TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Informatik VII

Solving Systems of Positive Polynomial Equations

Stefan Kiefer

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. H. Seidl
Prüfer der Dissertation:	1. Univ.-Prof. Dr. F. J. Esparza Estaun 2. Univ.-Prof. Dr. H.-J. Bungartz

Die Dissertation wurde am 25.06.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 02.10.2009 angenommen.

Abstract

In this thesis, we consider equation systems of the form

$$\begin{array}{rcl} X_1 & = & f_1(X_1, \ldots, X_n) \\ & \vdots & \\ X_n & = & f_n(X_1, \ldots, X_n) \end{array}$$

where $f_i(X_1, \ldots, X_n)$ is, for all $i \in \{1, \ldots, n\}$, an expression built up from real-valued variables $X_1, \ldots, X_n$, nonnegative real constants, and the operators multiplication, addition, minimum and maximum. We call such an equation system *positive* and denote it in vector form by $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$. The least solution is called $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}$ is the least fixed point of $\boldsymbol{f}$.

Positive equation systems appear naturally in the analysis of stochastic models like stochastic context-free grammars (with numerous applications to natural language processing and computational biology), probabilistic programs with procedures, web-surfing models with back buttons, branching processes, and termination games. The solution $\boldsymbol{\mu}$ of a positive equation system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ is of central interest for these models. Efficient methods to compute $\boldsymbol{\mu}$ are the main subject of this thesis.

For positive equation systems without minimum or maximum operator, Newton's method for approximating a zero of a differentiable function can be applied to approximate $\boldsymbol{\mu}$. In the first part of the thesis, we study in detail the convergence speed of Newton's method for such equation systems and show, in particular, that Newton's method converges at least linearly to $\boldsymbol{\mu}$. We also give concrete bounds on the convergence rate.

To compute the least fixed point of general positive equation systems with minimum and maximum operators, Newton's method cannot be directly used. In the second part, we suggest two algorithms that combine Newton's method with linear programming. We show that these methods converge linearly to $\boldsymbol{\mu}$ and give bounds on the convergence rate. We also show that one of those methods can be used to compute near-optimal strategies for the game associated with positive equation systems.

Acknowledgments

This thesis would not have been possible without the guidance, generosity and goodwill of many people. I feel grateful and indebted to have received all their help.

This applies, first and foremost, to my supervisor Prof. Javier Esparza, who has always had time for me. Countless fruitful discussions with him, his intelligent insights, and his continuous invaluable support have made this thesis possible. His inspiring personality will have a long-lasting influence on my future life.

Large parts of this thesis result from joint work with my colleague and friend Michael Luttenberger, whose ingenuity and hard work were indispensable for the results on Newton's method. I would like to express my gratitude to Prof. Helmut Seidl, who suggested the extensions to minimum and maximum operators, and to Thomas Gawlitza for a very friendly, intense and fruitful collaboration. Special thanks go to Prof. Hans-Joachim Bungartz for being a referee of this thesis and to Prof. Volker Diekert for his support in Stuttgart.

I want to thank the Universität Stuttgart, the Technische Universität München, and the Deutsche Forschungsgemeinschaft (DFG), which all provided essential financial and organizational support.

My colleagues made my time in Stuttgart and Munich a pleasure. The great atmosphere in our group was, in particular, due to Michael Luttenberger, Stefan Schwoon, and Dejvuth Suwimonteerabuth, who were always ready for both work and amusement. I thank them and many other former and present colleagues for contributing to the perfect working environment in Prof. Esparza's group.

My parents were and are the source of encouragement, love and support throughout the years. Thank you for everything.

Contents

Outline

In this thesis, we consider equation systems of the form

$$\begin{array}{rcl} X_1 & = & f_1(X_1, \ldots, X_n) \\ & \vdots & \\ X_n & = & f_n(X_1, \ldots, X_n) \end{array}$$

where $f_i(X_1, \ldots, X_n)$ is, for all $i \in \{1, \ldots, n\}$, an expression built up from the real-valued variables $X_1, \ldots, X_n$, nonnegative real constants, and the operators multiplication, addition, minimum and maximum. We call such an equation system *positive* and denote it in vector form by $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$. The least solution is called $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}$ is the least fixed point of $\boldsymbol{f}$.

Positive equation systems appear naturally in the analysis of stochastic models like stochastic context-free grammars (with numerous applications to natural language processing and computational biology), probabilistic programs with procedures, web-surfing models with back buttons, branching processes, and termination games. The solution $\boldsymbol{\mu}$ of a positive equation system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ is of central interest for these models. Efficient methods to compute $\boldsymbol{\mu}$ are the main subject of this thesis. **Chapter 0** contains an extensive introduction to the topic. All results are contained in Chapter 1 and Chapter 2.

In **Chapter 1**, the expressions f_i are restricted to be polynomials with nonnegative coefficients, i.e., the operators minimum and maximum are not allowed. For such equation systems, Etessami and Yannakakis [EY09] suggested to use Newton's method, the classical approximation technique in numerical analysis. More precisely, their algorithm decomposes the equation system in strongly connected components (where each variable depends directly or indirectly on every other variable) and applies Newton's method in each component. In Chapter 1 we extend and improve Etessami and Yannakakis' results. More concretely, we show:

- If Newton's method is started at the vector $\boldsymbol{0}$, it converges monotonically to $\boldsymbol{\mu}$, no matter if the equation system is strongly connected or not.
- Newton's method converges to $\boldsymbol{\mu}$ at least linearly, i.e., the number of valid bits is at least a linear function of the number of iterations performed. In addition, we show:
  - For strongly connected systems $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$, there is a "threshold" $k_{\boldsymbol{f}}$ such that for all $i \geq 0$, the $(k_{\boldsymbol{f}} + i)$-th Newton iterate, has at least i valid bits. By "at least i valid bits" we mean that, in each component, the relative error of the Newton iterate is at most 2^{-i}. In addition, we give concrete upper bounds on $k_{\boldsymbol{f}}$.
  - For systems that are not strongly connected, the convergence rate (i.e., the number of additional valid bits per iteration) is poorer. We provide bounds for the convergence rate and show that they are essentially tight.

In **Chapter 2**, we consider general positive equation systems, i.e., we allow minimum and maximum operators. Such equation systems arise in population models where two players are allowed to influence certain individuals; one player (the *terminator*) strives to extinguish the population, the other player (the *savior*) has the opposite objective. Newton's method, directly applied to such equation systems, does not always converge to $\boldsymbol{\mu}$. However, it can be adapted to a method which converges linearly to $\boldsymbol{\mu}$. More concretely, we obtain the following results:

- We propose two extensions of Newton's method that both approximate $\boldsymbol{\mu}$ for any positive equation system. We show that both of them converge monotonically and linearly to $\boldsymbol{\mu}$.
- One of the proposed algorithms computes, as a byproduct, for each iterate $\boldsymbol{\nu}$, a strategy for the terminator that guarantees the terminator a winning probability of at least $\boldsymbol{\nu}$. Since the iterates converge to $\boldsymbol{\mu}$, these strategies are near-optimal.

Chapter 2 builds on results of Chapter 1, but Chapter 2 can be understood without studying Chapter 1 in detail. We provide conclusions of our work at the end of Chapter 1 and Chapter 2, respectively.

The main themes of this work are fixed-point equations, and variants of Newton's method to solve them. This thesis ends with a kind of "epilogue" in **Chapter 3**, which sketches a generalization of positive fixed-point equations to fixed-point equations in semirings. Such equation systems can be solved using a generalization of Newton's method, and several results of this thesis find an analogue in a much more general setting.

Chapter 0

Introduction

In this thesis, we consider equation systems of the form

$$\begin{aligned} X_1 &= f_1(X_1, \ldots, X_n) \\ &\vdots \\ X_n &= f_n(X_1, \ldots, X_n) \end{aligned}$$

where, for all $i \in \{1, \ldots, n\}$, $f_i(X_1, \ldots, X_n)$ is an expression built up from the real-valued variables $X_1, \ldots, X_n$, nonnegative real constants, and the operators multiplication, addition, minimum and maximum. We call such an equation system *positive* and denote it in vector form by $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$. The least solution is called $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}$ is the least fixed point of $\boldsymbol{f}$.

Positive equation systems appear naturally in the analysis of stochastic models like stochastic context-free grammars (with numerous applications to natural language processing and computational biology), probabilistic programs with procedures, web-surfing models with back buttons, branching processes, and termination games. The solution $\boldsymbol{\mu}$ of a positive equation system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ is of central interest for these models. Efficient methods to compute $\boldsymbol{\mu}$ are the main subject of this thesis.

0.1 Systems of Positive Polynomials

In **Chapter 1**, the expressions f_i are restricted to be polynomials with nonnegative coefficients, i.e., the operators minimum and maximum are not allowed. In this case, $\boldsymbol{f}$ is a vector of polynomials, which we call a *system of positive polynomials*, or *SPP* for short. Figure 0.1 shows the graph of a 2-dimensional SPP equation $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$.

Equation systems $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ of this form appear naturally in the analysis of context-free grammars (with numerous applications to natural language processing [MS99, GJ02] and computational biology [SBH$^+$94, DEKM98, DE04, KH03]), probabilistic programs with procedures [EKM04, BKS05, EY09, EY05a, EKM05, EY05b, EY05c], and web-surfing models with back buttons [FKK$^+$00, FKK$^+$01]. More generally, they play an important role in the theory of *branching processes* [Har63, AN72], stochastic processes describing the evolution of a population whose individuals can die and reproduce. The probability of extinction of the population is the least solution of such a system, a result whose history goes back to [WG74].

Example 0.1. *One instance of the mentioned stochastic models is the web-surfing model with back buttons from [FKK$^+$00, FKK$^+$01]. Consider three webpages P_1, P_2, P_3 which are visited by a web surfer as follows.*

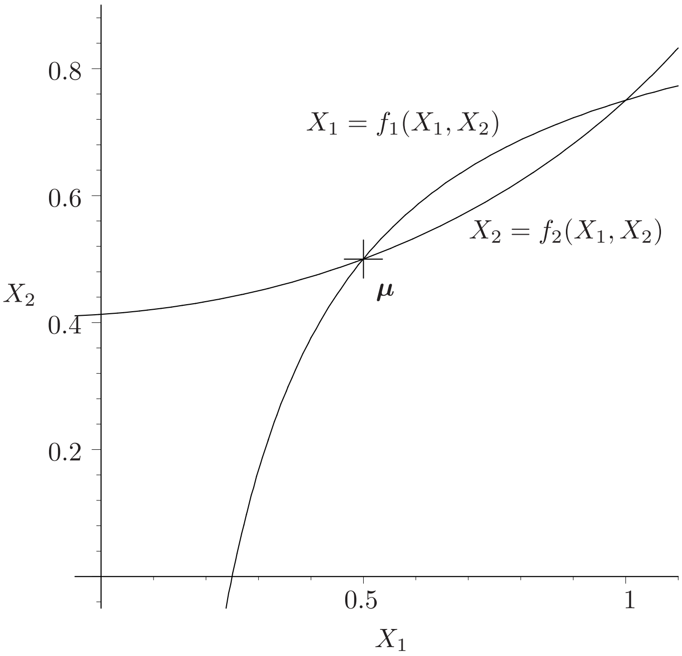

Figure 0.1: Graphs of the equations $X_1 = f_1(X_1, X_2)$ and $X_2 = f_2(X_1, X_2)$ with $f_1(X_1, X_2) = X_1X_2 + \frac{1}{4}$ and $f_2(X_1, X_2) = \frac{1}{6}X_1^2 + \frac{1}{9}X_1X_2 + \frac{2}{9}X_2^2 + \frac{3}{8}$. There are two real solutions in $\mathbb{R}^2$, the least one is labelled with $\boldsymbol{\mu}$.

- *If the surfer is at P_1, she follows a link to P_2 with probability* 0.4*, or presses the back button of the browser with probability* 0.6*.*
- *At P_2, she surfs to P_1 with probability* 0.3*, to P_2 with probability* 0.4*, or presses the back button with probability* 0.3*.*
- *At P_3, she surfs to P_1 with probability* 0.3*, or presses the back button with probability* 0.7*.*

As usual in web browsers, the history of the visited pages is recorded using a stack. When the surfer clicks a link from page P_i to P_j, the old page P_i is put on the stack, and P_j becomes the new current page. When the back button is clicked, the topmost stack symbol is popped and replaces the current page.

In the analysis of such a web-surfing model [FKK+00, FKK+01], the so-called revocation *probabilities play an important role. The revocation probability of a page P is the probability that, when currently visiting webpage P and having $H_nH_{n-1}\ldots H_1$ as the history stack, then during subsequent surfing from P the surfer eventually returns to webpage H_n with $H_{n-1}\ldots H_1$ as the remaining browser history. In our example, the revocation probabilities solve the following equation system.*

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 0.4X_2X_1 + 0.6 \\ 0.3X_1X_2 + 0.4X_3X_2 + 0.3 \\ 0.3X_1X_3 + 0.7 \end{pmatrix}$$

To explain this equation system, consider X_1, the revocation probability of P_1. If P_1 is the current page, it can be revoked either by pressing the back button or by following the link to P_2 and subsequently revoking both P_2 and P_1. The probability of the first possibility is 0.6*, the probability of the second possibility is* $0.4X_2X_1$*.*

In fact, one can show that the revocation probabilities are the least (nonnegative) solution of the equation system. We will later show for this particular example that, although the vector $(1, 1, 1)$ *is a solution, it is not the least one, which means that there is a positive probability of never revoking a page.*

The least solution is also the relevant solution in the other mentioned models, which motivates our interest in this solution.

Since SPPs have positive coefficients, $\boldsymbol{x} \leq \boldsymbol{y}$ implies $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{y})$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$, i.e., the functions $f_1, \ldots, f_n$ are monotone. This guarantees that any feasible SPP, i.e., any SPP with at least one fixed point, has a least fixed point $\boldsymbol{\mu}$. This fact can be seen by applying Kleene's theorem (see for instance [Kui97]) which says that, by monotonicity of $\boldsymbol{f}$, the sequence $\boldsymbol{0}, \boldsymbol{f}(\boldsymbol{0}), \boldsymbol{f}(\boldsymbol{f}(\boldsymbol{0})), \ldots$ converges to the least fixed point $\boldsymbol{\mu}$. We call this sequence the *Kleene sequence* and define the *Kleene iterates* $\boldsymbol{\kappa}^{(0)} = \boldsymbol{0}$ and $\boldsymbol{\kappa}^{(k+1)} = \boldsymbol{f}(\boldsymbol{\kappa}^{(k)})$ for all $k \geq 0$.

Example 0.2. *Consider the SPP equation* $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ *from Example 0.1 with*

$$\boldsymbol{f} = \begin{pmatrix} 0.4X_2X_1 + 0.6 \\ 0.3X_1X_2 + 0.4X_3X_2 + 0.3 \\ 0.3X_1X_3 + 0.7 \end{pmatrix} .$$

Then the first Kleene iterates are approximately:

$$\boldsymbol{\kappa}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(1)} = \begin{pmatrix} 0.6 \\ 0.3 \\ 0.7 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(2)} = \begin{pmatrix} 0.672 \\ 0.438 \\ 0.826 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(3)} = \begin{pmatrix} 0.718 \\ 0.533 \\ 0.867 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(4)} = \begin{pmatrix} 0.753 \\ 0.600 \\ 0.887 \end{pmatrix}$$

Galois theory [Ste00] implies that $\boldsymbol{\mu}$ can be irrational and non-expressible by radicals.

Example 0.3. *The least fixed point of* $\frac{1}{6}X^6 + \frac{1}{2}X^5 + \frac{1}{3}$ *is not expressible by radicals.*

Computational Complexity

We briefly present some results on the complexity of computing $\boldsymbol{\mu}$, or, more precisely, of computing bounds on $\boldsymbol{\mu}$. Let SPP-DECISION be the following problem:

> Given an SPP $\boldsymbol{f}$ and a vector $\boldsymbol{v}$ encoded in binary, decide whether $\boldsymbol{\mu} \leq \boldsymbol{v}$ holds.

It is known that SPP-DECISION is in PSPACE:

> In order to decide whether $\mu_1 \leq v$ holds for the first component of μ_1 of the least fixed point of a 2-dimensional SPP $\boldsymbol{f}$, one can equivalently decide if the following formula is true:
>
> $$\exists x_1 \in \mathbb{R}, x_2 \in \mathbb{R} : x_1 = f_1(x_1, x_2) \ \wedge \ x_2 = f_2(x_1, x_2) \ \wedge x_1, x_2 \geq 0 \ \wedge x_1 \leq a$$
>
> Such formulas can be decided in PSPACE, because the first-order theory of the reals is decidable, and its existential fragment is even in PSPACE [Can88].

On the other hand, SPP-DECISION is at least as hard [EY09] as the following problem, called SQUARE-ROOT-SUM:

> Given $k + 1$ natural numbers $n_1, \ldots, n_k$ and b, decide whether $\sum_{i=1}^{k} \sqrt{n_i} \leq b$ holds.

The SQUARE-ROOT-PROBLEM is a natural subproblem of many questions in computational geometry. For instance, the length of the boundary of a polygon whose vertices lie in $\mathbb{Z}^2$ is a sum of square roots of integers. It has been a major open problem since the 70s whether SQUARE-ROOT-SUM belongs to NP.

The following problem is also polynomial-time reducible [EY09] to SPP-DECISION. It is called PosSLP (positive straight-line program):

> Given an arithmetic circuit with integer inputs and gates $\{+, -, \cdot\}$, decide whether it outputs a positive number.

PosSLP has been recently shown to play a central role in understanding the Blum-Shub-Smale model of computation, where each single arithmetic operation over the reals can be carried out exactly and in constant time [ABKPM09].

We conclude that, while SPP-DECISION is in PSPACE, it is unlikely to be in P.

Approximating the Least Fixed Point and Newton's Method

While the mentioned results on SPP-DECISION provide important information on the complexity of solving SPP equations, for the practical applications mentioned above the problem of determining if $\boldsymbol{\mu}$ exceeds a given bound is less relevant than the complexity of, given a number $i \geq 0$, computing i *valid bits* of $\boldsymbol{\mu}$, i.e., computing a vector $\boldsymbol{\nu}$ such that $|\mu_j - \nu_j| \,/\, |\mu_j| \leq 2^{-i}$ for every $1 \leq j \leq n$. In this thesis we study this problem in the Blum-Shub-Smale model, where each single arithmetic operation over the reals can be carried out exactly and in constant time.

To approximate $\boldsymbol{\mu}$, one can use the sequence of Kleene iterates $\boldsymbol{\kappa}^{(k)} = \boldsymbol{f}^k(\mathbf{0})$, which converges to $\boldsymbol{\mu}$ by Kleene's theorem. However, the convergence may be very slow.

Example 0.4. *For the 1-dimensional SPP $f(X) = \frac{1}{2}X^2 + \frac{1}{2}$ (with $\mu = 1$), the k-th Kleene iterate $\kappa^{(k)}$ satisfies $\kappa^{(k)} \leq 1 - \frac{1}{k+1}$ for every $i \geq 0$, as shown in [EY09]. Hence, the number of iterations needed to compute i bits of μ is exponential in i. We call that* logarithmic convergence*, because the number of valid bits is a logarithmic function of the number of iterations. Here are some of the Kleene iterates.*

$$\begin{array}{l} \kappa^{(0)} = 0, \quad \kappa^{(1)} = 0.5, \quad \kappa^{(2)} = 0.625, \quad \kappa^{(3)} = 0.695, \quad \kappa^{(4)} = 0.742, \quad \kappa^{(5)} = 0.775 \\ \quad \ldots \\ \kappa^{(20)} = 0.920, \ \ldots, \ \kappa^{(200)} = 0.990, \ \ldots, \ \kappa^{(2000)} = 0.9990, \ \ldots, \ \kappa^{(20000)} = 0.99990, \ \ldots \end{array}$$

Faster approximation techniques have been known for a long time. In particular, *Newton's method*, suggested by Isaac Newton more than 300 years ago, is a standard efficient technique for approximating a zero of a differentiable function [OR70]. Since a fixed point of a function $f(X)$ is a zero of $F(X) = f(X) - X$, the method can be applied to search for fixed points of $f(X)$.

We briefly recall the method for the case of one variable, see Figure 0.2 for an illustration. Starting at some value $\nu^{(0)}$ "close enough" to the zero of $F(X)$, Newton's method proceeds iteratively: given $\nu^{(k)}$, we compute a value $\nu^{(k+1)}$ closer to the zero than $\nu^{(k)}$. For that, we compute the tangent to $F(X)$ passing through the point $(\nu^{(k)}, F(\nu^{(k)}))$, and take $\nu^{(k+1)}$ as the zero of the tangent (i.e., the X-coordinate of the point at which the tangent cuts the X-axis). Basic arithmetic leads to:

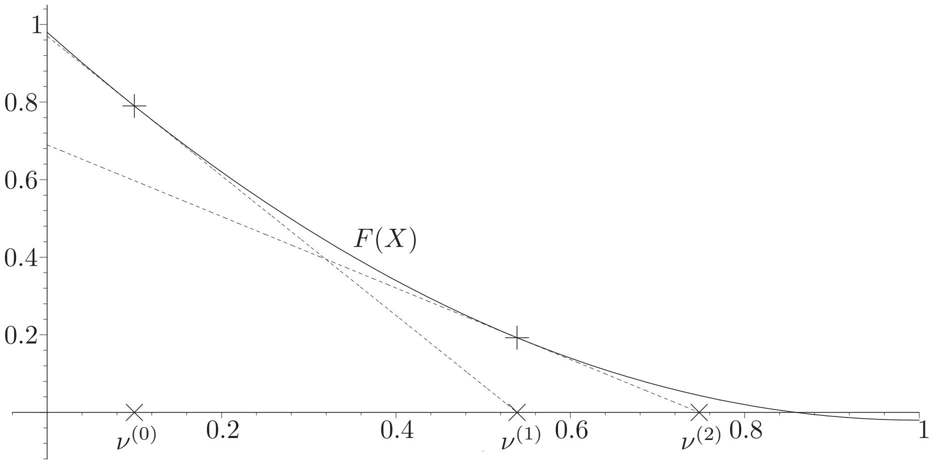

Figure 0.2: Newton's method to find a zero of a one-dimensional function $F(X)$.

$$\nu^{(k+1)} = \nu^{(k)} - \frac{F(\nu^{(k)})}{F'(\nu^{(k)})} = \nu^{(k)} + \frac{f(\nu^{(k)}) - \nu^{(k)}}{1 - f'(\nu^{(k)})}$$

Newton's method can be easily generalized to the multivariate case:

$$\boldsymbol{\nu}^{(k+1)} = \boldsymbol{\nu}^{(k)} + (I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}))^{-1}(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})$$

where $\boldsymbol{f}'(\boldsymbol{X})$ is the Jacobian of $\boldsymbol{f}$, i.e., the matrix of partial derivatives of $\boldsymbol{f}$, and I is the identity matrix. Computing the matrix inverse $(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}))^{-1}$ can be avoided by solving the linear equation system

$$\left(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})\right)(\boldsymbol{x} - \boldsymbol{\nu}^{(k)}) = \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)} \tag{1}$$

which is equivalent to

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})(\boldsymbol{x} - \boldsymbol{\nu}^{(k)})\,.$$

Notice that $\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})(\boldsymbol{x} - \boldsymbol{\nu}^{(k)})$ is the first-order Taylor approximation of $\boldsymbol{f}$ at $\boldsymbol{\nu}^{(k)}$, i.e., in each step, Newton's method computes a linearization $\overline{\boldsymbol{f}}$ of $\boldsymbol{f}$ and solves a linear system $\boldsymbol{X} = \overline{\boldsymbol{f}}(\boldsymbol{X})$ rather than the nonlinear system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$.

