]}\gamma_i \,\mathsf{supp}(w_{i,0}w_{i,1}^*\dots w_{i,l}^*)$ *with* $w_{i,j} \in \mathsf{A}^*$ *and* $\gamma_i \in \mathbb{N}_k$.

Proof. We identify a word $w \in \mathsf{A}^*$ with its Parikh vector $\mathsf{c}(w) \in \mathbb{N}^{\mathsf{A}}$. We show that, if $\mathsf{supp}(w_1^*\dots w_l^*) \neq w_1^*\dots w_l^*$ in $\mathbb{N}_k\langle\langle\mathbb{N}^{\mathsf{A}}\rangle\rangle$, then we can split the linear term in a finite sum of weighted linear terms where in each linear term with weight less than k the number of Kleene stars is strictly less than l. Then the result follows inductively.

W.l.o.g. we may assume that each $w_i \neq \varepsilon$, i.e. $\mathsf{c}(w_i) \neq \mathbf{0}$, as $\varepsilon^* = \infty = k$. Denote by $M \in \mathbb{N}^{\mathsf{A}\times l}$ the matrix whose i-th row is given by $\mathsf{c}(w_i)$ (w.r.t. some chosen order on A), and let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_l) \in \mathbb{N}^l$. Then the coefficient $c_{\boldsymbol{v}} := (w_1^* \ldots w_l^*, \boldsymbol{v})$ is exactly the number of solutions over $\mathbb{N}^l$ of the linear equation $\boldsymbol{v} = \boldsymbol{\lambda} M$. If the set $\{\mathsf{c}(w_1), \mathsf{c}(w_2), \ldots, \mathsf{c}(w_l)\}$ is linearly independent, then trivially $c_{\boldsymbol{v}} \leq 1$ and we are done.

Assume thus that the set $\{\mathsf{c}(w_1), \mathsf{c}(w_2), \ldots, \mathsf{c}(w_l)\}$ is linearly dependent, i.e. there is some kernel vector $\boldsymbol{n} = (n_1, \ldots, n_l) \in \mathbb{Z}^l \setminus \{\mathbf{0}\}$. Let $I_+ = \{i \in [l] \mid n_i > 0\}$, $I_- = \{i \in [l] \mid n_i < 0\}$, and $I_0 = \{i \in [l] \mid n_i = 0\}$. As all components of M are nonnegative, $\boldsymbol{n}$ necessarily has a positive and a negative component, i.e. $I_+ \neq \emptyset \neq I_-$. Let $\|n\|_\infty := \max_{i\in[l]} |n_i|$ and $C := \|n\|_\infty \cdot (k-1)$.

Consider now any $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_l) \in \mathbb{N}^l$ with $\lambda_i > C$ for all $i \in I_+$. Then also $\boldsymbol{\lambda} - i\boldsymbol{n} \in \mathbb{N}^l$ for $i = 0, \ldots, k-1$ and trivially $\boldsymbol{v} = \boldsymbol{\lambda} M = (\boldsymbol{\lambda} - i\boldsymbol{n})M$ which implies that $c_{\boldsymbol{v}} \geq k$. If $\lambda_i > C$ for all $i \in I_-$, consider analogously $\boldsymbol{\lambda} + i\boldsymbol{n}$. For $I \in \{I_+, I_-\}$ we split the series $\prod_{i\in I} w_i^*$ into series $\mathfrak{s}_I$ and t_I defined by

$$\mathfrak{s}_I := \prod_{i\in I} (w_i^C w_i^*) \text{ and } \mathsf{t}_I := \sum_{\emptyset \neq J \subseteq I} \prod_{i\in J} w_i^{<C} \prod_{i\in I-J} (w_i^C w_i^*)$$

As discussed above, all positive coefficients of $\mathfrak{s} = \prod_{i\in I}(w_i^C w_i^*)$ (for $I \in \{I_+, I_-\}$) are greater than or equal to k. Hence $\mathfrak{s}_I = k \,\mathsf{supp}(\mathfrak{s}_I)$ over $\mathbb{N}_k\langle\langle \mathbb{N}^{\mathsf{A}} \rangle\rangle$.

$$\begin{aligned}
& w_1^* w_2^* \ldots w_l^* \\
= & \prod_{i\in I_0} w_i^* (\mathfrak{s}_{I_+} + \mathsf{t}_+)(\mathfrak{s}_{I_-} + \mathsf{t}_{I_-}) \\
= & \prod_{i\in I_0} w_i^* (k\mathfrak{s}_{I_+} + \mathsf{t}_+)(k\mathfrak{s}_{I_-} + \mathsf{t}_{I_-}) \\
= & \prod_{i\in I_0} w_i^* \left(\mathsf{t}_{I_+}\mathsf{t}_{I_-} + k(\mathsf{t}_{I_+}\mathfrak{s}_{I_-} + \mathsf{t}_{I_-}\mathfrak{s}_{I_+} + \mathfrak{s}_{I_-}\mathfrak{s}_{I_+})\right) \\
= & \prod_{i\in I_0} w_i^* \left(\mathsf{t}_{I_+}\mathsf{t}_{I_-} + k(\mathsf{t}_{I_+}\mathfrak{s}_{I_-} + \mathsf{t}_{I_-}\mathfrak{s}_{I_+} + 2\mathfrak{s}_{I_-}\mathfrak{s}_{I_+})\right) \\
= & \prod_{i\in I_0} w_i^* \left(\mathsf{t}_{I_+}\mathsf{t}_{I_-} + k\mathfrak{s}_{I_+} \prod_{i\in I_-} w_i^* + k\mathfrak{s}_{I_-} \prod_{i\in I_+} w_i^*\right) \\
= & \prod_{i\in I_0} w_i^* \left(\mathsf{t}_{I_+}\mathsf{t}_{I_-} + k\left(\prod_{i\in I_+} w_i^C + \prod_{i\in I_-} w_i^C\right) \prod_{i\in I_+\cup I_-} w_i^*\right) \\
= & k\left(\prod_{i\in I_+} w_i^C + \prod_{i\in I_-} w_i^C\right) \prod_{i\in[l]} w_i^* + \mathsf{t}_{I_+}\mathsf{t}_{I_-} \prod_{i\in I_0} w_i^*
\end{aligned}$$

