Information and Computation 246 (2016) 43–61

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

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

Michael Luttenberger *, Maximilian Schlund *

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

ARTICLE INFO

Article history:
Received 20 July 2013
Available online 2 December 2015

Keywords:
Newton's method
Polynomial fixed-point equations
Semirings
Algebraic language theory
Horton–Strahler number

ABSTRACT

We give a lower bound on the speed at which Newton's method (as defined in [11]) 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 and Ésik [2]. 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.

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[1] guarantees convergence if F is an ω-continuous map over a complete partial order. In reference to Kleene's fixed-point theorem, 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, 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.

☆ 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 authors.
E-mail addresses: luttenbe@model.in.tum.de (M. Luttenberger), schlund@model.in.tum.de (M. Schlund).

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

http://dx.doi.org/10.1016/j.ic.2015.11.008
0890-5401/

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]). Independently 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)

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

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^nc^{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 Example 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[2] 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 $\mathbb{N}_k = \{0, 1, \ldots, k\}$.

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

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