Example 0.5. *Consider the equation system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ from Examples 0.1 and 0.2 with*

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} 0.4X_2X_1 + 0.6 \\ 0.3X_1X_2 + 0.4X_3X_2 + 0.3 \\ 0.3X_1X_3 + 0.7 \end{pmatrix}.$$

The Jacobian matrix of partial derivatives is

$$\boldsymbol{f}'(\boldsymbol{X}) = \begin{pmatrix} 0.4X_2 & 0.4X_1 & 0 \\ 0.3X_2 & 0.3X_1 + 0.4X_3 & 0.4X_2 \\ 0.3X_3 & 0 & 0.3X_1 \end{pmatrix}.$$

As starting point of Newton's method we take $\boldsymbol{\nu}^{(0)} = \boldsymbol{0}$. The next Newton iterate $\boldsymbol{\nu}^{(1)}$ can be obtained by solving (1)*:*

$$\begin{pmatrix} 1 - 0.4\cdot 0 & -0.4\cdot 0 & 0 \\ -0.3\cdot 0 & 1 - 0.3\cdot 0 - 0.4\cdot 0 & -0.4\cdot 0 \\ -0.3\cdot 0 & 0 & 1 - 0.3\cdot 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 - 0 \\ x_2 - 0 \\ x_3 - 0 \end{pmatrix} = \begin{pmatrix} 0.6 - 0 \\ 0.3 - 0 \\ 0.7 - 0 \end{pmatrix}$$

Its only solution is

$$\boldsymbol{\nu}^{(1)} = \begin{pmatrix} 0.6 \\ 0.3 \\ 0.7 \end{pmatrix} .$$

The next Newton iterate $\boldsymbol{\nu}^{(2)}$ *can, again, be obtained by solving* (1)*:*

$$\begin{pmatrix} 1 - 0.4 \cdot 0.3 & -0.4 \cdot 0.6 & 0 \\ -0.3 \cdot 0.3 & 1 - 0.3 \cdot 0.6 - 0.4 \cdot 0.7 & -0.4 \cdot 0.3 \\ -0.3 \cdot 0.7 & 0 & 1 - 0.3 \cdot 0.6 \end{pmatrix} \cdot \begin{pmatrix} x_1 - 0.6 \\ x_2 - 0.3 \\ x_3 - 0.7 \end{pmatrix}$$
$$= \begin{pmatrix} 0.4 \cdot 0.3 \cdot 0.6 + 0.6 - 0.6 \\ 0.3 \cdot 0.6 \cdot 0.3 + 0.4 \cdot 0.7 \cdot 0.3 + 0.3 - 0.3 \\ 0.3 \cdot 0.6 \cdot 0.7 + 0.7 - 0.7 \end{pmatrix}$$

Its only solution is

$$\boldsymbol{\nu}^{(2)} = \begin{pmatrix} 0.771 \\ 0.628 \\ 0.898 \end{pmatrix} .$$

The next Newton iterates can be obtained similarly:

$$\boldsymbol{\nu}^{(3)} = \begin{pmatrix} 0.877 \\ 0.812 \\ 0.948 \end{pmatrix} , \quad \boldsymbol{\nu}^{(4)} = \begin{pmatrix} 0.934 \\ 0.899 \\ 0.972 \end{pmatrix} , \quad \boldsymbol{\nu}^{(5)} = \begin{pmatrix} 0.962 \\ 0.942 \\ 0.984 \end{pmatrix} , \quad \ldots$$

Notice that the Newton sequence seems to be faster than the Kleene sequence (Example 0.2).

Example 0.6. *Consider again the 1-dimensional SPP* $f(X) = \frac{1}{2}X^2 + \frac{1}{2}$ *(with* $\mu = 1$*) from Example 0.4. Starting at* $\nu^{(0)} = 0$*, the first Newton iterates are:*

$$\nu^{(0)} = 0, \quad \nu^{(1)} = 1/2, \quad \nu^{(2)} = 3/4, \quad \nu^{(3)} = 7/8, \quad \nu^{(4)} = 15/16, \quad \ldots$$

In fact, it is easy to show that we have $\nu^{(k)} = 1 - \dfrac{1}{2^k}$ *for all* $k \geq 0$*. So the* k*-th iterate has* k *valid bits; we say the Newton sequence has* linear convergence*. This is in sharp contrast with the Kleene sequence (Example 0.4) which had only logarithmic convergence.*

Example 0.7. *If the SPP from Example 0.6 is slightly modified to* $f(X) = 2/3X^2 + 1/3$*, we get* $\mu = 1/2$*. Again starting at* $\nu^{(0)} = 0$*, the first Newton iterates are:*

$$\begin{aligned} &\nu^{(0)} = 0, \quad \nu^{(1)} = \tfrac{1}{3} \approx 0.33, \quad \nu^{(2)} = \tfrac{7}{15} \approx 0.47, \quad \nu^{(3)} = \tfrac{127}{255} \approx 0.498, \\ &\nu^{(4)} = \tfrac{32767}{65535} \approx 0.499992, \quad \ldots \end{aligned}$$

In fact, it is easy to show that we have $\nu^{(k)} = \dfrac{2^{2^k-1} - 1}{2^{2^k} - 1}$ *for all* $k \geq 0$*, and so the number of valid bits of the* k*-th iterate is approximately* 2^k*; we say the Newton sequence has* exponential convergence.[1]

Newton's method has to be used with care because it does not always converge, and may not even be well-defined. Figure 0.3 illustrates these problems for the equation $-X^4 + 3X^2 + 2 = 0$. If Newton's method is started at $+1$, it keeps oscillating between $+1$ and -1. If it is started at 0.1, it converges to the negative solution at ≈ -1.9, although the positive solution is closer. If it is started at 0, it is not even well-defined, because the tangent does not intersect the X-axis (or, more technically, the inverse of 0, i.e., the fraction 1/0, does not exist).

[1] In most of the literature, this convergence speed is called *quadratic convergence*, because the error is squared in each iteration. Our notion of convergence speed stresses that the precision is a function of the number of iterations.

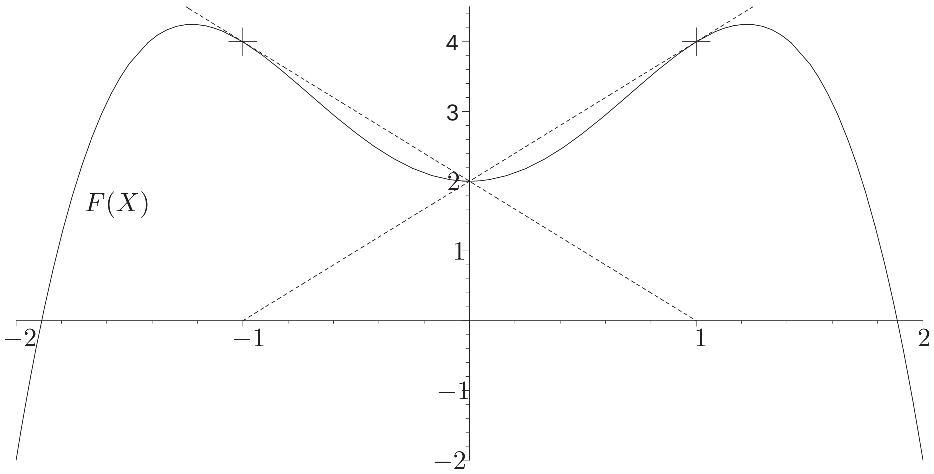

Figure 0.3: Newton's method for solving $-X^4 + 3X^2 + 2 = 0$ may oscillate.

Etessami and Yannakakis have initiated the study of fixed-point equations for SPPs in [EY09], and shown that a particular version of Newton's method always converges to $\boldsymbol{\mu}$, namely a version which decomposes the SPP into *strongly connected components* (SCCs)[2] and applies Newton's method to them in a bottom-up fashion. Our first result generalizes Etessami and Yannakakis': the ordinary Newton method converges to $\boldsymbol{\mu}$ for arbitrary SPPs, provided that $\boldsymbol{\mu}$ is nonzero in all components, which is easy to achieve by identifying and removing the 0-components.

While these results show that Newton's method can be an adequate algorithm for solving SPP equations, they provide no information on the number of iterations needed to compute i valid bits. To the best of our knowledge (and perhaps surprisingly), the rest of the literature does not contain relevant information either: it has not considered SPPs explicitly, and the existing results have very limited interest for SPPs, since they do not apply even for very simple and relevant SPP cases (see *Related work* below).

We obtain upper bounds on the number of iterations that Newton's method needs to produce i valid bits, first for strongly connected and then for arbitrary SPP equations. A single iteration requires $\mathcal{O}(n^3)$ arithmetic operations in a system of n equations, because a linear equation system can be solved by Gauss elimination which takes $\mathcal{O}(n^3)$ operations. This immediately gives an upper bound on the time complexity of Newton's method in the Blum-Shub-Smale model. We prove that for strongly connected SPP equations $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ there exists a threshold $k_{\boldsymbol{f}}$ such that, for every $i \geq 0$, the $(k_{\boldsymbol{f}} + i)$-th iteration of Newton's method has at least i valid bits of $\boldsymbol{\mu}$. So, loosely speaking, after $k_{\boldsymbol{f}}$ iterations Newton's method is guaranteed to compute at least 1 new bit of the solution per iteration; we say that Newton's method converges at least *linearly with rate 1.* Moreover, we show that the threshold $k_{\boldsymbol{f}}$ can be chosen as

$$k_{\boldsymbol{f}} = \lceil 4mn + 3n \max\{0, -\log \mu_{min}\} \rceil$$

where n is the number of polynomials of the SPP, m is such that all coefficients of the SPP can be given as ratios of m-bit integers, and μ_{min} is the minimal component of $\boldsymbol{\mu}$.

Notice that $k_{\boldsymbol{f}}$ depends on $\boldsymbol{\mu}$, which is what Newton's method should compute. For this reason we also obtain bounds on $k_{\boldsymbol{f}}$ depending only on m and n. We show that for arbitrary

[2] Loosely speaking, a subset of variables and their associated equations form an SCC if the value of any variable in the subset influences the value of all variables in the subset, see § 1.1 for details.

strongly connected SPP equations $k_{\boldsymbol{f}} = 4mn2^n$ is also a valid threshold. For SPP equations coming from stochastic models, such as the ones listed at the beginning of this chapter, we do far better. First, we show that if $\boldsymbol{f}(\boldsymbol{0})$ is greater than 0 in all components (a condition that always holds for back-button processes [FKK$^+$00, FKK$^+$01]), then a valid threshold is $k_{\boldsymbol{f}} = 2m(n+1)$. As a corollary, our result shows that for back-button processes, i valid bits can be computed in time $\mathcal{O}(mn^4 + in^3)$ in the Blum-Shub-Smale model. Second, we observe that, since $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\nu}^{(k+1)} \leq \boldsymbol{\mu}$ holds for every $k \geq 0$, the Newton iteration itself provides better and better lower bounds for μ_{min} and thus for $k_{\boldsymbol{f}}$. We exhibit an SPP for which, using this fact and our theorem, we can prove that no component of the solution reaches the value 1. This cannot be proved by just computing more iterations, no matter how many.

For general SPP equations, not necessarily strongly connected, we show that Newton's method still converges linearly, albeit the convergence rate is poorer. We expose a family of SPPs showing that this bound is essentially tight.

Related Work

There is a large body of literature on the convergence speed of Newton's method for arbitrary systems of differentiable functions. A comprehensive reference is Ortega and Rheinboldt's book [OR70] (see also Chapter 8 of Ortega's course [Ort72] or Chapter 5 of [Kel95] for a brief summary). Several theorems (for instance Theorem 8.1.10 of [Ort72]) prove that the number of valid bits grows linearly, superlinearly, or even exponentially in the number of iterations, but only under the hypothesis that $\boldsymbol{F}'(\boldsymbol{x})$ is non-singular everywhere, in a neighborhood of $\boldsymbol{\mu}$, or at least at the point $\boldsymbol{\mu}$ itself. However, the matrix $\boldsymbol{F}'(\boldsymbol{\mu})$ can be singular for an SPP, even for the 1-dimensional SPP $f(X) = \frac{1}{2}X^2 + \frac{1}{2}$.

The general case in which $\boldsymbol{F}'(\boldsymbol{\mu})$ may be singular for the solution $\boldsymbol{\mu}$ the method converges to has been thoroughly studied. In a seminal paper [Red78], Reddien shows that under certain conditions, the main ones being that the kernel of $\boldsymbol{F}'(\boldsymbol{\mu})$ has dimension 1 and that the initial point is close enough to the solution, Newton's method gains 1 bit per iteration. Decker and Kelly obtain results for kernels of arbitrary dimension, but they require a certain linear map $B(\boldsymbol{X})$ to be non-singular for all $\boldsymbol{x} \neq \boldsymbol{0}$ [DK80]. Griewank observes in [GO81] that the non-singularity of $B(\boldsymbol{X})$ is in fact a strong condition which, in particular, can only be satisfied by kernels of even dimension. He presents a weaker sufficient condition for linear convergence requiring $B(\boldsymbol{X})$ to be non-singular only at the initial point $\boldsymbol{\nu}^{(0)}$, i.e., it only requires to make "the right guess" for $\boldsymbol{\nu}^{(0)}$. Unfortunately, none of these results can be directly applied to arbitrary SPPs. The possible dimensions of the kernel of $\boldsymbol{F}'(\boldsymbol{\mu})$ for an SPP are to the best of our knowledge unknown, and deciding this question seems as hard as those related to the convergence rate.[3]

Kantorovich's famous theorem (see e.g. Theorem 8.2.6 of [OR70] and [PP80] for an improvement) guarantees global convergence and only requires $\boldsymbol{F}'$ to be non-singular at $\boldsymbol{\nu}^{(0)}$. However, it also requires to find a Lipschitz constant for $\boldsymbol{F}'$ on a suitable region and some other bounds on $\boldsymbol{F}'$. These latter conditions are far too restrictive for the applications mentioned above. For instance, in the back-button model described in Example 0.1, a webpage may not contain a link such that the product of the probabilities to click the the link and to press the back button is 1/4 or more. This class of models is too contrived to be of use.

Summarizing, while the convergence of Newton's method for systems of differentiable functions has been intensely studied, the case of SPPs does not seem to have been considered yet. The results obtained for other classes have very limited applicability to SPPs: either

[3] More precisely, SPPs with kernels of arbitrary dimension exist, but the cases we know of can be trivially reduced to SPPs with kernels of dimension 1.

they do not apply at all, or only apply to contrived SPP subclasses. Moreover, these results only provide information about the growth rate of the number of valid bits, but not about the number itself. Our thresholds lead to *explicit* lower bounds for the number of valid bits depending only on syntactical parameters: the number of equations and the size of the coefficients.

0.2 Systems of Positive Min-Max Polynomials

In **Chapter 2** we consider again positive equation systems:

$$\begin{array}{rcl} X_1 & = & f_1(X_1, \ldots, X_n) \\ & \vdots & \\ X_n & = & f_n(X_1, \ldots, X_n) \end{array}$$

In this chapter, the expressions f_i are min-max polynomials, i.e., they may contain $\wedge$ (minimum) and $\vee$ (maximum) operators. An example of a min-max polynomial is $3X_1X_2 + 5X_1^2 \ \wedge \ 4X_2$. A vector $\boldsymbol{f}$ of such min-max polynomials is called a *system of positive min-max-polynomials*, or *min-max-SPP* for short.

Min-max-SPPs naturally appear in the study of two-player stochastic games and competitive Markov decision processes, in which, broadly speaking, the next move is decided by one of the two players or by tossing a coin, depending on the game's position (see e.g. [NS03, FV97]). The min and max operators model the competition between the players. The product operator, which leads to non-linear equations, allows to deal with recursive stochastic games [EY05c, EY06], a class of games with an infinite number of positions, and having as special case *extinction games*, games in which players influence with their actions the development of a population whose members reproduce and die, and the players' goals are to extinguish the population or keep it alive.

Example 0.8. *Imagine a patient who has the flu. The doctor has two options:*

- *she can either not treat him with any medication;*
- *or she treats him with a newly developed medicine called* Muniflu.

If she does not treat him, the probability that the patient recovers without infecting anyone else is 0.3*, but with a probability of* 0.7 *he infects someone else. If she chooses to treat him with Muniflu, the therapy takes effect with a probability of* 0.9*, but with a probability of* 0.1 *the patient must still be considered as untreated. Letting* U *(resp.* T*) denote the probability to cure an initially* <u>u</u>*ntreated (resp.* <u>t</u>*reated) patient and all people he infects, this gives rise to the equation*

$$U = 0.3 + 0.7UU \ \vee \ 0.9T + 0.1U\,,$$

where the maximum operator is due to the fact that the doctor will choose the option that promises a higher probability of extinguishing the flu. We could have more complicated infection models with probabilities p_i *to infect* i *people. In this cases, the term* $0.3 + 0.7UU$ *would be replaced by* $\sum_{i=0}^{d} p_i U^i$ *for some number* $d \in \mathbb{N}$*, where* d *must be finite because we do not consider power series.*

A treated flu patient responds to Muniflu as follows. If he has Influenza A, the probability that he recovers without infecting anybody is 0.35*, but with a probability of* 0.65 *he infects another (initially untreated) person. If he has Influenza B, the probability that he recovers* *without infecting anybody is* 0.5*, but there is a probability of* 0.2 *to infect another person, and even a probability of* 0.3 *to infect two other people. This gives rise to the equation*

$$T = 0.35 + 0.65TU \ \wedge \ 0.5 + 0.2TU + 0.3TUU\,,$$

where the minimum operator expresses the fact that the doctor makes her decision based on a worst-case assumption on the influenza type.

As in the first part of the thesis, the relevant solution is $\boldsymbol{\mu}$, i.e., the least one. It can be interpreted as the probability to extinguish the flu, assuming that, initially, there is exactly one flu patient, and assuming that both the doctor (who decides whether she should use Muniflu) and the flu (which "decides" the influenza type A or B) play optimally.

This scenario is an instance of an extinction game*, which are games for two players, called* terminator *and* savior*. The terminator, here the doctor, tries to extinguish the flu patients (by curing them, of course!), the savior, here the flu, tries to prevent that. The doctor may also wish to know her optimal strategy, i.e., she wants to know whether she should use Muniflu or not in order to achieve success probabilities of (at least) $\boldsymbol{\mu}$.*

Min-max-SPP equations generalize several other classes of equation systems. If product of variables is disallowed, we obtain systems of min-max *linear* equations, which appear in classical two-person stochastic games with a finite number of game positions. The problem of solving these systems has been thoroughly studied [Con92, GS07a, GS07b]. If both min and max are disallowed, we obtain monotone systems of polynomial equations, which are central to the study of recursive Markov chains and probabilistic pushdown systems, and are studied in the first part of this thesis. If only one of min or max is disallowed, we obtain a class of systems corresponding to recursive Markov decision processes [EY05c, EY06]. All these models have applications in the analysis of probabilistic programs with procedures [WE07].

As for SPPs, Kleene's theorem guarantees that if a min-max-SPP has a fixed point then it also has a *least* one, denoted by $\boldsymbol{\mu f}$ or $\boldsymbol{\mu}$, which is also the relevant fixed point for the applications mentioned above. As for SPPs, Kleene's theorem also ensures that the Kleene sequence $(\boldsymbol{\kappa}^{(k)})_{k\in\mathbb{N}}$ with $\boldsymbol{\kappa}^{(0)} = \mathbf{0}$ and $\boldsymbol{\kappa}^{(k+1)} = \boldsymbol{f}(\boldsymbol{\kappa}^{(k)})$ converges to $\boldsymbol{\mu}$. However, as mentioned in § 0.1, this procedure can converge very slowly ("logarithmically"), even without minimum or maximum operators. Thus, the goal is again to replace the function $\boldsymbol{f}$ by an operator $G : \mathbb{R}^n \to \mathbb{R}^n$ such that the respective iterative process also converges to $\boldsymbol{\mu}$ but faster. In fact, we would like to use Newton's method also for min-max-SPPs. However, we cannot directly use the Newton operator from Definition 1.11 because for arbitrary min-max-SPPs there is no guarantee that the next approximant still lies below the least solution, and the sequence of approximants may even diverge.

Example 0.9. *Consider the 1-dimensional min-SPP f with $f(X) = g(X) \wedge h(X)$ where*

$$g(X) = 0.7 \cdot X^2 + 0.1 \cdot X + 0.4 \quad \text{and} \quad h(X) = 0.1 \cdot X^2 + 0.1 \cdot X + 1.4\,,$$

see Figure 0.4. As $f(X) = g(X) \wedge h(X)$, the graph of $f(X) - X$ is the lower, non-dashed, part of the graphs of $g(X) - X$ and $h(X) - X$. The least fixed point of f is $\mu = 2$. The figure shows what happens if Newton's method is applied to $f(X) - X = 0$. In this example we have $0 = \nu^{(0)} < \nu^{(1)} < \nu^{(2)} > \nu^{(3)}$, so the Newton sequence does not converge to μ, at least not monotonically. Therefore, Newton's method cannot be directly used for min-max-SPPs.

For this reason, the tool from [WE07], called PReMo, uses round-robin iteration for min-max-SPPs, a slight optimization of Kleene iteration. Unfortunately, this technique also exhibits logarithmic convergence order in the worst case.

In the second part of the thesis we overcome the problem of Newton's method. Instead of approximating $\boldsymbol{f}$ at the current approximant $\boldsymbol{\nu}^{(k)}$ by a linear function, we approximate it by a *piecewise* linear function, as illustrated in the following example.