It remains to consider the second summand which can be written as a finite sum of products of which each contains at most $|[l] - (J_+ \cup J_-)| \leq l-2$ Kleene stars:

$$\mathsf{t}_{I_+}\mathsf{t}_{I_-}\prod_{i\in I_0} w_i^* = \sum_{\substack{\emptyset\neq J_+\subseteq I_+\\ \emptyset\neq J_-\subseteq I_-}} \prod_{i\in J_+\cup J_-} w_i^{<C} \prod_{i\in(I_+-J_+)\cup(I_--J_-)} w_i^C \prod_{i\in[l]-(J_+\cup J_-)} w_i^*.$$

□

Example 5.1. *The rational expression $\rho = (a+2b)^*$ represents the series $\sum_{i,j\in\mathbb{N}} 2^j a^i b^j$ in $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus\rangle\rangle$. Computing over $N_2\langle\langle \mathsf{A}^\oplus\rangle\rangle$ we may transform ρ as follows:*

$$\begin{array}{lll} & (a+2b)^* & (\mathsf{I4}) \\ = & (a+2b)^{<2} + a^2a^* + (2b)^2(2b)^* + 2a(2b)a^*(2b)^* & (k=k+1) \\ = & \varepsilon + a + 2b + a^2a^* + 2b^2b^* + 2aba^*b^* & (x^* = \sum_{i\in\mathbb{N}} x^i) \\ = & a^* + 2(bb^* + aba^*b^*) & (x^* = \sum_{i\in\mathbb{N}} x^i, \text{comm.}) \\ = & a^* + 2(bb^*a^*) & (\mathsf{I1}) \\ = & 1\,\mathsf{supp}(a^*) + 2\,\mathsf{supp}(bb^*a^*) & \end{array}$$

Corollary 5.3. *For every $k\in\mathbb{N}_\infty$ we can construct a formula of Presburger arithmetic that represents the set $\{\boldsymbol{v}\in\mathbb{N}^{\mathsf{A}} \mid \mathsf{camb}_{G,X}(\boldsymbol{v}) = k\}$.*

Proof. As $\mathsf{c}(L(G,X)) = \mathsf{supp}(\mathsf{camb}_{G,X}) = \{\boldsymbol{v}\in\mathbb{N}^{\mathsf{A}} \mid \mathsf{camb}_{G,X}(\boldsymbol{v}) > 0\}$ is semilinear by Parikh's theorem, it is effectively representable by a formula of Presburger arithmetic, and so is its complement ($k=0$).

Assume thus $1\le k<\infty$ and let $K = k+1$. Then we may compute from $\mathsf{camb}_{X^{[n+\log\log K]}}$ a weighted semilinear representation of camb_X modulo $K = K+1$:

$$\mathsf{camb}_X = \sum_{i=1}^r \gamma_i\,\mathsf{supp}(\boldsymbol{v}_{i,0}\boldsymbol{v}_{i,1}^*\dots\boldsymbol{v}_{i,l_i}^*) \ \text{ with } \gamma_i\in\mathbb{N}_K \text{ and } \boldsymbol{v}_{i,j}\in\mathbb{N}^{\mathsf{A}}.$$

From each term $\mathsf{supp}(\boldsymbol{v}_{i,0}\boldsymbol{v}_{i,1}^*\dots\boldsymbol{v}_{i,l_i}^*)$ we can construct an equivalent Presburger formula F_i. Then $\mathsf{camb}_X(\boldsymbol{v}) = k$ if and only if

$$\boldsymbol{v}\models \exists y_1,\dots,y_r\colon \sum_{i=1}^r \gamma_i y_i = k \wedge \bigwedge_{i=1}^l (F_i(\boldsymbol{v})\to y_i = 1 \wedge \neg F_i(\boldsymbol{v})\to y_i = 0).$$