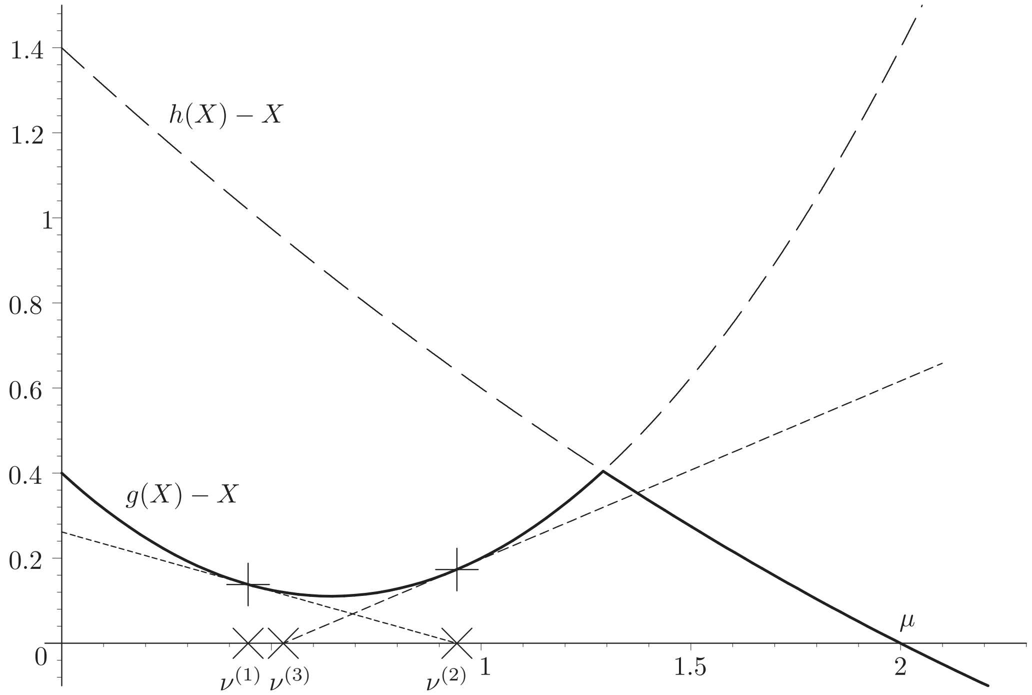

Figure 0.4: Newton's method applied to $f(X) - X = 0$ with $f(X) = g(X) \wedge h(X)$ does not converge to μ.

Example 0.10. *Consider again the 1-dimensional min-SPP f with $f(X) = g(X) \wedge h(X)$ from Example 0.9. In Example 0.9 we applied Newton's method to $\nu^{(2)}$ which yielded a point $\nu^{(3)}$ with $\nu^{(3)} < \nu^{(2)}$. This problem is overcome in two steps:*

(1) When Newton's method linearizes the function $f(X)$ at the point $\nu^{(2)}$, it actually linearizes $g(X)$ at $\nu^{(2)}$, because $f(\nu^{(2)}) = g(\nu^{(2)}) \wedge h(\nu^{(2)}) = g(\nu^{(2)})$. In our "repaired" Newton's method, we compute linearizations of both $g(X)$ and $h(X)$ at $\nu^{(2)}$, say $\overline{g}(X)$ and $\overline{h}(X)$. Then we let $\overline{f}(X) := \overline{g}(X) \wedge \overline{h}(X)$ and look for solutions of $\overline{f}(X) - X = 0$, see Figure 0.5.

(2) In the example, the piecewise linear equation $\overline{f}(X) - X = 0$ has two solutions, one approximately at 0.5, the other one approximately at 1.85, see Figure 0.5. In our "repaired" Newton's method, we take as next iterate $\nu^{(3)}$ the least solution that is greater than the current iterate $\nu^{(2)}$.

The approach of Example 0.10 can be suitably generalized to multidimensional min-SPPs. We can also treat maximum operators. In fact, we offer two methods that solve multidimensional min-max-SPP equations, which differ in the treatment of maximum operators. This is illustrated in the next example.

Example 0.11. *Consider the 1-dimensional max-SPP f with $f(X) = g(X) \vee h(X)$ where*

$$g(X) = 0.5 \cdot X^2 + 0.7 \cdot X + 0.04 \quad \textit{and} \quad h(X) = 0.1 + 2.2 \cdot X^2 \,,$$

see Figure 0.6. As $f(X) = g(X) \vee h(X)$, the graph of $f(X) - X$ is the upper, non-dashed, part of the graphs of $g(X) - X$ and $h(X) - X$. The least fixed point of f is $\mu = 0.2$. To approximate it, we start again at the point 0. We offer two methods to compute the next approximant.

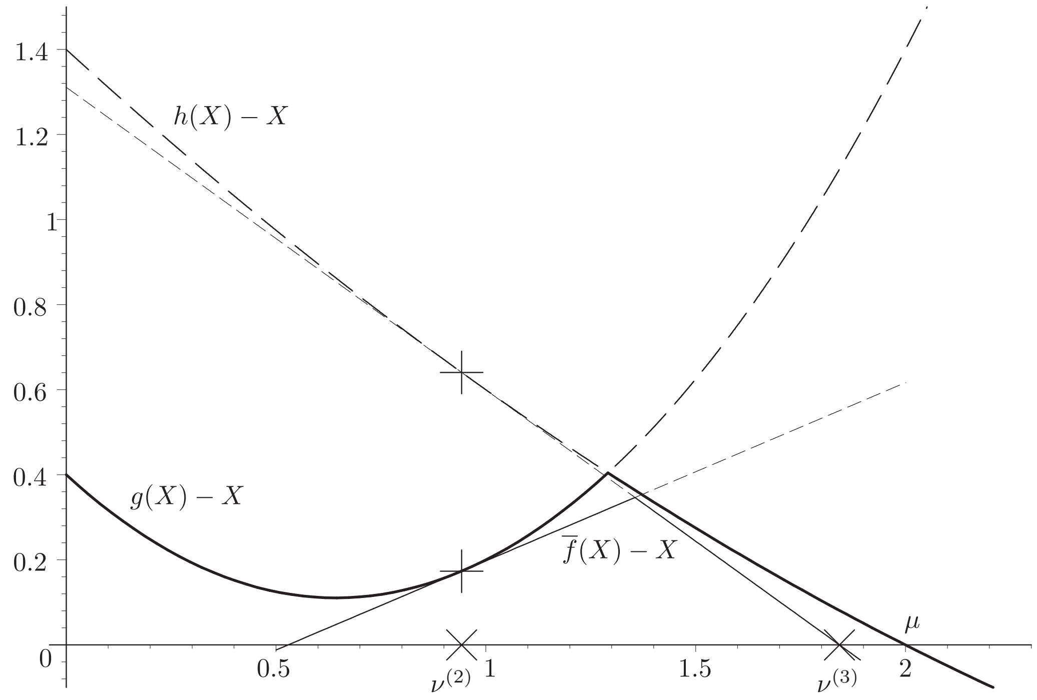

Figure 0.5: The “repaired” Newton’s method: Both $g(X)$ and $h(X)$ are linearized at the current iterate $\nu^{(2)}$, leading to a piecewise linear function $\overline{f}(X)$. The next approximant $\nu^{(3)}$ is the least solution of $\overline{f}(X) - X$ that is greater than $\nu^{(2)}$.

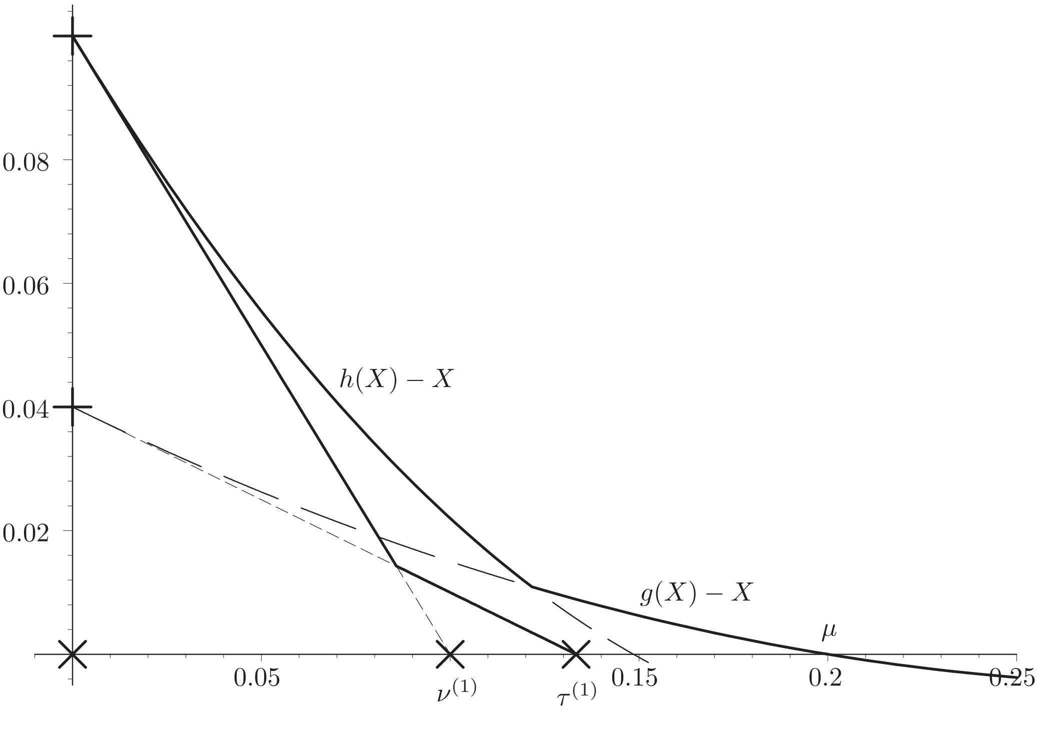

Figure 0.6: There are two methods to approximate the least fixed point μ of the function $g(X) \vee h(X)$. One leads to $\nu^{(1)}$ as the first iterate, the other one to $\tau^{(1)}$.

(a) We treat the maximum operator in the same way as the minimum operator, cf. Example 0.10. That is, we compute linearizations of both $g(X)$ and $h(X)$ at 0, say $\overline{g}(X)$ and $\overline{h}(X)$. Then we let $\overline{f}(X) := \overline{g}(X) \vee \overline{h}(X)$ and take as the next iterate $\tau^{(1)}$ the least solution of $\overline{f}(X) - X = 0$ that is greater than the current iterate 0, see Figure 0.6.

(b) We use the "raw" form of Newton's method. That is, we linearize f at 0. Since $f(0) = g(0) \vee h(0) = h(0)$, the linearization of f equals the linearization $\overline{h}$ of h at 0. We take as the next iterate $\nu^{(1)}$ the least solution of $\overline{h}(X) - X = 0$ that is greater than the current iterate 0, see Figure 0.6.

The approach of Example 0.10 to treat minimum operators can be combined with either of the approaches of Example 0.11 to treat maximum operators. This gives us two methods that iteratively approximate the least fixed point of arbitrary min-max-SPPs of arbitrary dimension. Since the algorithms are based on Newton's method, we can use the results of the first part of the thesis to show that both algorithms converge linearly to $\boldsymbol{\mu}$, i.e., the number of valid bits is at least a linear function of the number of iterations.

The method based on the idea of Example 0.11 (a), is called $\boldsymbol{\tau}$*-method.* In each step, it solves an equation system $\boldsymbol{X} = \overline{\boldsymbol{f}}(\boldsymbol{X})$ where each component of the vector $\overline{\boldsymbol{f}}(\boldsymbol{X})$ is an expression built up from *linear* (degree at most 1) polynomials and minimum and maximum operators. Such an equation system can be solved using a method from [GS07b] which is based on linear programming and strategy iteration.

The method based on the idea of Example 0.11 (b), is called $\boldsymbol{\nu}$*-method.* In each step, it solves an equation system $\boldsymbol{X} = \overline{\boldsymbol{f}}(\boldsymbol{X})$ where each component of the vector $\overline{\boldsymbol{f}}(\boldsymbol{X})$ is an expression built up from linear (degree at most 1) polynomials and minimum operators, but without maximum operators. The solution of such an equation system can be found by solving one linear programming (LP) problem.

Both methods converge monotonically to $\boldsymbol{\mu}$, i.e., all approximants are lower bounds on $\boldsymbol{\mu}$, and the approximants converge to $\boldsymbol{\mu}$. One step of the $\boldsymbol{\tau}$-method is more expensive than one step of the $\boldsymbol{\nu}$-method, but converges faster to $\boldsymbol{\mu}$. This can, in fact, already be observed in Example 0.11.

For min-max-SPPs derived from extinction games, the $\boldsymbol{\nu}$-method computes, as a byproduct, good strategies for the terminator. More precisely, the $\boldsymbol{\nu}$-method computes, along with each approximant $\boldsymbol{\nu}^{(k)}$, a strategy for the terminator that guarantees her/him termination probabilities of at least the current approximant $\boldsymbol{\nu}^{(k)}$. In other words, not only obtains the terminator lower bounds $\boldsymbol{\nu}^{(k)}$ on $\boldsymbol{\mu}$ (the success probability if both players play optimally), but also learns how to play in order to achieve at least $\boldsymbol{\nu}^{(k)}$. Since the $\boldsymbol{\nu}^{(k)}$ converge to $\boldsymbol{\mu}$, we say the computed strategies are ε-optimal. Applied to Example 0.8, this means the doctor will find out what to do in order to achieve a near-optimal curing probability, i.e., she will find out whether she should treat the patients with Muniflu or not.

Chapter 1

Systems of Positive Polynomials

In this chapter we study systems of positive polynomials (SPPs) and Newton's method to compute the least fixed point of SPPs. § 1.1 defines SPPs and describes their applications to stochastic systems. § 1.2 presents a short summary of our main theorems. § 1.3 proves some fundamental properties of Newton's method for SPP equations. § 1.4 and § 1.5 contain our results on the convergence speed for strongly connected and general SPP equations, respectively. § 1.6 shows that the bounds are essentially tight. § 1.7 contains conclusions.

1.1 Preliminaries

In this section we fix our notation, formalize the concepts mentioned in the introduction, and describe some stochastic models whose analysis leads to SPPs.

1.1.1 Notation

As usual, $\mathbb{R}$ and $\mathbb{N}$ denote the set of real, respectively natural numbers. We assume $0 \in \mathbb{N}$. $\mathbb{R}^n$ denotes the set of n-dimensional real valued column vectors and $\mathbb{R}^n_{\geq 0}$ the subset of vectors with nonnegative components. We use bold letters for vectors, e.g. $\boldsymbol{x} \in \mathbb{R}^n$, where we assume that $\boldsymbol{x}$ has the components $x_1, \ldots, x_n$. Similarly, the i-th component of a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is denoted by f_i. We define $\boldsymbol{0} := (0, \ldots, 0)^\top$ and $\boldsymbol{1} := (1, \ldots, 1)^\top$ where the superscript $^\top$ indicates the transpose of a vector or a matrix. Let $\|\cdot\|$ denote some norm on $\mathbb{R}^n$. Sometimes we use explicitly the maximum norm $\|\cdot\|_\infty$ with $\|\boldsymbol{x}\|_\infty := \max_{1 \leq i \leq n} |x_i|$.

The partial order $\leq$ on $\mathbb{R}^n$ is defined as usual by setting $\boldsymbol{x} \leq \boldsymbol{y}$ if $x_i \leq y_i$ for all $1 \leq i \leq n$. Similarly, $\boldsymbol{x} < \boldsymbol{y}$ if $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$. Finally, we write $\boldsymbol{x} \prec \boldsymbol{y}$ if $x_i < y_i$ for all $1 \leq i \leq n$, i.e., if every component of $\boldsymbol{x}$ is smaller than the corresponding component of $\boldsymbol{y}$.

We use $X_1, \ldots, X_n$ as variable identifiers and arrange them into the vector $\boldsymbol{X}$. In the following n always denotes the number of variables, i.e., the dimension of $\boldsymbol{X}$. While $\boldsymbol{x}, \boldsymbol{y}, \ldots$ denote arbitrary elements in $\mathbb{R}^n$ or $\mathbb{R}^n_{\geq 0}$, we write $\boldsymbol{X}$ if we want to emphasize that a function is given w.r.t. these variables. Hence, $\boldsymbol{f}(\boldsymbol{X})$ represents the function itself, whereas $\boldsymbol{f}(\boldsymbol{x})$ denotes its value for some $\boldsymbol{x} \in \mathbb{R}^n$.

If $S \subseteq \{1, \ldots, n\}$ is a set of components and $\boldsymbol{x}$ a vector, then by $\boldsymbol{x}_S$ we mean the vector obtained by restricting $\boldsymbol{x}$ to the components in S.

Let $S \subseteq \{1, \ldots, n\}$ and $\overline{S} = \{1, \ldots, n\} \setminus S$. Given a function $\boldsymbol{f}(\boldsymbol{X})$ and a vector $\boldsymbol{x}_S$, then $\boldsymbol{f}[S/\boldsymbol{x}_S]$ is obtained by replacing, for each $s \in S$, each occurrence of $\boldsymbol{X}_s$ by $\boldsymbol{x}_s$ and removing the s-component. In other words, if $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{f}(\boldsymbol{X}_S, \boldsymbol{X}_{\overline{S}})$, then $\boldsymbol{f}[S/\boldsymbol{x}_S](\boldsymbol{y}_{\overline{S}}) = \boldsymbol{f}_{\overline{S}}(\boldsymbol{x}_S, \boldsymbol{y}_{\overline{S}})$. For instance,

$$\text{if } \boldsymbol{f}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} X_1X_2 + 0.5 \\ X_2^2 + 0.2 \end{pmatrix}, \text{ then } \boldsymbol{f}[\{2\}/0.5] : \mathbb{R} \to \mathbb{R},\ X_1 \mapsto 0.5\, X_1 + 0.5\,.$$

$\mathbb{R}^{m \times n}$ denotes the set of matrices having m rows and n columns. The transpose of a vector or matrix is indicated by the superscript $^\top$. The identity matrix of $\mathbb{R}^{n \times n}$ is denoted by I.

The *matrix star* (or Neumann series) of $A \in \mathbb{R}^{n \times n}$ is defined by $A^* = \sum_{k \in \mathbb{N}} A^k$. It is well-known [BP79] that A^* exists if and only if the spectral radius of A is less than 1, i.e., $\max\{|\lambda| \mid \lambda \text{ is an eigenvalue of } A\} < 1$. If A^* exists, then $A^* = (I - A)^{-1}$.

The partial derivative of a function $f(\boldsymbol{X}) : \mathbb{R}^n \to \mathbb{R}$ with respect to the variable X_i is denoted by $\partial_{X_i} f$. The *Jacobian* of a function $\boldsymbol{f}(\boldsymbol{X})$ with $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is the matrix $\boldsymbol{f}'(\boldsymbol{X})$ defined by

$$\boldsymbol{f}'(\boldsymbol{X}) = \begin{pmatrix} \partial_{X_1} f_1 & \ldots & \partial_{X_n} f_1 \\ \vdots & & \vdots \\ \partial_{X_1} f_m & \ldots & \partial_{X_n} f_m \end{pmatrix}.$$

1.1.2 Systems of Positive Polynomials

Definition 1.1. *A function $\boldsymbol{f}(\boldsymbol{X})$ with $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ is a* system of positive polynomials (SPP) *if every component $f_i(\boldsymbol{X})$ is a polynomial in the variables $X_1, \ldots, X_n$ with coefficients in $\mathbb{R}_{\geq 0}$. We call an SPP $\boldsymbol{f}(\boldsymbol{X})$* feasible *if $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{y})$ for some $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. An SPP is called* linear *(resp.* quadratic*) if all polynomials have degree at most* 1 *(resp.* 2*).*

Notice that every SPP $\boldsymbol{f}$ is monotone on $\mathbb{R}^n_{\geq 0}$, i.e., for $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{y}$ we have $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{y})$.

We will need the following lemma, a version of Taylor's theorem.

Lemma 1.2 (Taylor)**.** *Let $\boldsymbol{f}$ be an SPP and $\boldsymbol{x}, \boldsymbol{u} \geq \boldsymbol{0}$. Then*

$$\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x})\boldsymbol{u} \leq \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{u}) \leq \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x} + \boldsymbol{u})\boldsymbol{u}\,.$$

Proof. It suffices to show this for a multivariate polynomial $f(\boldsymbol{X})$ with nonnegative coefficients. Consider $g(t) = f(\boldsymbol{x} + t\boldsymbol{u})$. We then have

$$f(\boldsymbol{x} + \boldsymbol{u}) = g(1) = g(0) + \int_0^1 g'(s)\, ds = f(\boldsymbol{x}) + \int_0^1 f'(\boldsymbol{x} + s\boldsymbol{u})\boldsymbol{u}\, ds.$$

The result follows as $f'(\boldsymbol{x}) \leq f'(\boldsymbol{x} + s\boldsymbol{u}) \leq f'(\boldsymbol{x} + \boldsymbol{u})$ for $s \in [0, 1]$. □

Since every SPP is monotone and continuous, Kleene's fixed-point theorem (see e.g. [Kui97]) applies.

Theorem 1.3 (Kleene's fixed-point theorem)**.** *Every feasible SPP $\boldsymbol{f}$ has a least fixed point $\boldsymbol{\mu f}$ in $\mathbb{R}^n_{\geq 0}$, i.e., $\boldsymbol{\mu f} = \boldsymbol{f}(\boldsymbol{\mu f})$ and, in addition, $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{y})$ implies $\boldsymbol{\mu f} \leq \boldsymbol{y}$. Moreover, the sequence $(\boldsymbol{\kappa}_{\boldsymbol{f}}^{(k)})_{k \in \mathbb{N}}$ with $\boldsymbol{\kappa}_{\boldsymbol{f}}^{(k)} = \boldsymbol{f}^k(\boldsymbol{0})$ is monotonically increasing with respect to $\leq$ (i.e., $\boldsymbol{\kappa}_{\boldsymbol{f}}^{(k)} \leq \boldsymbol{\kappa}_{\boldsymbol{f}}^{(k+1)}$)) and converges to $\boldsymbol{\mu f}$.*

In the following we call $(\boldsymbol{\kappa}_{\boldsymbol{f}}^{(k)})_{k\in\mathbb{N}}$ the *Kleene sequence* of $\boldsymbol{f}$, and drop the subscript whenever $\boldsymbol{f}$ is clear from the context. Similarly, we write $\boldsymbol{\mu}$ instead of $\boldsymbol{\mu f}$.

An SPP $\boldsymbol{f}$ is *clean* if $\boldsymbol{\mu} \succ \mathbf{0}$. It is easy to see that, if $\kappa_i^{(n)} = 0$, we have $\kappa_i^{(k)} = 0$ for all $k \in \mathbb{N}$, which implies $\mu_i = 0$ by Theorem 1.3. So we can "clean" an SPP $\boldsymbol{f}$ in time linear in the size of $\boldsymbol{f}$ by determining the components i with $\kappa_i^{(n)} = 0$ and removing them.

Example 1.4. *Consider the following SPP equation $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$.*

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} \frac{1}{4}X_2 + \frac{3}{4}X_1^2 \\ \frac{1}{3}X_3 + \frac{2}{3}X_1 \\ \frac{1}{2}X_4 + \frac{1}{2} \\ X_4^2 \end{pmatrix}$$

The first Kleene iterates are

$$\boldsymbol{\kappa}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(1)} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(2)} = \begin{pmatrix} 0 \\ \frac{1}{6} \\ \frac{1}{2} \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(3)} = \begin{pmatrix} \frac{1}{24} \\ \frac{1}{6} \\ \frac{1}{2} \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(4)} = \begin{pmatrix} \frac{11}{256} \\ \frac{7}{36} \\ \frac{1}{2} \\ 0 \end{pmatrix},$$

so $\mu_4 = 0$ and $\mu_1, \mu_2, \mu_3 > 0$. Since, at this stage, we are only interested in whether the components of $\boldsymbol{\mu}$ are zero or not, we need not actually compute the exact values of $\boldsymbol{\kappa}^{(k)}$. Rather, the following abstraction suffices:

$$\boldsymbol{\kappa}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(1)} = \begin{pmatrix} 0 \\ 0 \\ > 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(2)} = \begin{pmatrix} 0 \\ > 0 \\ > 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(3)} = \begin{pmatrix} > 0 \\ > 0 \\ > 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(4)} = \begin{pmatrix} > 0 \\ > 0 \\ > 0 \\ 0 \end{pmatrix}$$

So, the clean version $\overline{\boldsymbol{f}}$ of $\boldsymbol{f}$ is obtained by removing component 4:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{4}X_2 + \frac{3}{4}X_1^2 \\ \frac{1}{3}X_3 + \frac{2}{3}X_1 \\ \frac{1}{2} \end{pmatrix}$$

Notation 1.5. *In the following, we always assume that an SPP $\boldsymbol{f}$ is clean and feasible. That is, whenever we write "SPP", we mean "clean and feasible SPP", unless explicitly stated otherwise.*

We will also need the notion of *dependence* between variables.

Definition 1.6. *Let $f(\boldsymbol{X})$ be a polynomial. We say, $f(\boldsymbol{X})$* contains *a variable X_i if $\partial_{X_i} f(\boldsymbol{X})$ is not the zero-polynomial.*

Definition 1.7 (dependence, scSPP)**.** *Let $\boldsymbol{f}(\boldsymbol{X})$ be an SPP. A component i* depends directly *on a component k if $f_i(\boldsymbol{X})$ contains X_k. A component i* depends *on k if either i depends directly on k or there is a component j such that i depends on j and j depends on k. The components $\{1, \ldots, n\}$ can be partitioned into strongly components (SCCs) where an SCC S is a maximal set of components such that each component in S depends on every other component in S. An SCC is called* trivial *if it consists of a single component that does not depend on itself. An SPP is* strongly connected *(short: an* scSPP*) if $\{1, \ldots, n\}$ is a non-trivial SCC.*

Example 1.8. *In the clean SPP $\overline{\boldsymbol{f}}$ from Example 1.4 with*

$$\overline{\boldsymbol{f}}(\overline{\boldsymbol{X}}) = \begin{pmatrix} \frac{1}{4}X_2 + \frac{3}{4}X_1^2 \\ \frac{1}{3}X_3 + \frac{2}{3}X_1 \\ \frac{1}{2} \end{pmatrix} ,$$

component 1 *depends on components* 1 *and* 2*, component* 2 *depends on components* 1 *and* 3*, and component* 3 *depends on no component. Hence, the SCCs are* $\{1,2\}$ *and* $\{3\}$*. The SCC* $\{3\}$ *is a trivial SCC.*

1.1.3 Convergence Speed

We will analyze the convergence speed of Newton's method. To this end we need the notion of *valid bits.*

Definition 1.9. *Let $\boldsymbol{f}$ be an SPP. A vector $\boldsymbol{x}$ has i* valid bits *of the least fixed point $\boldsymbol{\mu}$ if*

$$\frac{|\mu_j - x_j|}{|\mu_j|} \leq 2^{-i}$$

for every $1 \leq j \leq n$. Let $(\boldsymbol{x}^{(k)})_{k \in \mathbb{N}}$ be a sequence with $\mathbf{0} \leq \boldsymbol{x}^{(k)} \leq \boldsymbol{\mu}$. Then the convergence order $\beta : \mathbb{N} \to \mathbb{N}$ *of the sequence $(\boldsymbol{x}^{(k)})_{k \in \mathbb{N}}$ is defined as follows: $\beta(k)$ is the greatest natural number i such that $\boldsymbol{x}^{(k)}$ has i valid bits (or ∞ if such a greatest number does not exist). We will always mean the convergence order of the Newton sequence $(\boldsymbol{\nu}^{(k)})_{k \in \mathbb{N}}$, unless explicitly stated otherwise.*

According to Definition 1.9, a vector $\boldsymbol{x}$ has i valid bits of $\boldsymbol{\mu}$, if the binary representations of $\boldsymbol{x}$ and $\boldsymbol{\mu}$, rounded to i binary places in all components, coincide.

We say that a sequence has logarithmic, linear, exponential, etc. convergence order if the function $\beta(k)$ grows logarithmically, linearly, or exponentially in k, respectively. Example of sequences with logarithmic, linear, and exponential convergence order are given in Examples 0.4, 0.6, and 0.7, respectively.

1.1.4 Stochastic Models

As mentioned in the introduction, several problems concerning stochastic models can be reduced to problems about the least fixed point $\boldsymbol{\mu}$ of an SPP $\boldsymbol{f}$. In these cases, $\boldsymbol{\mu}$ is a vector of probabilities, and so $\boldsymbol{\mu} \leq \mathbf{1}$.

Probabilistic Pushdown Automata

Our study of SPPs was initially motivated by the verification of probabilistic pushdown automata. A *probabilistic pushdown automaton (pPDA)* is a tuple $\mathcal{P} = (Q, \Gamma, \delta, \mathit{Prob})$ where Q is a finite set of *control states*, Γ is a finite *stack alphabet*, $\delta \subseteq Q \times \Gamma \times Q \times \Gamma^*$ is a finite *transition relation* (we write $pX \hookrightarrow q\alpha$ instead of $(p, X, q, \alpha) \in \delta$), and Prob is a function which to each transition $pX \hookrightarrow q\alpha$ assigns its probability $\mathit{Prob}(pX \hookrightarrow q\alpha) \in (0, 1]$ so that for all $p \in Q$ and $X \in \Gamma$ we have $\sum_{pX \hookrightarrow q\alpha} \mathit{Prob}(pX \hookrightarrow q\alpha) = 1$. We write $pX \overset{x}{\hookrightarrow} q\alpha$ instead of $\mathit{Prob}(pX \hookrightarrow q\alpha) = x$. A *configuration* of $\mathcal{P}$ is a pair qw, where q is a control state and $w \in \Gamma^*$ is a *stack content.* A probabilistic pushdown automaton $\mathcal{P}$ naturally induces a possibly infinite Markov chain with the configurations as states and transitions given by:

$pX\beta \overset{x}{\hookrightarrow} q\alpha\beta$ for every $\beta \in \Gamma^*$ iff $pX \overset{x}{\hookrightarrow} q\alpha$. We assume w.l.o.g. that if $pX \overset{x}{\hookrightarrow} q\alpha$ is a transition then $|\alpha| \leq 2$.

pPDAs and the equivalent model of recursive Markov chains have been very thoroughly studied [EKM04, BKS05, EY09, EY05a, EKM05, EY05b, EY05c]. This work has shown that the key to the analysis of pPDAs are the *termination probabilities* $[pXq]$, where p and q are states, and X is a stack letter, defined as follows (see e.g. [EKM04] for a more formal definition): $[pXq]$ is the probability that, starting at the configuration pX, the pPDA eventually reaches the configuration $q\varepsilon$ (empty stack). It is not difficult to show that the vector of these probabilities is the least solution of the SPP equation system containing the equation

$$\langle pXq\rangle = \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot \sum_{t\in Q} \langle rYt\rangle \cdot \langle tZq\rangle \quad + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot \langle rYq\rangle \quad + \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x$$

for each triple (p, X, q). Call this quadratic SPP the *termination SPP* of the pPDA (we assume that termination SPPs are clean, and it is easy to see that they are always feasible).

Example 1.10. *We model the spread of a disease using a simple probabilistic pushdown automaton $(Q, \Gamma, \delta, \mathit{Prob})$ with $Q = \{\mathit{res}, \mathit{eff}\}$, $\Gamma = \{X\}$ and δ, Prob as follows.*

$$\begin{array}{ll}
\mathit{res}\ X \overset{0.7}{\hookrightarrow} \mathit{res}\ XX & \qquad \mathit{eff}\ X \overset{0.3}{\hookrightarrow} \mathit{eff}\ XX \\
\mathit{res}\ X \overset{0.2}{\hookrightarrow} \mathit{res}\ \varepsilon & \qquad \mathit{eff}\ X \overset{0.6}{\hookrightarrow} \mathit{eff}\ \varepsilon \\
\mathit{res}\ X \overset{0.1}{\hookrightarrow} \mathit{eff}\ X & \qquad \mathit{eff}\ X \overset{0.1}{\hookrightarrow} \mathit{res}\ X
\end{array}$$

So, all configurations have either the form res X^k or the form eff X^k for some $k \geq 0$. The control state eff in a configuration indicates that an effective medication against the disease is available, whereas the control state res indicates that there is no or no effective medication, because, e.g., the disease has developed a resistance against the medication. The number of X-symbols in the configuration models the number of infected people. The rules above model how the disease spreads, depending on the availability of effective medication, and how the availability of effective medication may change. If there is initially one infected person with no available medication, the termination probability $[\mathit{res}X\mathit{eff}]$ (resp. $[\mathit{res}X\mathit{res}]$) is the probability that the disease is finally eradicated, with effective medication available (resp. unavailable). The number $1 - [\mathit{res}X\mathit{eff}] - [\mathit{res}X\mathit{res}]$ can be understood as the probability of a pandemic. The termination probabilities are the least solution of the following system of equations:

$$\begin{aligned}
\langle \mathit{res}X\mathit{res}\rangle &= 0.7 \cdot (\langle \mathit{res}X\mathit{res}\rangle \cdot \langle \mathit{res}X\mathit{res}\rangle + \langle \mathit{res}X\mathit{eff}\rangle \cdot \langle \mathit{eff}X\mathit{res}\rangle) \\
&\quad + \ 0.2 + 0.1\langle \mathit{eff}X\mathit{res}\rangle \\
\langle \mathit{res}X\mathit{eff}\rangle &= 0.7 \cdot (\langle \mathit{res}X\mathit{res}\rangle \cdot \langle \mathit{res}X\mathit{eff}\rangle + \langle \mathit{res}X\mathit{eff}\rangle \cdot \langle \mathit{eff}X\mathit{eff}\rangle) + 0.1 \cdot \langle \mathit{eff}X\mathit{eff}\rangle \\
\langle \mathit{eff}X\mathit{eff}\rangle &= 0.3 \cdot (\langle \mathit{eff}X\mathit{eff}\rangle \cdot \langle \mathit{eff}X\mathit{eff}\rangle + \langle \mathit{eff}X\mathit{res}\rangle \cdot \langle \mathit{res}X\mathit{eff}\rangle) \\
&\quad + \ 0.6 + 0.1\langle \mathit{res}X\mathit{eff}\rangle \\
\langle \mathit{eff}X\mathit{res}\rangle &= 0.3 \cdot (\langle \mathit{eff}X\mathit{eff}\rangle \cdot \langle \mathit{eff}X\mathit{res}\rangle + \langle \mathit{eff}X\mathit{res}\rangle \cdot \langle \mathit{res}X\mathit{res}\rangle) + 0.1 \cdot \langle \mathit{res}X\mathit{res}\rangle
\end{aligned}$$

The results of this chapter show that the termination probabilities can be efficiently approximated using Newton's method.

Strict pPDAs and Back-Button Processes

A pPDA is *strict* if for all $pX \in Q \times \Gamma$ and all $q \in Q$ the transition relation contains a pop-rule $pX \overset{x}{\hookrightarrow} q\epsilon$ for some $x > 0$. Essentially, strict pPDAs model programs in which every procedure has at least one terminating execution that does not call any other procedure. The termination SPP of a strict pPDA satisfies $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$.

In [FKK$^+$00, FKK$^+$01] a class of stochastic processes is introduced to model the behavior of web-surfers who from the current webpage P can decide either to follow a link to another page, say Q, with probability ℓ_{PQ}, or to press the "back button" with nonzero probability b_P (see Example 0.1 on page 3). These back-button processes correspond to a very special class of strict pPDAs having one single control state (which in the following we omit), and rules of the form $P \xhookrightarrow{b_P} \varepsilon$ (press the back button from P) or $P \xhookrightarrow{\ell_{PQ}} QP$ (follow the link from P to Q, remembering P as destination of pressing the back button at Q). The termination probabilities are given by an SPP equation system containing the equation

$$\langle P \rangle \quad = \quad b_P + \sum_{P \xhookrightarrow{\ell_{PQ}} QP} \ell_{PQ} \langle Q \rangle \langle P \rangle \quad = \quad b_P + \langle P \rangle \sum_{P \xhookrightarrow{\ell_{PQ}} QP} \ell_{PQ} \langle Q \rangle$$

for every webpage P. In [FKK$^+$00, FKK$^+$01] those termination probabilities are called *revocation* probabilities. The revocation probability of a page P is the probability that, when currently visiting webpage P and having $H_n H_{n-1} \ldots H_1$ as the stack of previously visited pages in the browser history, then during subsequent surfing from P the web-surfer eventually returns to webpage H_n with $H_{n-1} H_{n-2} \ldots H_1$ as the remaining browser history.

1.2 Newton's Method and an Overview of Our Results

In order to approximate the least fixed point $\boldsymbol{\mu}$ of an SPP $\boldsymbol{f}$ we employ Newton's method:

Definition 1.11. *Let $\boldsymbol{f}$ be an SPP. The Newton operator $\mathcal{N}_{\boldsymbol{f}}$ is defined as follows:*

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{X}) := \boldsymbol{X} + \left(I - \boldsymbol{f}'(\boldsymbol{X})\right)^{-1} (\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X})$$

The sequence $(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})_{k \in \mathbb{N}}$ with $\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)} = \mathcal{N}_{\boldsymbol{f}}^{k}(\mathbf{0})$ is called Newton sequence. *We drop the subscript of $\mathcal{N}_{\boldsymbol{f}}$ and $\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)}$ when $\boldsymbol{f}$ is understood.*

The main results of this chapter concern the application of Newton's method to SPPs. We summarize them in this section.

Theorem 1.12 states that the Newton sequence $(\boldsymbol{\nu}^{(k)})_{k \in \mathbb{N}}$ is well-defined (i.e., the inverse matrices $\left(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})\right)^{-1}$ exist for every $k \in \mathbb{N}$), monotonically increasing and bounded from above by $\boldsymbol{\mu}$ (i.e. $\boldsymbol{\nu}^{(k)} \le \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \le \boldsymbol{\nu}^{(k+1)} \le \boldsymbol{\mu}$), and converges to $\boldsymbol{\mu}$. This theorem generalizes the result of Etessami and Yannakakis in [EY09] to arbitrary SPPs and to the ordinary Newton's method.

For more quantitative results on the convergence speed it is convenient to focus on quadratic SPPs. **Theorem 1.26** shows that any SPP can be syntactically transformed into a quadratic SPP without changing the least fixed point and without accelerating Newton's method. This means, one can perform Newton's method on the original (possibly non-quadratic) SPP and convergence will be at least as fast as for the corresponding quadratic SPP.

For quadratic SPPs, one iteration of Newton's method involves $\mathcal{O}(n^3)$ arithmetical operations and $\mathcal{O}(n^3)$ operations in the Blum-Shub-Smale model. Hence, any bound on the number of iterations needed to compute a given number of valid bits immediately leads to a bound on the number of operations. In § 1.4 we prove such bounds for *strongly connected* quadratic SPPs. We give different thresholds for the number of iterations, and show that when any of these thresholds is reached, Newton's method gains at least one valid bit for each iteration. More precisely, **Theorem 1.40** states the following. Let $\boldsymbol{f}$ be a quadratic scSPP, let μ_{min} and μ_{max} be the minimal and maximal component of $\boldsymbol{\mu}$, respectively, and let the coefficients of $\boldsymbol{f}$ be given as ratios of m-bit integers. Then $\beta(k_{\boldsymbol{f}} + i) \geq i$ holds for all $i \in \mathbb{N}$ and for any of the following choices of $k_{\boldsymbol{f}}$:

(1) $4mn + \lceil 3n \max\{0, -\log \mu_{min}\} \rceil$;

(2) $4mn2^n$;

(3) $7mn$ if $\boldsymbol{f}$ satisfies $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$;

(4) $2m(n+1)$ if $\boldsymbol{f}$ satisfies both $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$ and $\mu_{max} \leq 1$.

We further show that Newton iteration can also be used to obtain a sequence of *upper* approximations of $\boldsymbol{\mu}$. Those upper approximations converge to $\boldsymbol{\mu}$, asymptotically as fast as the Newton sequence. More precisely, **Theorem 1.43** states the following: Let $\boldsymbol{f}$ be a quadratic scSPP, let c_{min} be the smallest nonzero coefficient of $\boldsymbol{f}$, and let μ_{min} be the minimal component of $\boldsymbol{\mu}$. Further, for all Newton approximants $\boldsymbol{\nu}^{(k)}$ with $\boldsymbol{\nu}^{(k)} \succ \mathbf{0}$, let $\nu^{(k)}_{min}$ be the smallest coefficient of $\boldsymbol{\nu}^{(k)}$. Then

$$\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu} \leq \boldsymbol{\nu}^{(k)} + \left[\frac{\left\| \boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)} \right\|_\infty}{\left(c_{min} \cdot \min\{\nu^{(k)}_{min}, 1\} \right)^n} \right]$$

where $[s]$ denotes the vector $\boldsymbol{x}$ with $x_j = s$ for all $1 \leq j \leq n$.

In § 1.5 we turn to general (not necessarily strongly connected) SPPs. We show in **Theorem 1.51** that Newton's method converges linearly and give a bound on the convergence rate, i.e., the number of iterations that is asymptotically needed to gain one valid bit. More precisely, the theorem proves that for every quadratic SPP $\boldsymbol{f}$, there is a threshold $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\beta(k_{\boldsymbol{f}} + i \cdot n \cdot 2^n) \geq i$ for all $i \in \mathbb{N}$. That is, in the worst case $n \cdot 2^n$ extra iterations are needed in order to get one new valid bit. § 1.6 shows that the bound is essentially tight.

1.3 Fundamental Properties of Newton's Method

1.3.1 Effectiveness

Etessami and Yannakakis [EY09] suggested to use Newton's method for SPPs. More precisely, they showed that the sequence obtained by applying Newton's method to the equation system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ converges to $\boldsymbol{\mu}$ as long as $\boldsymbol{f}$ is strongly connected. We extend their result to arbitrary SPPs, thereby reusing and extending several proofs of [EY09].

In Definition 1.11 we defined the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ and the associated Newton sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$. In this section we prove the following fundamental theorem on the Newton sequence.

Theorem 1.12. *Let $\boldsymbol{f}$ be an SPP. Let the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ be defined as in Definition 1.11:*

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{X}) := \boldsymbol{X} + \left(I - \boldsymbol{f}'(\boldsymbol{X})\right)^{-1} (\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X})$$

(1) Then the Newton sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ with $\boldsymbol{\nu}^{(k)} = \mathcal{N}^k_{\boldsymbol{f}}(\mathbf{0})$ is well-defined (i.e., the matrix inverses exist), monotonically increasing, bounded from above by $\boldsymbol{\mu}$ (i.e. $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{\nu}^{(k+1)} \leq \boldsymbol{\mu}$), and converges to $\boldsymbol{\mu}$.

(2) We have $(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}))^{-1} = \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*$ for all $k \in \mathbb{N}$.
We also have $(I - \boldsymbol{f}'(\boldsymbol{x}))^{-1} = \boldsymbol{f}'(\boldsymbol{x})^*$ for all $\boldsymbol{x} \prec \boldsymbol{\mu}$.

The proof of Theorem 1.12 consists of three steps. In the first proof step we study a sequence generated by a somewhat weaker version of the Newton operator and obtain the following:

Proposition 1.13. *Let $\boldsymbol{f}$ be an SPP. Let the operator $\widehat{\mathcal{N}}_{\boldsymbol{f}}$ be defined as follows:*

$$\widehat{\mathcal{N}}_{\boldsymbol{f}}(\boldsymbol{X}) := \boldsymbol{X} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{X})^d (\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}) \right) .$$

Then the sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ with $\boldsymbol{\nu}^{(k)} := \widehat{\mathcal{N}}_{\boldsymbol{f}}^k(\boldsymbol{0})$ is monotonically increasing, bounded from above by $\boldsymbol{\mu}$ (i.e. $\boldsymbol{\nu}^{(k)} \le \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \le \boldsymbol{\nu}^{(k+1)} \le \boldsymbol{\mu}$) and converges to $\boldsymbol{\mu}$.

In a second proof step, we show another intermediary proposition, namely that the star of the Jacobian matrix $\boldsymbol{f}'$ converges for all Newton approximants:

Proposition 1.14. *The matrix series $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^* := I + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}) + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^2 + \cdots$ converges in $\mathbb{R}_{\ge 0}$ for all Newton approximants $\boldsymbol{\nu}^{(k)}$, i.e., there are no ∞ entries.*

In the final third step we show that Propositions 1.13 and 1.14 imply Theorem 1.12.

First Step

For the first proof step (i.e., the proof of Proposition 1.13) we will need the following generalization of Taylor's theorem.

Lemma 1.15. *Let $\boldsymbol{f}$ be an SPP, $d \in \mathbb{N}$, and $\boldsymbol{0} \le \boldsymbol{u}$, and $\boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{f}(\boldsymbol{x})$. Then*

$$\boldsymbol{f}^d(\boldsymbol{x} + \boldsymbol{u}) \ge \boldsymbol{f}^d(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x})^d \boldsymbol{u} .$$

In particular, by setting $\boldsymbol{u} := \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}$ we get

$$\boldsymbol{f}^{d+1}(\boldsymbol{x}) - \boldsymbol{f}^d(\boldsymbol{x}) \ge \boldsymbol{f}'(\boldsymbol{x})^d (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}) .$$

Proof. By induction on d. For $d = 0$ the statement is trivial. Let $d \ge 0$. Then, by Taylor's theorem (Lemma 1.2), we have:

$$\begin{aligned} \boldsymbol{f}^{d+1}(\boldsymbol{x} + \boldsymbol{u}) &= \boldsymbol{f}(\boldsymbol{f}^d(\boldsymbol{x} + \boldsymbol{u})) \\ &\ge \boldsymbol{f}(\boldsymbol{f}^d(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x})^d \boldsymbol{u}) && \text{(induction hypothesis)} \\ &\ge \boldsymbol{f}^{d+1}(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{f}^d(\boldsymbol{x}))\boldsymbol{f}'(\boldsymbol{x})^d \boldsymbol{u} && \text{(Lemma 1.2)} \\ &\ge \boldsymbol{f}^{d+1}(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x})^{d+1} \boldsymbol{u} && (\boldsymbol{f}^d(\boldsymbol{x}) \ge \boldsymbol{x}) \end{aligned}$$ □

Lemma 1.15 can be used to prove the following.

Lemma 1.16. *Let $\boldsymbol{f}$ be an SPP. Let $\boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{\mu}$ and $\boldsymbol{x} \le \boldsymbol{f}(\boldsymbol{x})$. Then*

$$\boldsymbol{x} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{x})^d (\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}) \right) \le \boldsymbol{\mu} .$$

Proof. Observe that

$$\lim_{d\to\infty} \boldsymbol{f}^d(\boldsymbol{x}) = \boldsymbol{\mu} \tag{1.1}$$

because $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{\mu}$ implies $\boldsymbol{f}^d(\mathbf{0}) \leq \boldsymbol{f}^d(\mathbf{0}) \leq \boldsymbol{\mu}$ and as $(\boldsymbol{f}^d(\mathbf{0}))_{d\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$ by Theorem 1.3, so does $(\boldsymbol{f}^d(\boldsymbol{x}))_{d\in\mathbb{N}}$. We have:

$$\begin{aligned}
\boldsymbol{x} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{x})^d(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})\right) &\leq \boldsymbol{x} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}^{d+1}(\boldsymbol{x}) - \boldsymbol{f}^d(\boldsymbol{x})\right) && \text{(Lemma 1.15)} \\
&= \lim_{d\to\infty} \boldsymbol{f}^d(\boldsymbol{x}) \\
&= \boldsymbol{\mu} && \text{(by (1.1))} \qquad \square
\end{aligned}$$

Now we can prove Proposition 1.13.