Finally, let $k=\infty$. As for any $\boldsymbol{v}\in\mathbb{N}^{\mathsf{A}}$ there are only finitely many $w\in\mathsf{A}^*$ with $\mathsf{c}(w)=\boldsymbol{v}$, we have $\mathsf{camb}_{G,X}(\boldsymbol{v})=\infty$ if and only if there is a $w\in\mathsf{A}^*$ with $\mathsf{c}(w)=\boldsymbol{v}$ and $\mathsf{amb}_{G,X}(w)=\infty$. We therefore construct from $G=(\mathcal{X},\mathsf{A},P)$ a context-free grammar $G'=(\mathcal{X}',\mathsf{A},P')$ with $\mathcal{X}\subseteq\mathcal{X}'$ such that $L(G',X) = \{w\in\mathsf{A}^*\mid \mathsf{amb}_{G,X}(w)=\infty\}$. Then $\{\boldsymbol{v}\in\mathbb{N}^{\mathsf{A}}\mid \mathsf{camb}_{G,X}(\boldsymbol{v})=\infty\} = \mathsf{c}(L(G',X))$ and is a semilinear set by Parikh's theorem where the corresponding Presburger formula is again effectively constructible.

We discuss the construction of G' for the sake of completeness: we have $\mathsf{amb}_{G,X}(w)=\infty$ if and only if there are infinitely many X-trees t with $\mathsf{Y}(t)=w$. In particular, for every $h\in\mathbb{N}$ we can find an X-tree t of height at least h with

$\mathsf{Y}(t)$, as there are only finitely many X-trees of bounded height. For instance, choose $h \geq (|w|+1)\,|\mathcal{X}|$ and consider a maximal path $v_0 \dots v_h$ from the root of such a t to a leaf. For all $i = 0 \dots h$ assume $t|_{v_i}$ is an X_i-tree ($X = X_0$). This path then corresponds to a derivation of the form

$$\begin{array}{rll} X = X_0 & \Rightarrow^+ & u_0 X_1 v_0 \\ & \Rightarrow^+ & \dots \\ & \Rightarrow^+ & u_0 \dots u_{h-1} X_h v_{h-1} \dots v_0 \\ & \Rightarrow & u_1 \dots u_{h-1} u_h v_h v_{h-1} \dots v_1 = w \end{array}$$

for suitable $u_i, v_i \in \mathsf{A}^*$. In the sequence $X_0, X_1, \dots, X_h$ color X_i black if $|u_i v_i| = 0$; otherwise color X_i red. Then there are at most $|w|$ red variables in this sequence. In particular, there is a subsequence $X_i, X_{i+1}, \dots, X_{i+|\mathcal{X}|}$ consisting of $1+|\mathcal{X}|$ consecutive black variables, as otherwise $h+1 \leq (|w|+1)\,|\mathcal{X}|$. Hence, the derivation contains a cyclic derivation $Y \Rightarrow^+ Y$.

Therefore compute the set $\mathcal{X}_C = \{X \in \mathcal{X} \mid X \Rightarrow_G^+ X\}$ of cyclic variables as usual, and define G' such that a derivation can only terminate in a word if the derivation visits at least one cyclic variable:

- Set $\mathcal{X}' = \{X, X' \mid X \in \mathcal{X}\}$ with the intended meaning that an unprimed variable still has to be derived into a sentential form containing at least one cyclic variable $Y \in \mathcal{X}_C$.
- Construct P' as follows:
  - If $X \to_G u_0$ for $u_0 \in \mathsf{A}^*$, then $X' \to_{G'} u_0$.
  - If $X \to_G u_0 X_1 u_1 X_2 u_2 \dots u_{r-1} X_r u_r$ for $r > 0$ and $u_i \in \mathsf{A}^*$, then

    $$X' \to_{G'} u_0 X_1' u_1 X_2' u_2 \dots u_{r-1} X_r' u_r$$

    and

    $$\begin{array}{rll} X & \to_{G'} & u_0 X_1 u_1 X_2' u_2 \dots u_{r-1} X_r' u_r \\ X & \to_{G'} & u_0 X_1' u_1 X_2 u_2 \dots u_{r-1} X_r' u_r \\ & \vdots & \\ X & \to_{G'} & u_0 X_1' u_1 X_2' u_2 \dots u_{r-1} X_r u_r \end{array}$$

  - If $X \in \mathcal{X}_C$, then $X \to_{G'} X'$.

By construction, an unprimed variable Y can only be rewritten to a sentential form containing exactly one unprimed variable, except Y is cyclic in G, in which case the rule $Y \to_{G'} Y'$ can also be applied.

Then $w \in L(G', X)$ if and only if there is a derivation $X \Rightarrow_{G'}^+ uYv \Rightarrow_{G'} uY'v \Rightarrow_{G'}^+ w$, as only primed variables can be rewritten to terminal words. By construction, this is equivalent to $X \Rightarrow_G^+ uYv \Rightarrow_G^+ w$ and $Y \in \mathcal{X}_C$, which in turn is equivalent to $\mathsf{amb}_{G,X}(w) = \infty$. $\square$

This corollary can be applied to inclusion testing between two rational series over $\mathbb{N}_k\langle\!\langle \mathsf{A}^\oplus \rangle\!\rangle$ which is relevant e.g. for detecting early convergence of Newton's method, i.e. if $\boldsymbol{\nu}^{(h+1)} = \boldsymbol{\nu}^{(h)}$. Although we know that after $n + \log\log k$ steps the method has converged, in applications (see Sec. 5.2) n could be quite large and the $n + \log\log k$ bound might be too pessimistic.

5.2. Provenance Computation for Datalog

Roughly speaking, provenance is additional information attached to the results of a database query explaining how said results were obtained from the current facts in the database. Provenance information is important e.g. to implement updatable views [16]. Recently, commutative ω-continuous semirings were proposed as provenance annotations where the provenance of unions or projections is modeled by addition of the annotation and joins yield multiplications. Tagging the tuples from the facts in the database allows us to trace back the provenance of the results by solving an algebraic system [19].

For an example consider the binary relation E depicted below (first table). The Datalog query $T(x, y)$:- $E(x, y)$; $T(x, y)$:- $E(x, z), E(z, y)$ computes its transitive closure $T = E^*$ (second table).

$E(X)$	$E(Y)$	
a	b	e_1
b	b	e_2
b	c	e_3
c	d	e_4

$T(X)$	$T(Y)$	
a	b	$X_1 = e_1 + X_1X_4$
a	c	$X_2 = X_1X_5$
a	d	$X_3 = X_1X_6 + X_2X_7$
b	b	$X_4 = e_2 + X_4X_4$
b	c	$X_5 = e_3 + X_4X_5$
b	d	$X_6 = X_4X_6 + X_5X_7$
c	d	$X_7 = e_4$

$$\left.\begin{aligned}
X_1 &= X_4^* e_1 \\
X_2 &= (X_4^*)^2 e_1 e_3 \\
X_3 &= [(X_4^*)^2 + (X_4^*)^3] e_1 e_3 e_4 \\
X_4 &= \textstyle\sum_{n\geq 0} C_n (e_2)^{n+1} \\
X_5 &= X_4^* e_3 \\
X_6 &= (X_4^*)^2 e_3 e_4 \\
X_7 &= e_4
\end{aligned}\right\} \overset{(1=1+1)}{=} \begin{aligned}
& e_1 e_2^* \\
& e_1 e_2^* e_3 \\
& e_1 e_2^* e_3 e_4 \\
& e_2 e_2^* \\
& e_3 e_2^* \\
& e_2^* e_3 e_4 \\
& e_4
\end{aligned}$$

To capture the so called "how-provenance" we tag every tuple in E by a letter from $\Sigma = \{e_1, e_2, e_3, e_4\}$. The provenance of the k-th tuple in T is the value of X_k in the (least) solution (over a suitable semiring) of the algebraic system representing the query. In our example the solution over $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ can be computed by hand and we can also give a very short representation as rational expressions if we assume idempotence of addition $(1 = 1 + 1)$. From the result we can see that the tuple (b, d) can be obtained by a join of (b, c) and (c, d), preceded by any number of joins of (b, b) with itself [6].

Depending on our choice of the semiring we obtain a coarser or finer view on the provenance. As $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ is the commutative semiring, freely generated by A, we can regard it as the universal provenance semiring [19]. However, $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ is in some sense a bad choice for representing solutions, as we cannot do this finitely. Green et al. [19] therefore resort to compute the complete provenance series only if it is finite by enumerating all derivation trees using

[6] More precisely, the operations are joins followed by projections.

Kleene's method essentially; if the power series is an infinite sum they only compute the coefficient for a given monomial.

For many applications, idempotent semirings suffice to capture interesting provenance information. Useful examples are the tropical semiring $\langle \mathbb{N}_\infty, \min, +\rangle$, or the Viterbi-semiring $\langle [0,1], \max, \cdot\rangle$ for probabilistic settings. [19] raised the open question how to compute provenance over the tropical semiring, which can be done by Newton's method as already described in [12]. A useful generalization which is not idempotent is the k-tropical semiring $\mathcal{T}_k$ [24] which was used there for general k-shortest distance computations. This semiring satisfies the identity $k = k+1$, so by our results Newton's method can be used to calculate provenance series over $\mathcal{T}_k$ in $n + \log\log k$ steps.

As already remarked in [19], idempotent semirings are often too coarse an abstraction in a database context where one often considers the so called *bag-semantics* (i.e. we also care about the multiplicities of query results or provenance information). The k-collapsed semirings $\mathbb{N}_k\langle\langle \mathsf{A}^\oplus \rangle\rangle$ are a possible way out of the dilemma that we want to capture the bag-semantics to some extent but cannot use the most general semiring $\mathbb{N}_\infty\langle\langle \mathsf{A}^\oplus \rangle\rangle$ since its elements are not finitely representable in general. Suppose, we want to compute provenance for a recursive query and are satisfied with a power series having coefficients less than $k = 2^{64}$ (i.e. standard 64-bit integers). By Theorem 4.1 we know that Newton's method converges after at most $n + 6$ steps.

5.3. Analysis of Weighted Pushdown Systems

A pushdown system (PDS) $\langle Q, \Gamma, \Delta\rangle$ consists of a finite set of control states Q, a finite set of stack symbols Γ, and a set of rewrite rules $\Delta \subset Q\Gamma \to Q\Gamma^{\leq 2}$. A PDS induces an infinite graph over the $Q\Gamma^*$ of configurations: there is an edge from $q\gamma$ to $q'\gamma'$ if there is a rule $qA \to q'\rho \in \Delta$ such that $\gamma = A\gamma''$ and $\gamma' = \rho\gamma''$. In a weighted PDS each rule carries also as weight an element of a semiring $\langle S, +, \cdot\rangle$. The semiring multiplication is used to *extend* weights from single rules to paths, while addition is used to *combine* the weight of several paths. Such weighted graphs arise e.g. in the analysis of procedural programs [28] or in authorization problems [30]. A central problem is: given a configuration c of the graph, determine for any other configuration c' the weight of all finite paths leading from c to c'.