Proof of Proposition 1.13. First we prove the following inequality by induction on k:

$$\boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \tag{1.2}$$

The induction base ($k = 0$) is easy. For the step, let $k \geq 0$. Then

$$\begin{aligned}
\boldsymbol{\nu}^{(k+1)} &= \boldsymbol{\nu}^{(k)} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\right) \\
&= \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) + \sum_{d=1}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\right) \\
&= \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}) \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\right) \\
&\leq \boldsymbol{f}\left(\boldsymbol{\nu}^{(k)} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\right)\right) && \text{(Lemma 1.2)} \\
&= \boldsymbol{f}(\boldsymbol{\nu}^{(k+1)}) \,.
\end{aligned}$$

Using (1.2), the inequality $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu}$ follows from Lemma 1.16 by a straightforward induction proof. This implies $\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{f}(\boldsymbol{\mu}) = \boldsymbol{\mu}$. Further we have

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) &= \boldsymbol{\nu}^{(k)} + (\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}) \\
&\leq \boldsymbol{\nu}^{(k)} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\right) = \boldsymbol{\nu}^{(k+1)} \,.
\end{aligned} \tag{1.3}$$

So it remains to show that $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$. As we have already shown $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu}$ it suffices to prove $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\nu}^{(k)}$ because $(\boldsymbol{\kappa}^{(k)})_{k\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$ by Theorem 1.3. We proceed by induction on k. The induction base ($k = 0$) is easy. For the step, let $k \geq 0$. Then

$$\begin{aligned}
\boldsymbol{\kappa}^{(k+1)} &= \boldsymbol{f}(\boldsymbol{\kappa}^{(k)}) \\
&\leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) && \text{(induction hypothesis)} \\
&\leq \boldsymbol{\nu}^{(k+1)} && \text{(by (1.3))} \,. \qquad \square
\end{aligned}$$

This completes the first step towards the proof of Theorem 1.12.

Second Step

For the second proof step (i.e., the proof of Proposition 1.14) it is convenient to move to the *extended reals* $\mathbb{R}_{[0,\infty]}$, i.e., we extend $\mathbb{R}_{\geq 0}$ by an element ∞ such that addition satisfies $a + \infty = \infty + a = \infty$ for all $a \in \mathbb{R}_{\geq 0}$ and multiplication satisfies $0 \cdot \infty = \infty \cdot 0 = 0$ and $a \cdot \infty = \infty \cdot a = \infty$ for all $a \in \mathbb{R}_{\geq 0}$. In $\mathbb{R}_{[0,\infty]}$, one can rewrite $\widehat{\mathcal{N}}(\boldsymbol{\nu}^{(k)}) = \boldsymbol{\nu}^{(k)} + \sum_{d=0}^{\infty} \big(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\big)$ as $\boldsymbol{\nu}^{(k)} + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})$. Notice that Proposition 1.14 does not follow trivially from Proposition 1.13, because ∞ entries of $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*$ could be cancelled out by matching 0 entries of $\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}$.

For the proof of Proposition 1.14 we need several lemmata. In the following, if M is a matrix, we often write M^i_{jk} resp. M^*_{jk} when we mean $(M^i)_{jk}$ resp. $(M^*)_{jk}$.

The following lemma assures that a starred matrix has an ∞ entry if and only if it has an ∞ entry on the diagonal.

Lemma 1.17. *Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}_{\geq 0}$. Let A^* have an ∞ entry. Then A^* also has an ∞ entry on the diagonal, i.e., $A^*_{ii} = \infty$ for some $1 \leq i \leq n$.*

Proof. By induction on n. The base case $n = 1$ is clear. For $n > 1$ assume w.l.o.g. that $A^*_{1n} = \infty$. We have

$$A^*_{1n} = A^*_{11} \sum_{j=2}^{n} a_{1j} (A_{[2..n,2..n]})^*_{jn} \,, \tag{1.4}$$

where by $A_{[2..n,2..n]}$ we mean the square matrix obtained from A by erasing the first row and the first column. To see why (1.4) holds, think of A^*_{1n} as the sum of weights of paths from 1 to n in the complete graph over the vertices $\{1, \ldots, n\}$. The weight of a path P is the product of the weight of P's edges, and $a_{i_1 i_2}$ is the weight of the edge from i_1 to i_2. Each path P from 1 to n can be divided into two sub-paths P_1, P_2 as follows. The second sub-path P_2 is the suffix of P leading from 1 to n and not returning to 1. The first sub-path P_1, possibly empty, is chosen such that $P = P_1 P_2$. Now, the sum of weights of all possible P_1 equals A^*_{11}, and the sum of weights of all possible P_2 equals $\sum_{j=2}^{n} a_{1j}(A_{[2..n,2..n]})^*_{jn}$. So (1.4) holds.

As $A^*_{1n} = \infty$, it follows that either A^*_{11} or some $(A_{[2..n,2..n]})^*_{jn}$ equals ∞. In the first case, we are done. In the second case, by induction, there is an i such that $(A_{[2..n,2..n]})^*_{ii} = \infty$. But then also $A^*_{ii} = \infty$, because every entry of $(A_{[2..n,2..n]})^*$ is less or equal the corresponding entry of A^*. □

The following lemma treats the case that $\boldsymbol{f}$ is strongly connected (cf. [EY09]).

Lemma 1.18. *Let $\boldsymbol{f}$ be non-trivially strongly connected. Let $\boldsymbol{0} \leq \boldsymbol{x} \prec \boldsymbol{\mu}$. Then $\boldsymbol{f}'(\boldsymbol{x})^*$ does not have ∞ as an entry.*

Proof. By Theorem 1.3 the Kleene sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$. Furthermore, $\boldsymbol{\kappa}^{(i)} \prec \boldsymbol{\mu}$ holds for all i, because, as every component depends non-trivially on itself, any increase in any component results in an increase of the same component in a later Kleene approximant. So, we can choose a Kleene approximant $\boldsymbol{y} = \boldsymbol{\kappa}^{(i)}$ such that $\boldsymbol{x} \leq \boldsymbol{y} \prec \boldsymbol{\mu}$. Notice that $\boldsymbol{y} \leq \boldsymbol{f}(\boldsymbol{y})$. By monotonicity of $\boldsymbol{f}'$ it suffices to show that $\boldsymbol{f}'(\boldsymbol{y})^*$ does not have ∞ as an entry. By Lemma 1.15 (taking $\boldsymbol{x} := \boldsymbol{y}$ and $\boldsymbol{u} := \boldsymbol{\mu} - \boldsymbol{y}$) we have

$$\boldsymbol{f}'(\boldsymbol{y})^d(\boldsymbol{\mu} - \boldsymbol{y}) \leq \boldsymbol{\mu} - \boldsymbol{f}^d(\boldsymbol{y}) \,.$$

As $d \to \infty$, the right hand side converges to $\boldsymbol{0}$, because, by Kleene's theorem, $\boldsymbol{f}^d(\boldsymbol{y})$ converges to $\boldsymbol{\mu}$. So the left hand side also converges to $\boldsymbol{0}$. Since $\boldsymbol{\mu} - \boldsymbol{y} \succ \boldsymbol{0}$, every entry of $\boldsymbol{f}'(\boldsymbol{y})^d$ must converge to $\mathbf{0}$. Then, by standard facts about matrices (see e.g. Thm. 5.6.12 of [HJ85]), the spectral radius of $\boldsymbol{f}'(\boldsymbol{y})$ is less than 1, i.e., $|\lambda| < 1$ for all eigenvalues λ of $\boldsymbol{f}'(\boldsymbol{y})$. This, in turn, implies that the series $\boldsymbol{f}'(\boldsymbol{y})^* = I + \boldsymbol{f}'(\boldsymbol{y}) + \boldsymbol{f}'(\boldsymbol{y})^2 + \cdots$ converges in $\mathbb{R}_{\geq 0}$, see [LT85], page 531. In other words, $\boldsymbol{f}'(\boldsymbol{y})^*$ and hence $\boldsymbol{f}'(\boldsymbol{x})^*$ do not have ∞ as an entry. □

The following lemma states that Newton's method can only terminate in a component s after certain other components ℓ have reached μ_ℓ.

Lemma 1.19. *Let $1 \leq s, \ell \leq n$. Let the term $\boldsymbol{f}'(\boldsymbol{X})^*_{ss}$ contain the variable X_ℓ. Let $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{\mu}$ and $x_s < \mu_s$ and $x_\ell < \mu_\ell$. Then $\widehat{\mathcal{N}}(\boldsymbol{x})_s < \mu_s$.*

Proof. This proof follows closely a proof of [EY09]. Let $d \geq 0$ such that $\boldsymbol{f}'(\boldsymbol{X})^d_{ss}$ contains X_ℓ. Let $m' \geq 0$ such that $\boldsymbol{f}^{m'}(\boldsymbol{x}) \succ \mathbf{0}$ and $\boldsymbol{f}^{m'}(\boldsymbol{x})_\ell > x_\ell$. Such an m' exists because with Kleene's theorem the sequence $(\boldsymbol{f}^k(\boldsymbol{x}))_{k\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$. Notice that our choice of m' guarantees $\boldsymbol{f}'(\boldsymbol{f}^{m'}(\boldsymbol{x}))^d_{ss} > \boldsymbol{f}'(\boldsymbol{x})^d_{ss}$.

Now choose $m \geq m'$ such that $\boldsymbol{f}^{m+1}(\boldsymbol{x})_s > \boldsymbol{f}^m(\boldsymbol{x})_s$. Such an m exists because the sequence $(\boldsymbol{f}^k(\boldsymbol{x})_s)_{k\in\mathbb{N}}$ never reaches μ_s. This is because s depends on itself (since $\boldsymbol{f}'(\boldsymbol{X})^*_{ss}$ is not constant zero), and so every increase of the s-component results in an increase of the s-component in some later iteration of the Kleene sequence.

We have:

$$\begin{aligned}
\boldsymbol{f}^{d+m+1}(\boldsymbol{x}) - \boldsymbol{f}^{d+m}(\boldsymbol{x}) &\geq \boldsymbol{f}'(\boldsymbol{f}^m(\boldsymbol{x}))^d(\boldsymbol{f}^{m+1}(\boldsymbol{x}) - \boldsymbol{f}^m(\boldsymbol{x})) && \text{(Lemma 1.15)} \\
&\geq^* \boldsymbol{f}'(\boldsymbol{x})^d(\boldsymbol{f}^{m+1}(\boldsymbol{x}) - \boldsymbol{f}^m(\boldsymbol{x})) \\
&\geq \boldsymbol{f}'(\boldsymbol{x})^d\boldsymbol{f}'(\boldsymbol{x})^m(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}) && \text{(Lemma 1.15)} \\
&= \boldsymbol{f}'(\boldsymbol{x})^{d+m}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})
\end{aligned}$$

The inequality marked with $*$ (in the second line of the above inequality chain) is strict in the s-component, due to the choice of d and m above. So, with $b = d + m$ we have:

$$(\boldsymbol{f}^{b+1}(\boldsymbol{x}) - \boldsymbol{f}^b(\boldsymbol{x}))_s > (\boldsymbol{f}'(\boldsymbol{x})^b(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}))_s \tag{1.5}$$

Again by Lemma 1.15, inequality (1.5) holds for all $b \in \mathbb{N}$, but with $\geq$ instead of $>$. Therefore:

$$\begin{aligned}
\mu_s &= \Big(\boldsymbol{x} + \sum_{i=0}^{\infty}(\boldsymbol{f}^{i+1}(\boldsymbol{x}) - \boldsymbol{f}^i(\boldsymbol{x}))\Big)_s && \text{(Kleene)} \\
&> \big(\boldsymbol{x} + \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})\big)_s && \text{(inequality (1.5))} \\
&= \big(\widehat{\mathcal{N}}(\boldsymbol{x})\big)_s
\end{aligned}$$

□

Now we are ready to prove Proposition 1.14.

Proof of Proposition 1.14. Using Lemma 1.17 it is enough to show that $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*_{ss} \neq \infty$ for all s. If the s-component constitutes a trivial SCC then $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*_{ss} = 0 \neq \infty$. So we can assume in the following that the s-component belongs to a non-trivial SCC, say S. Let $\boldsymbol{X}_L$ be the set of variables that the term $\boldsymbol{f}'(\boldsymbol{X})^*_{ss}$ contains. For any $t \in S$ we have $\boldsymbol{f}'(\boldsymbol{X})^*_{ss} \geq \boldsymbol{f}'(\boldsymbol{X})^*_{st}\boldsymbol{f}'(\boldsymbol{X})^*_{tt}\boldsymbol{f}'(\boldsymbol{X})^*_{ts}$. Neither $\boldsymbol{f}'(\boldsymbol{X})^*_{st}$ nor $\boldsymbol{f}'(\boldsymbol{X})^*_{ts}$ is constant zero, because S is non-trivial. Therefore, $\boldsymbol{f}'(\boldsymbol{X})^*_{ss}$ contains all variables that $\boldsymbol{f}'(\boldsymbol{X})^*_{tt}$ contains, and vice versa, for all $t \in S$. So, $\boldsymbol{X}_L$ is, for all $t \in S$, exactly the set of variables that $\boldsymbol{f}'(\boldsymbol{X})^*_{tt}$ contains.

We distinguish two cases.

Case 1: There is a component $\ell \in L$ such that the sequence $(\nu_\ell^{(k)})_{k\in\mathbb{N}}$ does not terminate, i.e., $\nu_\ell^{(k)} < \mu_\ell$ holds for all k. Then, by Lemma 1.19, the sequence $(\nu_s^{(k)})_{k\in\mathbb{N}}$ cannot reach μ_s either. In fact, we have $\boldsymbol{\nu}_S^{(k)} \prec \boldsymbol{\mu}_S$. Let M denote the set of those components that the S-components depend on, but do not depend on S. In other words, M contains the components that are "lower" in the DAG of SCCs than S. Define $\boldsymbol{g}(\boldsymbol{X}_S) := \boldsymbol{f}_S(\boldsymbol{X})[M/\boldsymbol{\mu}_M]$. Then $\boldsymbol{g}(\boldsymbol{X}_S)$ is strongly connected with $\boldsymbol{\mu g} = \boldsymbol{\mu}_S$. As $\boldsymbol{\nu}_S^{(k)} \prec \boldsymbol{\mu g}$, Lemma 1.18 is applicable, so $\boldsymbol{g}'(\boldsymbol{\nu}_S^{(k)})^*$ does not have ∞ as an entry. With $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*_{SS} \le \boldsymbol{g}'(\boldsymbol{\nu}_S^{(k)})^*$, we get $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*_{ss} < \infty$, as desired.

Case 2: For all components $\ell \in L$ the sequence $(\nu_\ell^{(k)})_{k\in\mathbb{N}}$ terminates. Let $i \in \mathbb{N}$ the least number such that $\nu_\ell^{(i)} = \mu_\ell$ holds for all $\ell \in L$. By Lemma 1.19 we have $\nu_s^{(i)} < \mu_s$. But as, according to Proposition 1.13, $(\nu_s^{(k)})_{k\in\mathbb{N}}$ converges to μ_s, there must exist a $j \ge i$ such that $0 < \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(j)})^*(\boldsymbol{f}(\boldsymbol{\nu}^{(j)}) - \boldsymbol{\nu}^{(j)})\right)_s < \infty$. So there is a component u with $0 < \boldsymbol{f}'(\boldsymbol{\nu}^{(j)})^*_{su}(\boldsymbol{f}(\boldsymbol{\nu}^{(j)}) - \boldsymbol{\nu}^{(j)})_u < \infty$. This implies $0 < \boldsymbol{f}'(\boldsymbol{\nu}^{(j)})^*_{su} < \infty$, therefore also $\boldsymbol{f}'(\boldsymbol{\nu}^{(j)})^*_{ss} < \infty$. By monotonicity of $\boldsymbol{f}'$, we have $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*_{ss} \le \boldsymbol{f}'(\boldsymbol{\nu}^{(j)})^*_{ss} < \infty$ for all $k \le j$. On the other hand, since $\boldsymbol{f}'(\boldsymbol{X})^*_{ss}$ contains only L-variables and $\boldsymbol{\nu}_L^{(k)} = \boldsymbol{\mu}_L$ holds for all $k \ge j$, we also have $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*_{ss} = \boldsymbol{f}'(\boldsymbol{\nu}^{(j)})^*_{ss} < \infty$ for all $k \ge j$. □

This completes the second intermediary step towards the proof of Theorem 1.12.

Third and Final Step

Now we can use Proposition 1.13 and Proposition 1.14 to complete the proof of Theorem 1.12.

Proof of Theorem 1.12. By Proposition 1.14 the matrix $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*$ has no ∞ entries. Then we clearly have $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})) = I$, so $(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}))^{-1} = \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*$, which is the first claim of part (2) of the theorem. Hence, we also have

$$\begin{aligned}
\widehat{\mathcal{N}}(\boldsymbol{\nu}^{(k)}) &= \boldsymbol{\nu}^{(k)} + \sum_{d=0}^{\infty} \left(\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^d(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)})\right) \\
&= \boldsymbol{\nu}^{(k)} + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}) \\
&= \boldsymbol{\nu}^{(k)} + (I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)}))^{-1}(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}) \\
&= \mathcal{N}(\boldsymbol{\nu}^{(k)}) ,
\end{aligned}$$

so we can replace $\widehat{\mathcal{N}}$ by $\mathcal{N}$. Therefore, part (1) of the theorem is implied by Proposition 1.13. It remains to show $(I - \boldsymbol{f}'(\boldsymbol{x}))^{-1} = \boldsymbol{f}'(\boldsymbol{x})^*$ for all $\boldsymbol{x} \prec \boldsymbol{\mu}$. It suffices to show that $\boldsymbol{f}'(\boldsymbol{x})^*$ has no ∞ entries. By part (1) the sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$. So there is a k' such that $\boldsymbol{x} \le \boldsymbol{\nu}^{(k')}$. By Proposition 1.14, $\boldsymbol{f}'(\boldsymbol{\nu}^{(k')})^*$ has no ∞ entries, so, by monotonicity, $\boldsymbol{f}'(\boldsymbol{x})^*$ has no ∞ entries either. □

1.3.2 Monotonicity

We will use the following monotonicity property of the Newton operator for our convergence analysis.

Lemma 1.20 (Monotonicity of the Newton operator)**.** *Let $\boldsymbol{f}$ be an SPP. Let $\mathbf{0} \le \boldsymbol{x} \le \boldsymbol{y} \le \boldsymbol{f}(\boldsymbol{y}) \le \boldsymbol{\mu}$ and let $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$ exist. Then*

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \le \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y}) \; .$$

Proof. For $\boldsymbol{x} \le \boldsymbol{y}$ we have $\boldsymbol{f}'(\boldsymbol{x}) \le \boldsymbol{f}'(\boldsymbol{y})$ as every entry of $\boldsymbol{f}'(\boldsymbol{X})$ is a monotone polynomial. Hence, $\boldsymbol{f}'(\boldsymbol{x})^* \le \boldsymbol{f}'(\boldsymbol{y})^*$. With this at hand we get:

$$\begin{aligned}
\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y}) &= \boldsymbol{y} + \boldsymbol{f}'(\boldsymbol{y})^*(\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{y}) && \text{(Theorem 1.12)}\\
&\ge \boldsymbol{y} + \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{y}) && (\boldsymbol{f}'(\boldsymbol{y})^* \ge \boldsymbol{f}'(\boldsymbol{x})^*)\\
&\ge \boldsymbol{y} + \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) - \boldsymbol{y}) && \text{(Lemma 1.2)}\\
&= \boldsymbol{y} + \boldsymbol{f}'(\boldsymbol{x})^*((\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}) - (I - \boldsymbol{f}'(\boldsymbol{x}))(\boldsymbol{y} - \boldsymbol{x}))\\
&= \boldsymbol{y} + \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}) - (\boldsymbol{y} - \boldsymbol{x}) && (\boldsymbol{f}'(\boldsymbol{x})^* = \\
& && \quad (I - \boldsymbol{f}'(\boldsymbol{x}))^{-1})\\
&= \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) && \text{(Theorem 1.12)} \qquad \square
\end{aligned}$$

1.3.3 Exponential Convergence Order in the Nonsingular Case

If the matrix $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is nonsingular, Newton's method has exponential convergence order in the sense of Definition 1.9.[1] This is, in fact, a well known general property of Newton's method, see e.g. [OR70]. For completeness, we show that Newton's method for "nonsingular" SPPs has exponential convergence order, see Theorem 1.24 below.

Lemma 1.21. *Let $\boldsymbol{f}$ be an SPP. Let $\mathbf{0} \le \boldsymbol{x} \le \boldsymbol{\mu}$ such that $\boldsymbol{f}'(\boldsymbol{x})^*$ exists. Then there is a bilinear function $B : \mathbb{R}^n_{\ge 0} \times \mathbb{R}^n_{\ge 0} \to \mathbb{R}^n_{\ge 0}$ with*

$$\boldsymbol{\mu} - \mathcal{N}(\boldsymbol{x}) \le \boldsymbol{f}'(\boldsymbol{x})^* B(\boldsymbol{\mu} - \boldsymbol{x}, \boldsymbol{\mu} - \boldsymbol{x}) \; .$$

Proof. Write $\boldsymbol{d} := \boldsymbol{\mu} - \boldsymbol{x}$. By Taylor's theorem (cf. Lemma 1.2) we obtain

$$\boldsymbol{f}(\boldsymbol{x} + \boldsymbol{d}) \le \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}'(\boldsymbol{x})\boldsymbol{d} + B(\boldsymbol{d}, \boldsymbol{d}) \tag{1.6}$$

for the bilinear map $B(\boldsymbol{X}) := \boldsymbol{f}''(\boldsymbol{\mu})(\boldsymbol{X}, \boldsymbol{X})$, where $\boldsymbol{f}''(\boldsymbol{\mu})$ denotes the rank-3 tensor of the second partial derivatives evaluated at $\boldsymbol{\mu}$ [OR70]. We have:

$$\begin{aligned}
\boldsymbol{\mu} - \mathcal{N}(\boldsymbol{x}) &= \boldsymbol{d} - \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})\\
&= \boldsymbol{d} - \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{d} + \boldsymbol{f}(\boldsymbol{x}) - (\boldsymbol{x} + \boldsymbol{d}))\\
&= \boldsymbol{d} - \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{d} + \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{d})) && (\boldsymbol{x} + \boldsymbol{d} = \boldsymbol{\mu} = \boldsymbol{f}(\boldsymbol{\mu}))\\
&\le \boldsymbol{d} - \boldsymbol{f}'(\boldsymbol{x})^* \left(\boldsymbol{d} - \boldsymbol{f}'(\boldsymbol{x})\boldsymbol{d} - B(\boldsymbol{d}, \boldsymbol{d})\right) && \text{(by (1.6))}\\
&= \boldsymbol{d} - \boldsymbol{f}'(\boldsymbol{x})^* \left((I - \boldsymbol{f}'(\boldsymbol{x}))\boldsymbol{d} - B(\boldsymbol{d}, \boldsymbol{d})\right)\\
&= \boldsymbol{d} - \boldsymbol{d} + \boldsymbol{f}'(\boldsymbol{x})^* B(\boldsymbol{d}, \boldsymbol{d}) && (\boldsymbol{f}'(\boldsymbol{x})^* = (I - \boldsymbol{f}'(\boldsymbol{x}))^{-1})\\
&= \boldsymbol{f}'(\boldsymbol{x})^* B(\boldsymbol{d}, \boldsymbol{d}) && \square
\end{aligned}$$

[1] In numerical analysis, the terms "quadratic convergence" or "Q-quadratic convergence" are commonly used, see e.g. [OR70]. It means that the error e' of the new approximant is bounded by $c \cdot e^2$ where e is the error of the old approximant and $c > 0$ is some constant. "Quadratic convergence" implies exponential convergence order in the sense of Definition 1.9. We avoid the notion of "quadratic convergence" in the following.

Define for the following lemmata $\boldsymbol{\Delta}^{(k)} := \boldsymbol{\mu} - \boldsymbol{\nu}^{(k)}$, i.e., $\boldsymbol{\Delta}^{(k)}$ is the error after k Newton iterations. The following lemma bounds $\left\|\boldsymbol{\Delta}^{(k+1)}\right\|$ (see § 1.1.1 for notation) in terms of $\left\|\boldsymbol{\Delta}^{(k)}\right\|^2$ if $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is nonsingular.

Lemma 1.22. *Let $\boldsymbol{f}$ be an SPP such that $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is nonsingular. Then there is a constant $c > 0$ such that*

$$\left\|\boldsymbol{\Delta}^{(k+1)}\right\| \leq c \cdot \left\|\boldsymbol{\Delta}^{(k)}\right\|^2 \quad \textit{for all } k \in \mathbb{N}.$$

Proof. As $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is nonsingular, we have, by Theorem 1.12, $(I - \boldsymbol{f}'(\boldsymbol{x}))^{-1} = \boldsymbol{f}'(\boldsymbol{x})^*$ for all $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{\mu}$. By continuity, there is a $c_1 > 0$ such that $\left\|\boldsymbol{f}'(\boldsymbol{x})^*\right\| \leq c_1$ for all $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{\mu}$. Similarly, there is a $c_2 > 0$ such that $\|B(\boldsymbol{x}, \boldsymbol{x})\| \leq c_2 \|\boldsymbol{x}\|^2$ for all $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{\mu}$, because B is bilinear. So it follows from Lemma 1.21 that $\left\|\boldsymbol{\Delta}^{(k+1)}\right\| \leq c_1 c_2 \left\|\boldsymbol{\Delta}^{(k)}\right\|^2$. □

Lemma 1.22 can be used to show that the error $\boldsymbol{\Delta}^{(i)}$ decays double-exponentially in the nonsingular case:

Lemma 1.23. *Let $\boldsymbol{f}$ be an SPP such that $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is nonsingular. Then there is a constant $\widetilde{k}_{\boldsymbol{f}} \in \mathbb{N}$ such that for all $i \in \mathbb{N}$*

$$\left\|\boldsymbol{\Delta}^{(\widetilde{k}_{\boldsymbol{f}}+i)}\right\| \leq 2^{-2^i} \quad \textit{for all } i \in \mathbb{N}.$$

Proof. We can assume w.l.o.g. that $c \geq 1$ for the c from Lemma 1.22. As the $\boldsymbol{\Delta}^{(k)}$ converge to $\mathbf{0}$, we can choose $\widetilde{k}_{\boldsymbol{f}} \in \mathbb{N}$ large enough such that $d := -\log\left\|\boldsymbol{\Delta}^{(\widetilde{k}_{\boldsymbol{f}})}\right\| - \log c \geq 1$. As $c, d \geq 1$, it suffices to show the following inequality:

$$\left\|\boldsymbol{\Delta}^{(\widetilde{k}_{\boldsymbol{f}}+i)}\right\| \leq \frac{2^{-d \cdot 2^i}}{c}\,.$$

We proceed by induction on i. For $i = 0$, the inequality above follows from the definition of d. Let $i \geq 0$. Then

$$\begin{aligned}
\left\|\boldsymbol{\Delta}^{(\widetilde{k}_{\boldsymbol{f}}+i+1)}\right\| &\leq c \cdot \left\|\boldsymbol{\Delta}^{(\widetilde{k}_{\boldsymbol{f}}+i)}\right\|^2 && \text{(Lemma 1.22)}\\
&\leq c \cdot \frac{2^{-d \cdot 2^i \cdot 2}}{c^2} && \text{(induction hypothesis)}\\
&= \frac{2^{-d \cdot 2^{i+1}}}{c}\,.
\end{aligned}$$

□

Now it follows easily that Newton's method has an exponential convergence order in the nonsingular case. More precisely:

Theorem 1.24. *Let $\boldsymbol{f}$ be an SPP such that $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is nonsingular. Then there is a constant $k_{\boldsymbol{f}} \in \mathbb{N}$ such that*

$$\beta(k_{\boldsymbol{f}} + i) \geq 2^i \quad \textit{for all } i \in \mathbb{N}.$$

Proof. Choose $m \in \mathbb{N}$ large enough such that $2^{m+i} + \log(\mu_j) \geq 2^i$ holds for all components j. Thus

$$\begin{aligned}
\Delta_j^{(\widetilde{k}_{\boldsymbol{f}}+m+i)}/\mu_j &\leq 2^{-2^{m+i}}/\mu_j && \text{(Lemma 1.23 with } \|\cdot\|_\infty\text{-norm)} \\
&= 2^{-(2^{m+i}+\log(\mu_j))} \\
&\leq 2^{-2^i} && \text{(choice of } m\text{)} .
\end{aligned}$$

So, with $k_{\boldsymbol{f}} := \widetilde{k}_{\boldsymbol{f}} + m$, the approximant $\boldsymbol{\nu}^{(k_{\boldsymbol{f}}+i)}$ has at least 2^i valid bits of $\boldsymbol{\mu}$. □

This type of analysis has severe shortcomings. In particular, Theorem 1.24 excludes the case where $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is singular. We will include this case in our convergence analysis in § 1.4 and § 1.5. Furthermore, and maybe even more severely, Theorem 1.24 does not give any bound on $k_{\boldsymbol{f}}$. We solve this problem for strongly connected SPPs in § 1.4.

1.3.4 Reduction to the Quadratic Case

In this section we reduce SPPs to quadratic SPPs, i.e., to SPPs in which every polynomial $f_i(\boldsymbol{X})$ has degree at most 2, and show that the convergence on the quadratic SPP is no faster than on the original SPP. In the following sections we will obtain convergence speed guarantees of Newton's method on quadratic SPPs. Hence, one can perform Newton's method on the original SPP and, using the results of this section, convergence is at least as fast as on the corresponding quadratic SPP.

The idea to reduce the degree of our SPP $\boldsymbol{f}$ is to introduce auxiliary variables that express quadratic subterms. This can be done repeatedly until all polynomials in the system have reached degree at most 2. The construction is very similar to the one that transforms a context-free grammar into another grammar in Chomsky normal form.

Example 1.25. *Consider the following equation system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ where $\boldsymbol{f}(\boldsymbol{X})$ is a non-quadratic SPP:*

$$\begin{aligned}
X_1 &= \frac{1}{2} X_2^3 + \frac{1}{2} \\
X_2 &= \frac{1}{3} X_1^2 X_2 + \frac{2}{3}
\end{aligned}$$

This equation system can be transformed into the following equation system $\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{f}}(\widetilde{\boldsymbol{X}})$ where $\widetilde{\boldsymbol{f}}(\widetilde{\boldsymbol{X}})$ is a quadratic SPP:

$$\begin{aligned}
X_1 &= \frac{1}{2} X_2 Y_1 + \frac{1}{2} \\
X_2 &= \frac{1}{3} X_1 Y_2 + \frac{2}{3} \\
Y_1 &= X_2^2 \\
Y_2 &= X_1 X_2
\end{aligned}$$

Those two equation systems have the same solutions, i.e., for each solution of the one equation system there is a corresponding solution of the other equation system that coincides in the X_1- and the X_2-component.

The following theorem shows that this transformation does not accelerate the convergence of Newton's method.

Theorem 1.26. *Let $\boldsymbol{f}(\boldsymbol{X})$ be an SPP such that $f_s(\boldsymbol{X}) = g(\boldsymbol{X}) + h(\boldsymbol{X})X_iX_j$ for some $1 \le i, j, s \le n$, where $g(\boldsymbol{X})$ and $h(\boldsymbol{X})$ are polynomials with nonnegative coefficients. Let $\widetilde{\boldsymbol{f}}(\boldsymbol{X}, Y)$ be the SPP given by*

$$\begin{array}{rcll} \widetilde{f}_\ell(\boldsymbol{X},Y) & = & f_\ell(\boldsymbol{X}) & \textit{for every } \ell \in \{1,\dots,s-1\} \\ \widetilde{f}_s(\boldsymbol{X},Y) & = & g(\boldsymbol{X}) + h(\boldsymbol{X})Y & \\ \widetilde{f}_\ell(\boldsymbol{X},Y) & = & f_\ell(\boldsymbol{X}) & \textit{for every } \ell \in \{s+1,\dots,n\} \\ \widetilde{f}_{n+1}(\boldsymbol{X},Y) & = & X_iX_j. & \end{array}$$

Then the function $b : \mathbb{R}^n \to \mathbb{R}^{n+1}$ given by $b(\boldsymbol{X}) = (X_1, \dots, X_n, X_iX_j)^\top$ is a bijection between the set of fixed points of $\boldsymbol{f}(\boldsymbol{X})$ and $\widetilde{\boldsymbol{f}}(\boldsymbol{X}, Y)$. Moreover, $\widetilde{\boldsymbol{\nu}}^{(k)} \le (\nu_1^{(k)}, \dots, \nu_n^{(k)}, \nu_i^{(k)}\nu_j^{(k)})^\top$ for all $k \in \mathbb{N}$, where $\widetilde{\boldsymbol{\nu}}^{(k)}$ and $\boldsymbol{\nu}^{(k)}$ are the Newton approximants of $\widetilde{\boldsymbol{f}}$ and $\boldsymbol{f}$, respectively.

Proof. We first show the claim regarding b: if $\boldsymbol{x}$ is a fixed point of $\boldsymbol{f}$, then $b(\boldsymbol{x}) = (\boldsymbol{x}, x_ix_j)^\top$ is a fixed point of $\widetilde{\boldsymbol{f}}$. Conversely, if $(\boldsymbol{x}, y)^\top$ is a fixed point of $\widetilde{\boldsymbol{f}}$, then we have $y = x_ix_j$ implying that $\boldsymbol{x}$ is a fixed point of $\boldsymbol{f}$. Therefore, the least fixed point $\boldsymbol{\mu}$ of $\boldsymbol{f}$ determines $\boldsymbol{\mu}\widetilde{\boldsymbol{f}}$, and vice versa.

Now we show that the Newton sequence of $\boldsymbol{f}$ converges at least as fast as the Newton sequence of $\widetilde{\boldsymbol{f}}$. In the following we write $\boldsymbol{Y}$ for the $(n+1)$-dimensional vector of variables $(X_1, \dots, X_n, Y)^\top$ and, as usual, $\boldsymbol{X}$ for $(X_1, \dots, X_n)^\top$. For an $(n+1)$-dimensional vector $\boldsymbol{x}$, we let $\boldsymbol{x}_{[1,n]}$ denote its restriction to the n first components, i.e., $\boldsymbol{x}_{[1,n]} := (x_1, \dots, x_n)^\top$. Note that $\boldsymbol{Y}_{[1,n]} = \boldsymbol{X}$. Let $\boldsymbol{e}_s$ denote the unit vector $(0, \dots, 0, 1, 0, \dots, 0)^\top$, where the "1" is on the s-th place. We have

$$\widetilde{\boldsymbol{f}}(\boldsymbol{Y}) = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{e}_s h(\boldsymbol{X})(Y - X_iX_j) \\ X_iX_j \end{pmatrix}$$

and

$$\widetilde{\boldsymbol{f}}'(\boldsymbol{Y}) = \begin{pmatrix} \boldsymbol{f}'(\boldsymbol{X}) + \boldsymbol{e}_s \partial_{\boldsymbol{X}} h(\boldsymbol{X})(Y - X_iX_j) & \boldsymbol{e}_s h(\boldsymbol{X}) \\ \partial_{\boldsymbol{X}} X_iX_j & 0 \end{pmatrix} .$$

We need the following lemma.

Lemma 1.27. *Let $\boldsymbol{z} \in \mathbb{R}^n_{\ge 0}$, $\boldsymbol{\delta} = \big(I - \boldsymbol{f}'(\boldsymbol{z})\big)^{-1}(\boldsymbol{f}(\boldsymbol{z}) - \boldsymbol{z})$ and*

$$\widetilde{\boldsymbol{\delta}} = \left(I - \widetilde{\boldsymbol{f}}'\begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix}\right)^{-1} \left(\widetilde{\boldsymbol{f}}\begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix} - \begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix}\right) .$$

Then $\boldsymbol{\delta} = \widetilde{\boldsymbol{\delta}}_{[1,n]}$.

Proof of the lemma.

$$\begin{aligned} \widetilde{\boldsymbol{f}}'\begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix} &= \begin{pmatrix} \boldsymbol{f}'(\boldsymbol{z}) + \boldsymbol{e}_s h(\boldsymbol{z}) \partial_{\boldsymbol{X}}(Y - X_iX_j)|_{\boldsymbol{Y}=(\boldsymbol{z}, z_iz_j)^\top} & \boldsymbol{e}_s h(\boldsymbol{z}) \\ \partial_{\boldsymbol{X}} X_iX_j|_{\boldsymbol{Y}=(\boldsymbol{z}, z_iz_j)^\top} & 0 \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{f}'(\boldsymbol{z}) - \boldsymbol{e}_s h(\boldsymbol{z}) \partial_{\boldsymbol{X}}(X_iX_j)|_{\boldsymbol{X}=\boldsymbol{z}} & \boldsymbol{e}_s h(\boldsymbol{z}) \\ \partial_{\boldsymbol{X}} X_iX_j|_{\boldsymbol{X}=\boldsymbol{z}} & 0 \end{pmatrix} \end{aligned}$$

We have

$$\left(I - \widetilde{\boldsymbol{f}}'\begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix}\right)\widetilde{\boldsymbol{\delta}} = \widetilde{\boldsymbol{f}}\begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix} - \begin{pmatrix} \boldsymbol{z} \\ z_iz_j \end{pmatrix} ,$$

or equivalently:

$$\begin{pmatrix} I - \boldsymbol{f}'(\boldsymbol{z}) + \boldsymbol{e}_s h(\boldsymbol{z}) \partial_{\boldsymbol{X}} (X_i X_j)|_{\boldsymbol{X}=\boldsymbol{z}} & -\boldsymbol{e}_s h(\boldsymbol{z}) \\ -\partial_{\boldsymbol{X}} X_i X_j|_{\boldsymbol{X}=\boldsymbol{z}} & 1 \end{pmatrix} \cdot \begin{pmatrix} \widetilde{\boldsymbol{\delta}}_{[1,n]} \\ \widetilde{\delta}_{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{z}) - \boldsymbol{z} \\ 0 \end{pmatrix} .$$

Multiplying the last row by $\boldsymbol{e}_s h(\boldsymbol{z})$ and adding to the first n rows yields:

$$\left(I - \boldsymbol{f}'(\boldsymbol{z})\right) \widetilde{\boldsymbol{\delta}}_{[1,n]} = \boldsymbol{f}(\boldsymbol{z}) - \boldsymbol{z}$$

So we have $\widetilde{\boldsymbol{\delta}}_{[1,n]} = \left(I - \boldsymbol{f}'(\boldsymbol{z})\right)^{-1} (\boldsymbol{f}(\boldsymbol{z}) - \boldsymbol{z}) = \boldsymbol{\delta}$, which proves the lemma. □

Now we proceed by induction on k to show $\widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]} \leq \boldsymbol{\nu}^{(k)}$, where $\widetilde{\boldsymbol{\nu}}^{(k)}$ is the Newton sequence for $\widetilde{\boldsymbol{f}}$. By definition of the Newton sequence this is true for $k = 0$. For the step, let $k \geq 0$ and define $\boldsymbol{u} := (\widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]}, \widetilde{\nu}^{(k)}_i \cdot \widetilde{\nu}^{(k)}_j)^\top$. Then we have:

$$\begin{aligned}
\widetilde{\boldsymbol{\nu}}^{(k+1)}_{[1,n]} &= \mathcal{N}_{\widetilde{\boldsymbol{f}}}(\widetilde{\boldsymbol{\nu}}^{(k)})_{[1,n]} \\
&\leq \mathcal{N}_{\widetilde{\boldsymbol{f}}}(\boldsymbol{u})_{[1,n]} && \text{(see below)} \\
&= \widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]} + \left((I - \widetilde{\boldsymbol{f}}'(\boldsymbol{u}))^{-1}(\widetilde{\boldsymbol{f}}(\boldsymbol{u}) - \boldsymbol{u})\right)_{[1,n]} \\
&= \widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]} + (I - \boldsymbol{f}'(\widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]}))^{-1}(\boldsymbol{f}(\widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]}) - \widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]}) && \text{(Lemma 1.27)} \\
&= \mathcal{N}_{\boldsymbol{f}}(\widetilde{\boldsymbol{\nu}}^{(k)}_{[1,n]}) \\
&\leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\nu}^{(k)}) && \text{(induction)} \\
&= \boldsymbol{\nu}^{(k+1)}
\end{aligned}$$

For the inequality $\mathcal{N}_{\widetilde{\boldsymbol{f}}}(\widetilde{\boldsymbol{\nu}}^{(k)})_{[1,n]} \leq \mathcal{N}_{\widetilde{\boldsymbol{f}}}(\boldsymbol{u})_{[1,n]}$ we have used the monotonicity of $\mathcal{N}_{\widetilde{\boldsymbol{f}}}$ (Lemma 1.20) combined with Theorem 1.12, which states $\widetilde{\boldsymbol{\nu}}^{(k)} \leq \widetilde{\boldsymbol{f}}(\widetilde{\boldsymbol{\nu}}^{(k)})$, hence in particular $\widetilde{\nu}^{(k)}_{n+1} \leq \widetilde{\nu}^{(k)}_i \widetilde{\nu}^{(k)}_j$. This concludes the proof of Theorem 1.26. □

1.4 Strongly Connected SPPs

In this section we study the convergence speed of Newton's method on strongly connected SPPs, short scSPPs, see Definition 1.7.

1.4.1 Cone Vectors

Our convergence speed analysis makes crucial use of the existence of *cone vectors*.

Definition 1.28. *Let $\boldsymbol{f}$ be an SPP. A vector $\boldsymbol{d} \in \mathbb{R}^n_{\geq 0}$ is a* cone vector *if $\boldsymbol{d} \succ \boldsymbol{0}$ and $\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} \leq \boldsymbol{d}$.*

The following example illustrates the concept of cone vectors.

Example 1.29. *Consider again the 2-dimensional SPP $\boldsymbol{f} = (f_1, f_2)^\top$ from Figure 0.1 (page 4). Figure 1.1 below shows the graphs of the equations*

$$X_1 = f_1(X_1, X_2) \textit{ and } X_2 = f_2(X_1, X_2)$$

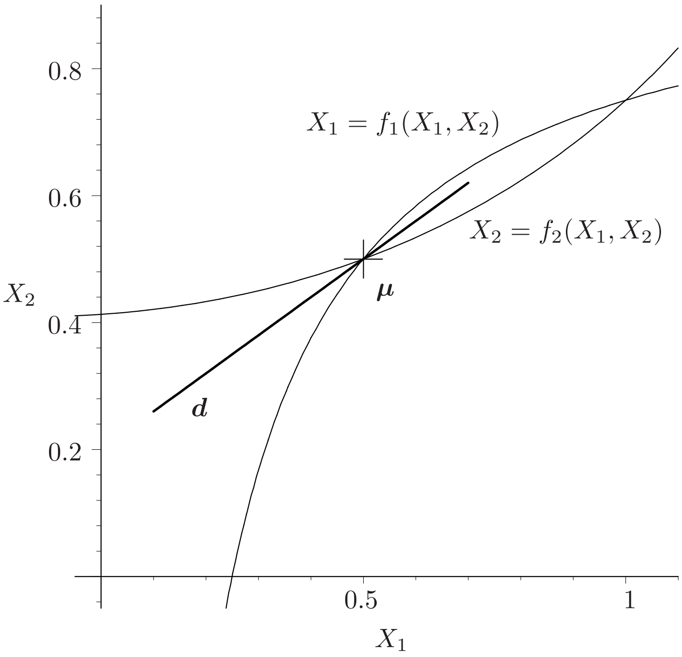

Figure 1.1: The graph of a 2-dimensional SPP equation along with a cone vector $\boldsymbol{d}$.

along with a cone vector $\boldsymbol{d} = (5,3)^\top$*. More precisely, the thick line in Figure 1.1 is the set of points* $\{\boldsymbol{\mu} + r\boldsymbol{d} \mid r \in \mathbb{R}\}$*, i.e., the straight line through* $\boldsymbol{\mu}$ *in the direction of* $\boldsymbol{d}$*. It is easy to check that* $\boldsymbol{d} = (5,3)^\top$ *is indeed a cone vector: Since*

$$f_1(X_1, X_2) = X_1X_2 + \frac{1}{4} \quad \text{and} \quad f_2(X_1, X_2) = \frac{1}{6}X_1^2 + \frac{1}{9}X_1X_2 + \frac{2}{9}X_2^2 + \frac{3}{8}$$

and $\boldsymbol{\mu} = (1/2,\ 1/2)^\top$*, we have*

$$\boldsymbol{f}'(\boldsymbol{\mu}) \cdot \boldsymbol{d} = \begin{pmatrix} 1/2 & 1/2 \\ 2/9 & 5/18 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ 35/18 \end{pmatrix} \leq \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \boldsymbol{d}.$$

Graphically, the straight line $\{\boldsymbol{\mu} + r\boldsymbol{d} \mid r \in \mathbb{R}\}$ *connects the "prefixed points" (i.e., the points* $\boldsymbol{x}$ *with* $\boldsymbol{f}(\boldsymbol{x}) \geq \boldsymbol{x}$*) with the "postfixed points" (i.e., the points* $\boldsymbol{x}$ *with* $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{x}$*). This can be seen as follows. By Taylor's theorem we have*

$$\boldsymbol{f}(\boldsymbol{\mu} + r\boldsymbol{d}) = \underbrace{\boldsymbol{f}(\boldsymbol{\mu})}_{\boldsymbol{\mu}} + r\underbrace{\boldsymbol{f}'(\boldsymbol{\mu}) \cdot \boldsymbol{d}}_{\leq \boldsymbol{d}} + \mathcal{O}(r^2),$$

i.e., for small negative r *we have* $\boldsymbol{f}(\boldsymbol{\mu} + r\boldsymbol{d}) \geq \boldsymbol{\mu} + r\boldsymbol{d}$*, and for small positive* r *we have* $\boldsymbol{f}(\boldsymbol{\mu} + r\boldsymbol{d}) \leq \boldsymbol{\mu} + r\boldsymbol{d}$*. In Figure 1.1, the "prefixed points" are at the lower left of* $\boldsymbol{\mu}$*, and the "postfixed points" are at the upper right of* $\boldsymbol{\mu}$*.*

We will show that any scSPP has a cone vector, see Proposition 1.32 below. As a first step, we show the following lemma.