To solve this problem for arbitrary configurations, one builds a weighted finite automaton whose transitions corresponds to particular runs starting in a configuration pA with a single stack symbol and ending in a configuration $q\varepsilon$ with empty stack. The total weight of these paths is the least solution of an algebraic system over the given semiring S. In the standard approach [28] this algebraic system is solved on the fly while constructing the automaton. For this a work list variant of Kleene's method is used. This approach therefore only works for certain semirings and its running time is directly proportional to the number of iterations needed by Kleene's method to converge which depends on the given semiring. Alternatively, as discussed in [3], one can first build the unweighted automaton, and then solve the algebraic system explicitly. We give an example how Newton's method in combination with Theorem 4.1 allows to speed this up:

Consider the PDS $pA \xrightarrow{a} pAA$, $pA \xrightarrow{b} q$, and $qA \xrightarrow{c} p$ where we have assigned a unique label (weight) to each rule. The PDS encodes a program which always starts in the configuration pA, and we expect it to terminate in $p\varepsilon$. Termination in configuration $q\varepsilon$ is considered to be an error. To simplify debugging, we would like to have, say the k paths from pa to $q\varepsilon$, in particular, these paths should be short. All paths from pa to $p\varepsilon$ resp. $q\varepsilon$ are described by the grammar

$$X \to aXX \mid aYc \text{ and } Y \to aXY \mid b.$$

We first determine the length of the k shortest paths. To this end, we can collapse the alphabet to a singleton, say $\iota(a) = \iota(b) = \iota(c) = z$, and compute the commutative ambiguity of the resulting grammar modulo $k = k + 1$. The coefficient of z^i in camb_X resp. camb_Y then tells us, how many paths (up to k) of length i lead from pA to $p\varepsilon$ resp. $q\varepsilon$. For simplicity, assume $k = 4$. By virtue of Theorem 4.1 we know that at most $n+1+\log\log k = 4$ Newton iterations suffice to compute camb modulo $k = k+1$. (For comparison, Kleene's method can take up to $\mathcal{O}(k)$ iterations, consider e.g. $pA \to pAA, pA \to qA, qA \to q\varepsilon$.) This gives us: $\mathsf{camb}_X = z^3 + 2z^7 + 2z^{11} + \mathcal{O}(z^{12})$ and $\mathsf{camb}_Y = z + z^5 + 3z^9 + \mathcal{O}(z^{10})$. The partial expansion of camb_Y tells us the four shortest paths from pA to $q\varepsilon$ consist of one path of length 1, one path of length 5, and two paths of length 9 each. For constructing the actual paths, these lengths allows us to early discard paths which cannot contribute to the k shortest paths. For instance, we can now apply Kleene's method and discard after each iteration any path of length at least 10. This will take 5 iterations until we have discovered enough paths.

On the other hand, by virtue of Theorem 4.1 we know that we discover a sufficient number of paths of any given length l when considering only derivation trees of low dimension. Consider e.g. the restriction of the grammar to derivation trees of dimension at most one (see Def. 3.3). Dimension 0 gives us the shortest path b from pA to $q\varepsilon$. The unfolding of the grammar to dimension exactly 1 is:

$$\begin{array}{lcl} X^{(1)} & \to & aX^{(1)}abc \mid aabcX^{(1)} \mid aabcabc \mid a\hat{Y}^{(1)}c \\ \hat{Y}^{(1)} & \to & aX^{(1)}b \mid aabc\hat{Y}^{(1)} \mid aabcb \end{array}$$

Applying Kleene iteration now to this unfolded grammar, we only enumerate trees of dimension 1 with at most 9 leaves. Within two iterations we obtain enough paths, namely $aabcb$, $(aabc)^2b$, $aaaabcbcb$, and $aaabcabcb$, to answer the query. Note that a path of the form $(aabc)^hb$ has a derivation tree of dimension 1, but of height $h+1$, i.e. it takes $h+1$ Kleene iterations on the original grammar to discover this path. By increasing k, the gap between Newton's method and Kleene's method can thus be made arbitrarily large.

6. Tree Dimension: Related Work and Concepts

Here, we want to give a comprehensive survey of related work we have become aware of in the past few years. These also introduce and study the concept of dimension directly or investigate closely related ideas.

To the best of our knowledge it has first been mentioned in hydrology around 1945 by Horton and Strahler [20, 31] where it is called *(Horton-)Strahler number* and is used to define the stream size of a hierarchy of tributaries. In 1958, Ershov [9] introduced the same idea as *register number* for measuring the minimal number of registers needed to evaluate an arithmetic expression. For binary trees, the combinatorial properties of the register number were studied in detail by Flajolet et al. [15] and Kemp [21]. In 1978, Ehrenfeucht et al. introduced the same concept for derivation trees w.r.t. ET0L systems in [8] where it was called *tree-rank*.

Closely related to the dimension of a tree are also two other well-known notions: the index of a derivation and the pathwidth of a graph.

Recall that the index (see e.g. [35] or [18]) of a derivation w.r.t. a (not necessarily context-free) grammar is the largest number of nonterminals occurring in any sentential form within the derivation. For a context-free grammar in Chomsky normal form, it is easy to see that a derivation tree of dimension d can be serialized into a derivation of minimal index $d+1$: simply serialize the derivation tree recursively by processing subtrees of minimal dimension first; consequently, if a word can be obtained by a derivation of minimal index i, then it has at least one derivation tree of minimal dimension $i-1$ (see e.g. [11]).[7]

Meggido et al. introduced in [23] (1981) the concept of search number of a tree: it is the minimal number of police officers required to capture a fugitive when police officers may move along edges from one node to another, while the fugitive may pass from one edge to an incident one as long as the common vertex is not blocked by a police officer; the fugitive is captured when he cannot move anymore. When lifted to general graphs, the concept of search number is equivalent to the pathwidth (see e.g. [1]).

As expected, the index of a derivation immediately yields an upper bound on the pathwidth of the derivation tree (see e.g. [6]), hence, so does the dimension; but as the pathwidth is independent of the root of a (derivation) tree, it does not coincide with the index in general. We can show the following relation between the pathwidth and the dimension of a tree:

Proposition 6.1. *For any tree $T=(V,E)$:*

$$\mathsf{pw}(T)-1 \leq \mathsf{mindim}(T) \text{ and } \mathsf{mindim}(T) \leq 2\mathsf{pw}(T)$$

Proof. For convenience, we recall the formal definition of pathwidth first:

Definition 6.1. *Let $G=(V,E)$ be an undirected graph with nodes V and edges E. A* path decomposition *of G is a finite sequence $(B_1,B_2,\dots,B_s)$ of sets $B_i \subseteq V$ satisfying the following two properties:*

1. $V=\bigcup_{i\in[s]} B_i$.

[7]If the grammar is not in Chomsky normal form, then the relation between the dimension of a derivation tree and the index of a derivation is less strict: Let r be the maximal number of nonterminals occurring on the right-hand side of any production; then a derivation tree of dimension d can be serialized into a derivation of index at most $(r-1)\cdot d+1$.

2. *For all* $\{u, v\} \in E$ *there is an* $i \in [s]$ *such that* $\{u, v\} \subseteq B_i$.
3. *For all* $i \le j \le k$ *it holds that* $B_i \cap B_k \subseteq B_j$.

The width *of a path decomposition* $(X_i)_{i\in[s]}$ *is* $-1+\max_{i\in[s]}|X_i|$. *The* pathwidth $\mathsf{pw}(G)$ *of* G *is the minimum width taken over all path decompositions of* G.

We first show that $\mathsf{pw}(T) \le 1 + \mathsf{mindim}(T)$:

Choose any $r \in V$ such that $\mathsf{dim}(T, r) = \mathsf{mindim}(T)$. If $\mathsf{dim}(T, r) = 0$, then (T, r) is by definition a chain and $\mathsf{pw}(T) \le 1$ immediately follows. Thus assume $\mathsf{dim}(T, r) > 0$. Let $T_1, \ldots, T_k$ be the connected components of $T \setminus \{r\}$, and $r_1, \ldots, r_k$ the children of r in (T, r). For $(B_i^{(j)})_{i\in[s_j]}$ a path decomposition of T_j, we can construct the following path decomposition of T (to be read from left to right):

$$B_1^{(1)} \cup \{r\}, \ldots, B_{s_1}^{(1)} \cup \{r\}, B_1^{(2)} \cup \{r\}, \ldots, B_{s_{k-1}}^{(k-1)} \cup \{r\},\ \{r, r_k\}, B_1^{(k)}, \ldots, B_{s_k}^{(k)}.$$

By definition of path decomposition, every r_j is covered by $(B_i^{(j)})_{i\in[s_j]}$, so that by adding r to the first $k-1$ decompositions all the edges $\{(r, r_1), \ldots, (r, r_{k-1})\}$ are covered. After covering also the edge (r, r_k), only edges and nodes of T_k remain to be covered, so that the decomposition $(B_i^{(k)})_{i\in[s_k]}$ suffices. Hence, $\mathsf{pw}(T) \le 1 + \max_{i\in[k]} \mathsf{pw}(T_i)$. In particular, if the maximum $\max_{i\in[k]} \mathsf{pw}(T_i)$ is uniquely determined, wlog. by T_k, then $\mathsf{pw}(T) \le \max_{i\in[k]} \mathsf{pw}(T_i)$.

Now by induction $\mathsf{pw}(T_i) \le 1+\mathsf{mindim}(T_i) \le 1+\mathsf{dim}(T_i, r_i)$. By definition, either (a) $\mathsf{dim}(T, r) = \max_{i\in[k]} \mathsf{dim}(T_i, r_i)$ or (b) $\mathsf{dim}(T, r) = 1+\max_{i\in[k]} \mathsf{dim}(T_i, r_i)$. In case (a) the maximum is unique so that for at most one subtree T_i, say T_k, we have $\mathsf{pw}(T_k) = 1 + \mathsf{dim}(T, r)$, while for the remaining components $\mathsf{pw}(T_i) < 1 + \mathsf{dim}(T, r)$. Thus $\mathsf{pw}(T) \le 1 + \mathsf{dim}(T, r) = 1 + \mathsf{mindim}(T)$ by choice of r. In case (b) we have $\mathsf{pw}(T_i) \le 1 + \mathsf{dim}(T_i, r_i) \le \mathsf{dim}(T, r)$, and thus $\mathsf{pw}(T) \le 1 + \mathsf{dim}(T, r)$, too.