Lemma 1.30. *Any scSPP* $\boldsymbol{f}$ *has a vector* $\boldsymbol{d} > \boldsymbol{0}$ *with* $\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} \leq \boldsymbol{d}$*.*

Proof. Consider the Kleene sequence $(\boldsymbol{\kappa}^{(k)})_{k\in\mathbb{N}}$. We have $\boldsymbol{0} \leq \boldsymbol{\kappa}^{(k)} \prec \boldsymbol{\mu}$ for all $k \in \mathbb{N}$. By Theorem 1.12.2., the matrices $(I - \boldsymbol{f}'(\boldsymbol{\kappa}^{(k)}))^{-1} = \boldsymbol{f}'(\boldsymbol{\kappa}^{(k)})^*$ exist for all k. Let $\|\cdot\|$ be any norm. Define the vectors

$$\boldsymbol{d}^{(k)} := \frac{\boldsymbol{f}'(\boldsymbol{\kappa}^{(k)})^*\boldsymbol{1}}{\left\|\boldsymbol{f}'(\boldsymbol{\kappa}^{(k)})^*\boldsymbol{1}\right\|}\,.$$

Notice that for all $k \in \mathbb{N}$ we have $(I - \boldsymbol{f}'(\boldsymbol{\kappa}^{(k)}))\boldsymbol{d}^{(k)} = \frac{\mathbf{1}}{\|\boldsymbol{f}'(\boldsymbol{\kappa}^{(k)})^{*}\mathbf{1}\|} \geq \mathbf{0}$. Furthermore we have $\boldsymbol{d}^{(k)} \in C$, where $C := \{\boldsymbol{x} \geq \mathbf{0} \mid \|\boldsymbol{x}\| = 1\}$ is compact. So the sequence $(\boldsymbol{d}^{(k)})_{k\in\mathbb{N}}$ has a convergent subsequence, whose limit, say $\boldsymbol{d}$, is also in C. In particular $\boldsymbol{d} > \mathbf{0}$. As $(\boldsymbol{\kappa}^{(k)})_{k\in\mathbb{N}}$ converges to $\boldsymbol{\mu}$ and $(I - \boldsymbol{f}'(\boldsymbol{\kappa}^{(k)}))\boldsymbol{d}^{(k)} \geq \mathbf{0}$, it follows by continuity $(I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{d} \geq \mathbf{0}$. □

Lemma 1.31. *Let $\boldsymbol{f}$ be an scSPP and let $\boldsymbol{d} > \mathbf{0}$ with $\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} \leq \boldsymbol{d}$. Then $\boldsymbol{d}$ is a cone vector, i.e., $\boldsymbol{d} \succ \mathbf{0}$.*

Proof. Since $\boldsymbol{f}$ is an SPP, every component of $\boldsymbol{f}'(\boldsymbol{\mu})$ is nonnegative. So,

$$\mathbf{0} \leq \boldsymbol{f}'(\boldsymbol{\mu})^{n}\boldsymbol{d} \leq \boldsymbol{f}'(\boldsymbol{\mu})^{n-1}\boldsymbol{d} \leq \ldots \leq \boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} \leq \boldsymbol{d}.$$

Let w.l.o.g. $d_1 > 0$. As $\boldsymbol{f}$ is strongly connected, there is for all j with $1 \leq j \leq n$ an $r_j \leq n$ such that $(\boldsymbol{f}'(\boldsymbol{\mu})^{r_j})_{j1} > 0$. Hence, $(\boldsymbol{f}'(\boldsymbol{\mu})^{r_j}\boldsymbol{d})_j > 0$ for all j. With above inequality chain, it follows that $d_j \geq (\boldsymbol{f}'(\boldsymbol{\mu})^{r_j}\boldsymbol{d})_j > 0$. So, $\boldsymbol{d} \succ \mathbf{0}$. □

The following proposition follows immediately by combining Lemmata 1.30 and 1.31.

Proposition 1.32. *Any scSPP has a cone vector.*

We remark that using Perron-Frobenius theory [BP79] there is a simpler proof for Proposition 1.32: By Theorem 1.12 $\boldsymbol{f}'(\boldsymbol{x})^{*}$ exists for all $\boldsymbol{x} \prec \boldsymbol{f}$. So, by fundamental matrix facts [BP79], the spectral radius of $\boldsymbol{f}'(\boldsymbol{x})$ is less than 1 for all $\boldsymbol{x} \prec \boldsymbol{\mu}$. As the eigenvalues of a matrix depend continuously on the matrix, the spectral radius of $\boldsymbol{f}'(\boldsymbol{\mu})$, say ρ, is at most 1. Since $\boldsymbol{f}$ is strongly connected, $\boldsymbol{f}'(\boldsymbol{\mu})$ is irreducible, and so Perron-Frobenius theory guarantees the existence of an eigenvector $\boldsymbol{d} \succ \mathbf{0}$ of $\boldsymbol{f}'(\boldsymbol{\mu})$ with eigenvalue ρ. So we have $\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} = \rho\boldsymbol{d} \leq \boldsymbol{d}$, i.e., the eigenvector $\boldsymbol{d}$ is a cone vector.

1.4.2 Convergence Speed in Terms of Cone Vectors

Now we show that cone vectors play a fundamental role for the convergence speed of Newton's method. The following lemma gives a lower bound of the Newton approximant $\boldsymbol{\nu}^{(1)}$ in terms of a cone vector.

Lemma 1.33. *Let $\boldsymbol{f}$ be a (not necessarily clean) SPP such that $\boldsymbol{f}'(\mathbf{0})^{*}$ exists. Let $\boldsymbol{d}$ be a cone vector of $\boldsymbol{f}$. Let $\mathbf{0} \geq \boldsymbol{\mu} - \lambda\boldsymbol{d}$ for some $\lambda \geq 0$. Then*

$$\mathcal{N}(\mathbf{0}) \geq \boldsymbol{\mu} - \frac{1}{2}\lambda\boldsymbol{d}\,.$$

Proof. We write $\boldsymbol{f}(\boldsymbol{X})$ as a sum

$$\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{c} + \sum_{k=1}^{D} L_k(\boldsymbol{X}, \ldots, \boldsymbol{X})\boldsymbol{X}$$

where D is the degree of $\boldsymbol{f}$, and every L_k is a $(k-1)$-linear map from $(\mathbb{R}^n)^{k-1}$ to $\mathbb{R}^{n\times n}$. Notice that $\boldsymbol{f}'(\boldsymbol{X}) = \sum_{k=1}^{D} k \cdot L_k(\boldsymbol{X}, \ldots, \boldsymbol{X})$. We write L for L_1, and $\boldsymbol{h}(\boldsymbol{X})$ for $\boldsymbol{f}(\boldsymbol{X}) - L\boldsymbol{X} - \boldsymbol{c}$.

$$\begin{aligned}
\frac{\lambda}{2}\boldsymbol{d} &= \frac{\lambda}{2}(L^*\boldsymbol{d} - L^*L\boldsymbol{d}) && (L^* = I + L^*L)\\
&\geq \frac{\lambda}{2}(L^*\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} - L^*L\boldsymbol{d}) && (\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} \leq \boldsymbol{d})\\
&= \frac{\lambda}{2}L^*\boldsymbol{h}'(\boldsymbol{\mu})\boldsymbol{d} && (\boldsymbol{f}'(\boldsymbol{x}) = \boldsymbol{h}'(\boldsymbol{x}) + L)\\
&= L^*\frac{1}{2}\boldsymbol{h}'(\boldsymbol{\mu})\lambda\boldsymbol{d}\\
&\geq L^*\frac{1}{2}\boldsymbol{h}'(\boldsymbol{\mu})\boldsymbol{\mu} && (\lambda\boldsymbol{d} \geq \boldsymbol{\mu})\\
&= L^*\frac{1}{2}\sum_{k=2}^{D} k \cdot L_k(\boldsymbol{\mu}, \ldots, \boldsymbol{\mu})\boldsymbol{\mu}\\
&\geq L^*\sum_{k=2}^{D} L_k(\boldsymbol{\mu}, \ldots, \boldsymbol{\mu})\boldsymbol{\mu}\\
&= L^*\boldsymbol{h}(\boldsymbol{\mu})\\
&= L^*(\boldsymbol{f}(\boldsymbol{\mu}) - L\boldsymbol{\mu} - \boldsymbol{c}) && (\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}) + L\boldsymbol{x} + \boldsymbol{c})\\
&= L^*\boldsymbol{\mu} - L^*L\boldsymbol{\mu} - L^*\boldsymbol{c} && (\boldsymbol{f}(\boldsymbol{\mu}) = \boldsymbol{\mu})\\
&= \boldsymbol{\mu} - L^*\boldsymbol{c} && (L^* = I + L^*L)\\
&= \boldsymbol{\mu} - \mathcal{N}(\boldsymbol{0}) && (\mathcal{N}(\boldsymbol{0}) = \boldsymbol{f}'(\boldsymbol{0})^*\boldsymbol{f}(\boldsymbol{0}) = L^*\boldsymbol{c}) \qquad \square
\end{aligned}$$

We extend Lemma 1.33 to arbitrary vectors $\boldsymbol{x}$ as follows.

Lemma 1.34. *Let $\boldsymbol{f}$ be a (not necessarily clean) SPP. Let $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{\mu}$ and $\boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x})$ such that $\boldsymbol{f}'(\boldsymbol{x})^*$ exists. Let $\boldsymbol{d}$ be a cone vector of $\boldsymbol{f}$. Let $\boldsymbol{x} \geq \boldsymbol{\mu} - \lambda\boldsymbol{d}$ for some $\lambda \geq 0$. Then*

$$\mathcal{N}(\boldsymbol{x}) \geq \boldsymbol{\mu} - \frac{1}{2}\lambda\boldsymbol{d}\,.$$

Proof. Define $\boldsymbol{g}(\boldsymbol{X}) := \boldsymbol{f}(\boldsymbol{X} + \boldsymbol{x}) - \boldsymbol{x}$. We first show that $\boldsymbol{g}$ is an SPP (not necessarily clean). The only coefficients of $\boldsymbol{g}$ that could be negative are those of degree 0. But we have $\boldsymbol{g}(\boldsymbol{0}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x} \geq \boldsymbol{0}$, and so these coefficients are also nonnegative.

It follows immediately from the definition that $\boldsymbol{\mu} - \boldsymbol{x} \geq \boldsymbol{0}$ is the least fixed point of $\boldsymbol{g}$. Moreover, $\boldsymbol{g}$ satisfies $\boldsymbol{g}'(\boldsymbol{\mu} - \boldsymbol{x})\boldsymbol{d} \leq \boldsymbol{d}$, and so $\boldsymbol{d}$ is also a cone vector of $\boldsymbol{g}$. Finally, we have $\boldsymbol{0} \geq \boldsymbol{\mu} - \boldsymbol{x} - \lambda\boldsymbol{d} = \boldsymbol{\mu}\boldsymbol{g} - \lambda\boldsymbol{d}$. So, Lemma 1.33 can be applied as follows.

$$\begin{aligned}
\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) &= \boldsymbol{x} + \boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})\\
&= \boldsymbol{x} + \boldsymbol{g}'(\boldsymbol{0})^*(\boldsymbol{g}(\boldsymbol{0}) - \boldsymbol{0})\\
&= \boldsymbol{x} + \mathcal{N}_{\boldsymbol{g}}(\boldsymbol{0})\\
&\geq \boldsymbol{x} + \boldsymbol{\mu}\boldsymbol{g} - \frac{1}{2}\lambda\boldsymbol{d} && \text{(Lemma 1.33)}\\
&= \boldsymbol{\mu} - \frac{1}{2}\lambda\boldsymbol{d} && \square
\end{aligned}$$

By induction we can extend this lemma to the whole Newton sequence:

Lemma 1.35. *Let $\boldsymbol{d}$ be a cone vector of an SPP $\boldsymbol{f}$ and let $\lambda_{max} = \max_j\{\frac{\mu_j}{d_j}\}$. Then*

$$\boldsymbol{\nu}^{(k)} \geq \boldsymbol{\mu} - 2^{-k}\lambda_{max}\boldsymbol{d}\,.$$

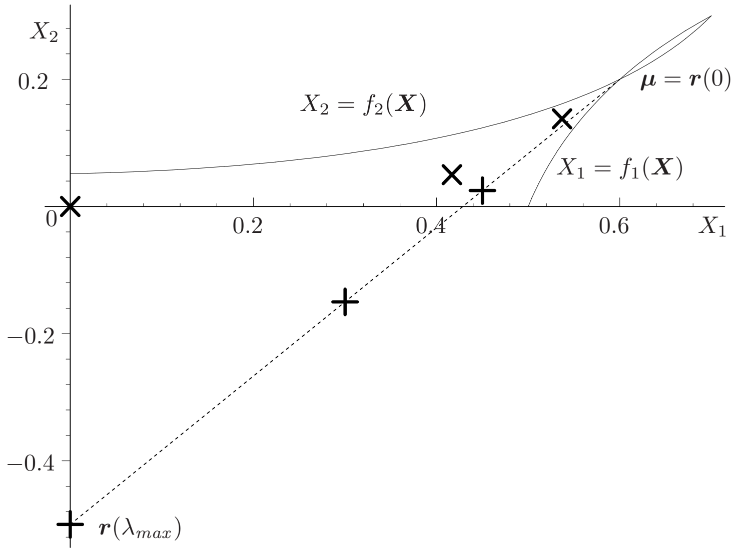

Figure 1.2: Illustration of Lemma 1.35: The points (shape: +) on the ray r along a cone vector are lower bounds on the Newton approximants (shape: ×).

Before proving the lemma we illustrate it by a picture. The dashed line in Figure 1.2 is the ray $\boldsymbol{r}(t) = \boldsymbol{\mu} - t\boldsymbol{d}$ along a cone vector $\boldsymbol{d}$. Notice that $\boldsymbol{r}(0)$ equals $\boldsymbol{\mu}$ and $\boldsymbol{r}(\lambda_{max})$ is the greatest point on the ray that is $\leq \boldsymbol{0}$. The figure also shows the Newton iterates $\boldsymbol{\nu}^{(k)}$ for $0 \leq k \leq 2$ (shape: ×) and the corresponding points $\boldsymbol{r}(2^{-k}\lambda_{max})$ (shape: +) located on the ray $\boldsymbol{r}$. Observe that $\boldsymbol{\nu}^{(k)} \geq \boldsymbol{r}(2^{-k}\lambda_{max})$, as claimed by Lemma 1.35.

Proof of Lemma 1.35. By induction on k. For the induction base ($k = 0$) we have for all components i:

$$(\boldsymbol{\mu} - \lambda_{max}\boldsymbol{d})_i = \left(\boldsymbol{\mu} - \max_j \left\{\frac{\mu_j}{d_j}\right\} \boldsymbol{d}\right)_i \leq \mu_i - \frac{\mu_i}{d_i} d_i = 0 \,,$$

so $\boldsymbol{\nu}^{(0)} = \boldsymbol{0} \geq \boldsymbol{\mu} - \lambda_{max}\boldsymbol{d}$.

For the induction step, let $k \geq 0$. By induction hypothesis we have $\boldsymbol{\nu}^{(k)} \geq \boldsymbol{\mu} - 2^{-k}\lambda_{max}\boldsymbol{d}$. So we can apply Lemma 1.34 to get

$$\boldsymbol{\nu}^{(k+1)} = \mathcal{N}(\boldsymbol{\nu}^{(k)}) \geq \boldsymbol{\mu} - \frac{1}{2} 2^{-k} \lambda_{max}\boldsymbol{d} = \boldsymbol{\mu} - 2^{-(k+1)}\lambda_{max}\boldsymbol{d} \,. \qquad \square$$

The following proposition guarantees a convergence order of the Newton sequence in terms of a cone vector.

Proposition 1.36. *Let $\boldsymbol{d}$ be a cone vector of an SPP $\boldsymbol{f}$ and let $\lambda_{max} = \max_j \left\{\frac{\mu_j}{d_j}\right\}$ and $\lambda_{min} = \min_j \left\{\frac{\mu_j}{d_j}\right\}$. Let $k_{\boldsymbol{f},\boldsymbol{d}} = \left\lceil \log \frac{\lambda_{max}}{\lambda_{min}} \right\rceil$. Then $\beta(k_{\boldsymbol{f},\boldsymbol{d}} + i) \geq i$ for all $i \in \mathbb{N}$.*

Proof. For all $1 \leq j \leq n$ the following holds.

$$\begin{aligned}
\left(\boldsymbol{\mu} - \boldsymbol{\nu}^{(k_{\boldsymbol{f},\boldsymbol{d}}+i)}\right)_j &\leq 2^{-(k_{\boldsymbol{f},\boldsymbol{d}}+i)} \lambda_{max} d_j && \text{(Lemma 1.35)} \\
&\leq \frac{\lambda_{min}}{\lambda_{max}} 2^{-i} \lambda_{max} d_j && \text{(def. of } k_{\boldsymbol{f},\boldsymbol{d}}) \\
&= \lambda_{min} d_j \cdot 2^{-i} \\
&\leq \mu_j \cdot 2^{-i} && \text{(def. of } \lambda_{min})
\end{aligned}$$

Hence, $\boldsymbol{\nu}^{(k_{\boldsymbol{f},\boldsymbol{d}}+i)}$ has i valid bits of $\boldsymbol{\mu}$. □

1.4.3 Convergence Speed Independent from Cone Vectors

The convergence order provided by Proposition 1.36 depends on a cone vector $\boldsymbol{d}$. While Proposition 1.32 guarantees the existence of a cone vector for scSPPs, it does not give any information on the magnitude of its components. So we do not have any bound yet on the "threshold" $k_{\boldsymbol{f},\boldsymbol{d}}$ from Proposition 1.36. The following theorem solves this problem.

Theorem 1.37. *Let $\boldsymbol{f}$ be a quadratic scSPP. Let c_{min} be the smallest nonzero coefficient of $\boldsymbol{f}$ and let μ_{min} and μ_{max} be the minimal and maximal component of $\boldsymbol{\mu}$, respectively. Let*

$$k_{\boldsymbol{f}} = \left\lceil \log \frac{\mu_{max}}{\mu_{min} \cdot (c_{min} \cdot \min\{\mu_{min}, 1\})^n} \right\rceil .$$

Then

$$\beta(k_{\boldsymbol{f}} + i) \geq i \textit{ for all } i \in \mathbb{N}.$$

Before we prove Theorem 1.37 we give an example.

Example 1.38. *As an example of application of Theorem 1.37 consider the scSPP equation of the back button process of Example 0.1.*

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 0.4X_2X_1 + 0.6 \\ 0.3X_1X_2 + 0.4X_3X_2 + 0.3 \\ 0.3X_1X_3 + 0.7 \end{pmatrix}$$

We wish to know if there is a component $s \in \{1,2,3\}$ with $\mu_s = 1$. Notice that $\boldsymbol{f}(\mathbf{1}) = \mathbf{1}$, so $\boldsymbol{\mu} \leq \mathbf{1}$. Performing 14 Newton steps (e.g. with Maple) yields an approximation $\boldsymbol{\nu}^{(14)}$ to $\boldsymbol{\mu}$ with

$$\begin{pmatrix} 0.98 \\ 0.97 \\ 0.992 \end{pmatrix} \leq \boldsymbol{\nu}^{(14)} \leq \begin{pmatrix} 0.99 \\ 0.98 \\ 0.993 \end{pmatrix} .$$

We have $c_{min} = 0.3$. In addition, since Newton's method converges to $\boldsymbol{\mu}$ from below, we know $\mu_{min} \geq 0.97$. Moreover, $\mu_{max} \leq 1$, as $\mathbf{1} = \boldsymbol{f}(\mathbf{1})$ and so $\boldsymbol{\mu} \leq \mathbf{1}$. Hence $k_{\boldsymbol{f}} \leq \left\lceil \log \frac{1}{0.97 \cdot (0.3 \cdot 0.97)^3} \right\rceil = 6$. Theorem 1.37 then implies that $\boldsymbol{\nu}^{(14)}$ has 8 valid bits of $\boldsymbol{\mu}$. As $\boldsymbol{\mu} \leq \mathbf{1}$, the absolute errors are bounded by the relative errors, and since $2^{-8} \leq 0.004$ we know:

$$\boldsymbol{\mu} \leq \boldsymbol{\nu}^{(14)} + \begin{pmatrix} 2^{-8} \\ 2^{-8} \\ 2^{-8} \end{pmatrix} \leq \begin{pmatrix} 0.994 \\ 0.984 \\ 0.997 \end{pmatrix} \prec \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

So Theorem 1.37 yields a proof that $\mu_s < 1$ for all three components s.

Notice also that the Newton sequence converges much faster than the Kleene sequence $(\boldsymbol{\kappa}^{(k)})_{k \in \mathbb{N}}$. We have $\boldsymbol{\kappa}^{(14)} \prec \left(0.89, 0.83, 0.96\right)^\top$, so $\boldsymbol{\kappa}^{(14)}$ has no more than 4 valid bits in any component, whereas $\boldsymbol{\nu}^{(14)}$ has, in fact, more than 30 valid bits in each component.

For the proof of Theorem 1.37 we need the following lemma.

Lemma 1.39. *Let $\boldsymbol{d}$ be a cone vector of a quadratic scSPP $\boldsymbol{f}$. Let c_{min} be the smallest nonzero coefficient of $\boldsymbol{f}$ and μ_{min} the minimal component of $\boldsymbol{\mu}$. Let d_{min} and d_{max} be the smallest and the largest component of $\boldsymbol{d}$, respectively. Then*

$$\frac{d_{min}}{d_{max}} \geq (c_{min} \cdot \min\{\mu_{min}, 1\})^n .$$

Proof. Let w.l.o.g. $d_1 = d_{max}$ and $d_n = d_{min}$. We claim the existence of indices s, t with $1 \leq s, t \leq n$ such that $\boldsymbol{f}'_{st}(\boldsymbol{\mu}) \neq 0$ and

$$\frac{d_{min}}{d_{max}} \geq \left(\frac{d_s}{d_t}\right)^n . \tag{1.7}$$

To prove that such s, t exist, we use the fact that $\boldsymbol{f}$ is strongly connected, i.e., that there is a sequence $1 = r_1, r_2, \ldots, r_q = n$ with $q \leq n$ such that $\boldsymbol{f}'_{r_{j+1}r_j}(\boldsymbol{X})$ is not constant zero. As $\boldsymbol{\mu} \succ \boldsymbol{0}$, we have $\boldsymbol{f}'_{r_{j+1}r_j}(\boldsymbol{\mu}) \neq 0$. Furthermore

$$\begin{aligned} \frac{d_1}{d_n} &= \frac{d_{r_1}}{d_{r_2}} \cdots \frac{d_{r_{q-1}}}{d_{r_q}} \text{ , and so} \\ \log \frac{d_1}{d_n} &= \log \frac{d_{r_1}}{d_{r_2}} + \cdots + \log \frac{d_{r_{q-1}}}{d_{r_q}} . \end{aligned}$$

So there must exist a j such that

$$\begin{aligned} \log \frac{d_1}{d_n} &\leq (q-1) \log \frac{d_{r_j}}{d_{r_{j+1}}} \leq n \log \frac{d_{r_j}}{d_{r_{j+1}}} \quad \text{ , and so} \\ \frac{d_n}{d_1} &\geq \left(\frac{d_{r_{j+1}}}{d_{r_j}}\right)^n . \end{aligned}$$

Hence one can choose $s = r_{j+1}$ and $t = r_j$.