It remains to show that $\mathsf{mindim}(T) \le 2\mathsf{pw}(t)$ (the proof is similar to the proof of Lemma 9 in [32]):

Set $w := \mathsf{pw}(t)$. If $w = 0$, then t consists of a single node and $\mathsf{mindim}(T) = 0$.

Thus assume $w > 0$ and let $(B_i)_{i\in[s]}$ be some path decomposition of minimal width w. Choose any $v_1 \in B_1$ and $v_s \in B_s$, and let π be the unique simple path connecting v_1, v_s in T. As before, let $r_1, \ldots, r_k$ be the nodes of T which are not located on π but connected to some node along π by an edge; then denote by (T_i, r_i) the corresponding subtrees w.r.t. (T, v_1). Since every edge of π is covered by at least one B_i $(B_i - \{v_1, \ldots, v_s\})_{i\in[s]}$ is a path decomposition of width at most $w - 1$ for every tree T_i, i.e., $\mathsf{pw}(T_i) \le w - 1$. By induction, we therefore have

$$\mathsf{dim}(T_i, r_i) \le 1 + \mathsf{mindim}(T_i) \le 1 + 2\mathsf{pw}(T_i) \le 1 + 2(w - 1).$$

Thus

$$\mathsf{mindim}(T) \le \mathsf{dim}(T, v_1) \le 1 + \max_{i\in[s]} \mathsf{dim}(T_i, r_i) \le 2w = 2\mathsf{pw}(T). \qquad \square$$

These bounds are sharp up to ± 1: For the left inequality consider the perfect ternary tree P_3^h of height h: No matter which node we pick as root the resulting rooted tree will have two P_3^{h-1} subtrees so that induction immediately yields that $\mathsf{mindim}(P_3^h) = h$. On the other hand we have $\mathsf{pw}(P_3^h) = h$, see e.g. [32].

Similarly for the right inequality we use P_2^h the perfect binary tree P_2^h of height h. Here we have $\mathsf{mindim}(P_2^h) = h - 1$ and $\mathsf{pw}(P_2^h) \leq \lceil \frac{h}{2} \rceil$:

The upper bound is obviously true for $h = 0, 1$. Inductively, one can construct a path decomposition of P_2^h of width $w+1$ using decompositions $(X_i)_{i\in[s]}$ of P_2^{h-2} of width w as follows (we identify the nodes of P_2^h with the strings from $\{0,1\}^{\leq h}$):

$$(00X_i \cup \{0\})_{i\in[s]}, (01X_i \cup \{0\})_{i\in[s]}, \{\varepsilon, 0\}, \{\varepsilon, 1\}, (10X_i \cup \{1\})_{i\in[s]}, (11X_i \cup \{1\})_{i\in[s]}.$$

Using a theorem on pathwidth obstructions, one can show that $\mathsf{pw}(P_2^h) \geq \lceil \frac{h}{2} \rceil$ (see e.g. [5]) so in fact this path decomposition is optimal. Our proposition yields the lower bound $\mathsf{pw}(P_2^h) \geq \frac{h-1}{2} = \lfloor \frac{h}{2} \rfloor$, so it is off by 1 for perfect binary trees of odd height.

7. Future Work

For proper binary trees, [15] provide a closed form for the number of trees with n leaves and dimension less than h, i.e. for $\boldsymbol{\nu}^{(h)}(a^{n-1}c^n)$. They show that the expected dimension of a random binary tree with n leaves is tightly concentrated around $1/2 \log_2 n$. This implies a much faster convergence of Newton's method in the case of G_L. We conjecture that a similar result can also be derived for arbitrary context-free grammars.

In the idempotent case, we can use a result of [34] to obtain for a given context-free grammar G a Presburger formula of size linear in $|G|$ defining its Parikh image. It would be interesting, if one could generalize this procedure to semirings collapsed at k as the result of Sec. 5.1 in general leads to very large expressions.

[1] Dan Bienstock, Neil Robertson, Paul Seymour, and Robin Thomas. Quickly Excluding a Forest. *Journal of Combinatorial Theory, Series B*, 52(2):274 – 283, 1991.

[2] S. L. Bloom and Z. Ésik. Axiomatizing Rational Power Series over Natural Numbers. *Inf. Comput.*, 207(7):793–811, 2009.

[3] A. Bouajjani, J. Esparza, and T. Touili. A Generic Approach to the Static Analysis of Concurrent Programs with Procedures. *Int. J. Found. Comput. Sci.*, 14(4):551–, 2003.

[4] S. Bozapalidis. Equational Elements in Additive Algebras. *Theory Comput. Syst.*, 32(1):1–33, 1999.

[5] Kevin Cattell, Michael J. Dinneen, and Michael R. Fellows. A Simple Linear-Time Algorithm for Finding Path-Decompositions of Small Width. *Information Processing Letters*, 57(4):197 – 203, 1996.

[6] Michal Chytil and Burkhard Monien. Caterpillars and Context-Free Languages. In *STACS*, pages 70–81, 1990.

[7] M. Droste, W. Kuich, and H. Vogler. *Handbook of Weighted Automata*. Springer, 2009.

[8] Andrzej Ehrenfeucht, Grzegorz Rozenberg, and Dirk Vermeir. On ET0L Systems with Finite Tree-Rank. *SIAM J. Comput.*, 10(1):40–58, 1981.

[9] A. P. Ershov. On Programming of Arithmetic Operations. *Comm. ACM*, 1(8):3–9, 1958.

[10] J. Esparza, P. Ganty, S. Kiefer, and M. Luttenberger. Parikh's Theorem: A Simple and Direct Automaton Construction. *Inf. Process. Lett.*, 111(12):614–619, 2011.

[11] J. Esparza, S. Kiefer, and M. Luttenberger. An Extension of Newton's Method to ω-Continuous Semirings. In *DLT*, volume 4588 of *LNCS*, pages 157–168. Springer, 2007.

[12] J. Esparza, S. Kiefer, and M. Luttenberger. On Fixed Point Equations over Commutative Semirings. In *STACS*, volume 4393 of *LNCS*, pages 296–307. Springer, 2007.

[13] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian Program Analysis. *J. ACM*, 57(6):33, 2010.

[14] K. Etessami and M. Yannakakis. Recursive Markov Chains, Stochastic Grammars, and Monotone Systems of Nonlinear Equations. *J. ACM*, 56(1), 2009.

[15] P. Flajolet, J.-C. Raoult, and J. Vuillemin. The Number of Registers Required for Evaluating Arithmetic Expressions. *Theor. Comput. Sci.*, 9:99–125, 1979.

[16] J. Nathan Foster and Grigoris Karvounarakis. Provenance and Data Synchronization. *IEEE Data Eng. Bull.*, 30(4):13–21, 2007.

[17] P. Ganty, R. Majumdar, and B. Monmege. Bounded Underapproximations. In *CAV*, pages 600–614, 2010.

[18] S. Ginsburg and E. Spanier. Derivation-Bounded Languages. *Journal of Computer and System Sciences*, 2:228–250, 1968.

[19] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance Semirings. In *PODS*, pages 31–40, 2007.

[20] R. E. Horton. Erosional Development of Streams and Their Drainage Basins: Hydro-Physical Approach to Quantitative Morphology. *Geol. Soc. Am. Bull.*, 56(3):275–370, 1945.

[21] R. Kemp. The Average Number of Registers Needed to Evaluate a Binary Tree Optimally. *Acta Informatica*, 11:363–372, 1979.

[22] Michael Luttenberger and Maximilian Schlund. Regular Expressions for Provenance (extended version). In *6th Workshop on the Theory and Practice of Provenance, TaPP'14, Cologne, Germany, June 12-13, 2014*, 2014.

[23] N. Megiddo, S. Louis Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The Complexity of Searching a Graph (Preliminary Version). In *FOCS*, pages 376–385, 1981.

[24] M. Mohri. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *J. Autom. Lang. Comb.*, 7(3):321–350, 2002.

[25] I. Petre. Parikh's Theorem Does not Hold for Multiplicities. *J. Autom. Lang. Comb.*, 4(1):17–30, 1999.

[26] D. L. Pilling. Commutative Regular Equations and Parikh's Theorem. *J. London Math. Soc.*, pages 663–666, 1973.

[27] C. Pivoteau, B. Salvy, and M. Soria. Algorithms for Combinatorial Structures: Well-Founded Systems and Newton Iterations. *J. Comb. Theory, Ser. A*, 119(8):1711–1773, 2012.

[28] T. W. Reps, S. Schwoon, S. Jha, and D. Melski. Weighted Pushdown Systems and their Application to Interprocedural Dataflow Analysis. *Sci. Comput. Program.*, 58(1-2):206–263, 2005.

[29] G. Rozenberg and A. Salomaa. *Handbook of Formal Languages: Word, Language, Grammar*, volume 1. Springer Verlag, 1997.

[30] S. Schwoon, S. Jha, T. W. Reps, and S. G. Stubblebine. On Generalized Authorization Problems. In *CSFW*, pages 202–218, 2003.

[31] A. N. Strahler. Hypsometric (Area-Altitude) Analysis of Erosional Topology. *Geol. Soc. Am. Bull.*, 63(11):1117–1142, 1952.

[32] M. Suderman. Pathwidth And Layered Drawings Of Trees. *Int. J. Comput. Geometry Appl.*, 14(3):203–225, 2004.

[33] A. Tarski. A Lattice-Theoretical Fixpoint Theorem and its Applications. *Pacific J. Math.*, 5(2):285–309, 1955.

[34] K. N. Verma, H. Seidl, and T. Schwentick. On the Complexity of Equational Horn Clauses. In *CADE*, volume 3632 of *LNCS*, pages 337–352, 2005.

[35] M.K. Yntema. Inclusion relations among families of context-free languages. *Information and Control*, 10:572–597, 1967.