As $\boldsymbol{d}$ is a cone vector we have $\boldsymbol{f}'(\boldsymbol{\mu})\boldsymbol{d} \leq \boldsymbol{d}$ and thus $\boldsymbol{f}'_{st}(\boldsymbol{\mu}) d_t \leq d_s$. Hence

$$\boldsymbol{f}'_{st}(\boldsymbol{\mu}) \leq \frac{d_s}{d_t} . \tag{1.8}$$

On the other hand, since $\boldsymbol{f}$ is quadratic, $\boldsymbol{f}'$ is a linear mapping such that

$$\boldsymbol{f}'_{st}(\boldsymbol{\mu}) = 2(b_1 \cdot \mu_1 + \cdots + b_n \cdot \mu_n) + \ell$$

where $b_1, \ldots, b_n$ and ℓ are coefficients of quadratic, respectively linear, monomials of $\boldsymbol{f}$. As $\boldsymbol{f}'_{st}(\boldsymbol{\mu}) \neq 0$, at least one of these coefficients must be nonzero and so greater than or equal to c_{min}. It follows $\boldsymbol{f}'_{st}(\boldsymbol{\mu}) \geq c_{min} \cdot \min\{\mu_{min}, 1\}$. So we have

$$\begin{aligned} (c_{min} \cdot \min\{\mu_{min}, 1\})^n &\leq \left(\boldsymbol{f}'_{st}(\boldsymbol{\mu})\right)^n \\ &\leq \left(\frac{d_s}{d_t}\right)^n && \text{(by (1.8))} \\ &\leq \frac{d_{min}}{d_{max}} && \text{(by (1.7))} . \end{aligned}$$ □

Now we can prove Theorem 1.37.

Proof of Theorem 1.37. By Proposition 1.32, $\boldsymbol{f}$ has a cone vector $\boldsymbol{d}$. Let $d_{max} = \max_j\{d_j\}$ and $d_{min} = \min_j\{d_j\}$ and $\lambda_{max} = \max_j \left\{\frac{\mu_j}{d_j}\right\}$ and $\lambda_{min} = \min_j \left\{\frac{\mu_j}{d_j}\right\}$. We have:

$$\begin{aligned} \frac{\lambda_{max}}{\lambda_{min}} &\leq \frac{\mu_{max} \cdot d_{max}}{\mu_{min} \cdot d_{min}} && \left(\text{as } \lambda_{max} \leq \frac{d_{max}}{\mu_{min}} \text{ and } \lambda_{min} \geq \frac{d_{min}}{\mu_{max}}\right) \\ &\leq \frac{\mu_{max}}{\mu_{min} \cdot (c_{min} \cdot \min\{\mu_{min}, 1\})^n} && \text{(Lemma 1.39)} . \end{aligned}$$

So the statement follows with Proposition 1.36. □

The following consequence of Theorem 1.37 removes some of the parameters on which the $k_{\boldsymbol{f}}$ from Theorem 1.37 depends.

Theorem 1.40. *Let $\boldsymbol{f}$ be a quadratic scSPP, let μ_{min} and μ_{max} be the minimal and maximal component of $\boldsymbol{\mu}$, respectively, and let the coefficients of $\boldsymbol{f}$ be given as ratios of m-bit integers. Then*

$$\beta(k_{\boldsymbol{f}} + i) \geq i \textit{ for all } i \in \mathbb{N}$$

holds for any of the following choices of $k_{\boldsymbol{f}}$.

(1) $\lceil 4mn + 3n \max\{0, -\log \mu_{min}\} \rceil$*;*

(2) $4mn2^n$*;*

(3) $7mn$ *whenever* $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$*;*

(4) $2mn + m$ *whenever both* $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$ *and* $\mu_{max} \leq 1$*.*

Items (3) and (4) of Theorem 1.40 apply in particular to termination SPPs of strict pPDAs (§ 1.1.4), i.e., they satisfy $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$ and $\mu_{max} \leq 1$.

To prove Theorem 1.40 we need some relations between the parameters of $\boldsymbol{f}$. We collect them in the following lemma.

Lemma 1.41. *Let $\boldsymbol{f}$ be a quadratic scSPP. With the terminology of Theorem 1.37 and Theorem 1.40 the following relations hold.*

(1) $c_{min} \geq 2^{-m}$*.*

(2) If $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$ *then* $\mu_{min} \geq c_{min}$*.*

(3) If $c_{min} > 1$ *then* $\mu_{min} > 1$*.*

(4) If $c_{min} \leq 1$ *then* $\mu_{min} \geq c_{min}^{2^n-1}$*.*

(5) If $\boldsymbol{f}$ is strictly quadratic, i.e. nonlinear, then the following inequalities hold: $c_{min} \leq 1$ *and* $\mu_{max} \cdot c_{min}^{3n-2} \cdot \min\{\mu_{min}^{2n-2}, 1\} \leq 1$*.*

Proof. We show the relations in turn.

(1) The smallest nonzero coefficient representable as a ratio of m-bit numbers is $\frac{1}{2^m}$.

(2) As $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$, in all components i there is a nonzero coefficient c_i such that $f_i(\mathbf{0}) = c_i$. We have $\boldsymbol{\mu} \geq \boldsymbol{f}(\mathbf{0})$, so $\mu_i \geq f_i(\mathbf{0}) = c_i \geq c_{min} > 0$ holds for all i. Hence $\mu_{min} > 0$.

(3) Let $c_{min} > 1$. Recall the Kleene sequence $(\boldsymbol{\kappa}^{(k)})_{k \in \mathbb{N}}$ with $\boldsymbol{\kappa}^{(k)} = \boldsymbol{f}^k(\mathbf{0})$. We first show by induction on k that for all $k \in \mathbb{N}$ and all components i either $\kappa_i^{(k)} = 0$ holds or $\kappa_i^{(k)} > 1$. For the induction base we have $\boldsymbol{\kappa}^{(0)} = \mathbf{0}$. Let $k \geq 0$. Then $\kappa_i^{(k+1)} = f_i(\boldsymbol{\kappa}^{(k)})$ is a sum of products of numbers which are either coefficients of $\boldsymbol{f}$ (and hence by assumption greater than 1) or $\kappa_j^{(k)}$ for some j. By induction, $\kappa_j^{(k)}$ is either 0 or greater than 1. So, $\kappa_i^{(k+1)}$ must be 0 or greater than 1.

By Theorem 1.3, the Kleene sequence converges to $\boldsymbol{\mu}$. As $\boldsymbol{f}$ is clean, we have $\boldsymbol{\mu} \succ \mathbf{0}$, and so there is a $k \in \mathbb{N}$ such that $\boldsymbol{\kappa}^{(k)} \succ \mathbf{1}$. The statement follows with $\boldsymbol{\mu} \geq \boldsymbol{\kappa}^{(k)}$.

(4) Let $c_{min} \leq 1$. We prove the following stronger claim by induction on k: For every k with $0 \leq k \leq n$ there is a set $S_k \subseteq \{1,\ldots,n\}$, $|S_k| = k$, such that $\mu_s \geq c_{min}^{2^k-1}$ holds for all $s \in S_k$. The induction base ($k = 0$) is trivial. Let $k \geq 0$. Consider the SPP $\widehat{\boldsymbol{f}}(\boldsymbol{X}_{\{1,\ldots,n\}\setminus S_k})$ that is obtained from $\boldsymbol{f}(\boldsymbol{X})$ by removing the S_k-components from $\boldsymbol{f}$ and replacing every S_k-variable in the polynomials by the corresponding component of $\boldsymbol{\mu}$. Clearly, $\boldsymbol{\mu}\widehat{\boldsymbol{f}} = \boldsymbol{\mu}_{\{1,\ldots,n\}\setminus S_k}$. By induction, the smallest nonzero coefficient $\widehat{c}_{min}$ of $\widehat{\boldsymbol{f}}$ satisfies $\widehat{c}_{min} \geq c_{min}(c_{min}^{2^k-1})^2 = c_{min}^{2^{k+1}-1}$. Pick a component i with $\widehat{f}_i(\boldsymbol{0}) > 0$. Then $\boldsymbol{\mu}\widehat{\boldsymbol{f}}_i \geq \widehat{f}_i(\boldsymbol{0}) \geq \widehat{c}_{min} \geq c_{min}^{2^{k+1}-1}$. So set $S_{k+1} := S_k \cup \{i\}$.

(5) Let w.l.o.g. $\mu_{max} = \mu_1$. The proof is based on the idea that X_1 indirectly depends quadratically on itself. More precisely, as $\boldsymbol{f}$ is strongly connected and strictly quadratic, component 1 depends (indirectly) on some component, say i_r, such that f_{i_r} contains a degree-2-monomial. The variables in that monomial, in turn, depend on X_1. This gives an inequality of the form $\mu_1 \geq C \cdot {\mu_1}^2$, implying $\mu_1 \cdot C \leq 1$.

We give the details in the following. As $\boldsymbol{f}$ is strongly connected and strictly quadratic there exists a sequence of variables $X_{i_1}, \ldots, X_{i_r}$ and a sequence of monomials $m_{i_1}, \ldots, m_{i_r}$ $(1 \leq r \leq n)$ with the following properties:

- $X_{i_1} = X_1$,
- m_{i_u} is a monomial appearing in f_{i_u} $(1 \leq u \leq r)$,
- $m_{i_u} = c_{i_u} \cdot X_{i_{u+1}}$ $(1 \leq u \leq r)$,
- $m_{i_r} = c_{i_r} \cdot X_{j_1} \cdot X_{k_1}$ for some variables X_{j_1}, X_{k_1}.

Notice that

$$\begin{aligned} \mu_{max} = \mu_1 &\geq c_{i_1} \cdot \ldots \cdot c_{i_r} \cdot \mu_{j_1} \cdot \mu_{k_1} \\ &\geq \min(c_{min}^n, 1) \cdot \mu_{j_1} \cdot \mu_{k_1} \,. \end{aligned} \tag{1.9}$$

Again using that $\boldsymbol{f}$ is strongly connected, there exists a sequence of variables $X_{j_1}, \ldots, X_{j_s}$ and a sequence of monomials $m_{j_1}, \ldots, m_{j_{s-1}}$ $(1 \leq s \leq n)$ with the following properties:

- $X_{j_s} = X_1$,
- m_{j_u} is a monomial appearing in f_{j_u} $(1 \leq u \leq s-1)$,
- $m_{j_u} = c_{j_u} \cdot X_{j_{u+1}}$ or $m_{j_u} = c_{j_u} \cdot X_{j_{u+1}} \cdot X_{j'_{u+1}}$ for some variable $X_{j'_{u+1}}$ $(1 \leq u \leq s-1)$.

Notice that

$$\begin{aligned} \mu_{j_1} &\geq c_{j_1} \cdot \ldots \cdot c_{j_{s-1}} \cdot \min(\mu_{min}^{s-1}, 1) \cdot \mu_1 \\ &\geq \min(c_{min}^{n-1}, 1) \cdot \min(\mu_{min}^{n-1}, 1) \cdot \mu_1 \,. \end{aligned} \tag{1.10}$$

Similarly, there exists a sequence of variables $X_{k_1}, \ldots, X_{k_t}$ $(1 \leq t \leq n)$ with $X_{k_t} = X_1$ showing

$$\mu_{k_1} \geq \min(c_{min}^{n-1}, 1) \cdot \min(\mu_{min}^{n-1}, 1) \cdot \mu_1 \,. \tag{1.11}$$

Combining (1.9) with (1.10) and (1.11) yields

$$\mu_{max} \geq \min(c_{min}^{3n-2}, 1) \cdot \min(\mu_{min}^{2n-2}, 1) \cdot \mu_{max}^2 \,,$$

or

$$\mu_{max} \cdot \min(c_{min}^{3n-2}, 1) \cdot \min(\mu_{min}^{2n-2}, 1) \leq 1 \,. \tag{1.12}$$

Now it suffices to show $c_{min} \leq 1$. Assume for a contradiction $c_{min} > 1$. Then, by part (3), $\mu_{min} > 1$. Plugging this into (1.12) yields $\mu_{max} \leq 1$. This implies $\mu_{max} < \mu_{min}$, contradicting the definition of μ_{max} and μ_{min}. □

Now we are ready to prove Theorem 1.40.

Proof of Theorem 1.40.

(1) First we check the case where $\boldsymbol{f}$ is linear, i.e., all polynomials f_i have degree at most 1. In this case, Newton's method reaches $\boldsymbol{\mu}$ after one iteration, so the statement holds. Consequently, we can assume in the following that $\boldsymbol{f}$ is strictly quadratic, meaning that $\boldsymbol{f}$ is quadratic and there is a polynomial in $\boldsymbol{f}$ of degree 2.

By Theorem 1.37 it suffices to show

$$\log \frac{\mu_{max}}{\mu_{min} \cdot c_{min}^n \cdot \min\{\mu_{min}^n, 1\}} \leq 4mn + 3n \max\{0, -\log \mu_{min}\} \,.$$

We have

$$\begin{aligned} &\log \frac{\mu_{max}}{\mu_{min} \cdot c_{min}^n \cdot \min\{\mu_{min}^n, 1\}} \\ \leq\ &\log \frac{1}{c_{min}^{4n-2} \cdot \min\{\mu_{min}^{3n-1}, 1\}} && \text{(Lemma 1.41.5)} \\ \leq\ &4n \cdot \log \frac{1}{c_{min}} - \log(\min\{\mu_{min}^{3n-1}, 1\}) && \text{(Lemma 1.41.5: } c_{min} \leq 1\text{)} \\ \leq\ &4mn - \log(\min\{\mu_{min}^{3n-1}, 1\}) && \text{(Lemma 1.41.1)} \,. \end{aligned}$$

If $\mu_{min} \geq 1$ we have $-\log(\min\{\mu_{min}^{3n-1}, 1\}) \leq 0$, so we are done in this case. If $\mu_{min} \leq 1$ we have $-\log(\min\{\mu_{min}^{3n-1}, 1\}) = -(3n-1)\log \mu_{min} \leq 3n \cdot (-\log \mu_{min})$.

(2) By part (1) of this theorem, it suffices to show that $4mn + 3n \max\{0, -\log \mu_{min}\} \leq 4mn2^n$. This inequality obviously holds if $\mu_{min} \geq 1$. So let $\mu_{min} \leq 1$. Then, by Lemma 1.41.3, $c_{min} \leq 1$. Hence, by Lemma 1.41 parts (4) and (1), $\mu_{min} \geq c_{min}^{2^n-1} \geq 2^{-m(2^n-1)}$. So we have an upper bound on $-\log \mu_{min}$ with $-\log \mu_{min} \leq m(2^n-1)$ and get:

$$\begin{aligned} 4mn + 3n \max\{0, -\log \mu_{min}\} &\leq 4mn + 3nm(2^n-1) \\ &\leq 4mn + 4nm(2^n-1) = 4mn2^n \end{aligned}$$

(3) Let $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$. By part (1) of this theorem it suffices to show that $4mn + 3n \max\{0, -\log \mu_{min}\} \leq 7mn$ holds. By Lemma 1.41 parts (2) and (1), we have $\mu_{min} \geq c_{min} \geq 2^{-m}$, so $-\log \mu_{min} \leq m$. Hence, $4mn + 3n \max\{0, -\log \mu_{min}\} \leq 4mn + 3nm = 7mn$.

(4) Let $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$ and $\mu_{max} \leq 1$. By Theorem 1.37 it suffices to show that $\log \dfrac{\mu_{max}}{\mu_{min} \cdot c_{min}^n \cdot \min\{\mu_{min}^n, 1\}} \leq 2mn + m$. We have:

$$\begin{aligned} &\log \frac{\mu_{max}}{\mu_{min} \cdot c_{min}^n \cdot \min\{\mu_{min}^n, 1\}} \\ &\leq -n \log c_{min} - (n+1) \log \mu_{min} && \text{(as } \mu_{min} \leq \mu_{max} \leq 1\text{)} \\ &\leq -(2n+1) \log c_{min} && \text{(Lemma 1.41.2)} \\ &\leq 2mn + m && \text{(Lemma 1.41.1)} \end{aligned}$$

□

1.4.4 Upper Bounds on the Least Fixed Point Via Newton Approximants

By Theorem 1.12 each Newton approximant $\boldsymbol{\nu}^{(k)}$ is a lower bound on $\boldsymbol{\mu}$. Theorem 1.37 and Theorem 1.40 give us upper bounds on the error $\boldsymbol{\Delta}^{(k)} := \boldsymbol{\mu} - \boldsymbol{\nu}^{(k)}$. Those bounds can directly transformed into upper bounds on $\boldsymbol{\mu}$, as $\boldsymbol{\mu} = \boldsymbol{\nu}^{(k)} + \boldsymbol{\Delta}^{(k)}$, cf. Example 1.38.

Theorem 1.37 and Theorem 1.40 allow to compute bounds on $\boldsymbol{\Delta}^{(k)}$ even before the Newton iteration has been started. However, knowing in advance how many iterations are needed to reach a certain precision may be more than actually needed. We may be interested in computing $\boldsymbol{\mu}$ up to some given error bound and stop the Newton iteration as soon as this error bound can be guaranteed. The following two theorems can be used to this end.

Theorem 1.42. *Let $\boldsymbol{f}$ be a quadratic scSPP. Let $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{\mu}$ and $\boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x})$ such that $\boldsymbol{f}'(\boldsymbol{x})^*$ exists. Let c_{min} be the smallest nonzero coefficient of $\boldsymbol{f}$ and μ_{min} the minimal component of $\boldsymbol{\mu}$. Then*

$$\frac{\|\mathcal{N}(\boldsymbol{x}) - \boldsymbol{x}\|_\infty}{\|\boldsymbol{\mu} - \mathcal{N}(\boldsymbol{x})\|_\infty} \geq (c_{min} \cdot \min\{\mu_{min}, 1\})^n \ .$$

We prove Theorem 1.42 at the end of the section. It can be applied to the Newton approximants:

Theorem 1.43. *Let $\boldsymbol{f}$ be a quadratic scSPP. Let c_{min} be the smallest nonzero coefficient of $\boldsymbol{f}$ and μ_{min} the minimal component of $\boldsymbol{\mu}$. For all Newton approximants $\boldsymbol{\nu}^{(k)}$ with $\boldsymbol{\nu}^{(k)} \succ \mathbf{0}$, let $\nu_{min}^{(k)}$ be the smallest coefficient of $\boldsymbol{\nu}^{(k)}$. Then*

$$\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu} \leq \boldsymbol{\nu}^{(k)} + \left[\frac{\left\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\right\|_\infty}{\left(c_{min} \cdot \min\{\nu_{min}^{(k)}, 1\}\right)^n} \right]$$

where $[s]$ denotes the vector $\boldsymbol{x}$ with $x_j = s$ for all $1 \leq j \leq n$.

Proof of Theorem 1.43. Theorem 1.42 applies, due to Theorem 1.12, to the Newton approximants with $\boldsymbol{x} = \boldsymbol{\nu}^{(k-1)}$. So we get

$$\begin{aligned} \left\|\boldsymbol{\mu} - \boldsymbol{\nu}^{(k)}\right\|_\infty &\leq \frac{\left\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\right\|_\infty}{(c_{min} \cdot \min\{\mu_{min}, 1\})^n} \\ &\leq \frac{\left\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\right\|_\infty}{\left(c_{min} \cdot \min\{\nu_{min}^{(k)}, 1\}\right)^n} \qquad \text{(as } \boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu}) \ . \end{aligned}$$

Hence the statement follows from $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu}$. □

Example 1.44. *Consider the equation system from Example 1.10:*

$$\begin{aligned} \langle res\, X\, res \rangle &= 0.7 \cdot (\langle res\, X\, res \rangle \cdot \langle res\, X\, res \rangle + \langle res\, X\, eff \rangle \cdot \langle eff\, X\, res \rangle) \\ &\quad + \ 0.2 + 0.1 \langle eff\, X\, res \rangle \\ \langle res\, X\, eff \rangle &= 0.7 \cdot (\langle res\, X\, res \rangle \cdot \langle res\, X\, eff \rangle + \langle res\, X\, eff \rangle \cdot \langle eff\, X\, eff \rangle) + 0.1 \cdot \langle eff\, X\, eff \rangle \\ \langle eff\, X\, eff \rangle &= 0.3 \cdot (\langle eff\, X\, eff \rangle \cdot \langle eff\, X\, eff \rangle + \langle eff\, X\, res \rangle \cdot \langle res\, X\, eff \rangle) \\ &\quad + \ 0.6 + 0.1 \langle res\, X\, eff \rangle \\ \langle eff\, X\, res \rangle &= 0.3 \cdot (\langle eff\, X\, eff \rangle \cdot \langle eff\, X\, res \rangle + \langle eff\, X\, res \rangle \cdot \langle res\, X\, res \rangle) + 0.1 \cdot \langle res\, X\, res \rangle \end{aligned}$$

It is strongly connected, because $\langle res\,X\,res\rangle$ *depends on* $\langle eff\,X\,res\rangle$*, which depends on* $\langle eff\,X\,eff\rangle$*, which depends on* $\langle res\,X\,eff\rangle$*, which depends on* $\langle res\,X\,res\rangle$*. Performing 18 Newton iterations yields*

$$\boldsymbol{\nu}^{(18)} \approx \begin{pmatrix} 0.268 \\ 0.478 \\ 0.892 \\ 0.041 \end{pmatrix}$$

and $\left\|\boldsymbol{\nu}^{(18)} - \boldsymbol{\nu}^{(17)}\right\| \leq 10^{-17}$*. Hence, Theorem 1.43 implies that we have computed the termination probabilities within an error of* $\dfrac{10^{-17}}{(0.1 \cdot 0.04)^4} \leq 10^{-7}$*. Interpreting the termination probabilities, the risk of a pandemic is about* $1 - 0.268 - 0.478 \approx 0.26$*.*

Example 1.45. *Consider again the equation* $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ *from Examples 0.1 and 1.38:*

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 0.4X_2X_1 + 0.6 \\ 0.3X_1X_2 + 0.4X_3X_2 + 0.3 \\ 0.3X_1X_3 + 0.7 \end{pmatrix}$$

Again we wish to verify that there is no component $s \in \{1,2,3\}$ *with* $\mu_s = 1$*. Performing* 10 *Newton steps yields an approximation* $\boldsymbol{\nu}^{(10)}$ *to* $\boldsymbol{\mu}$ *with*

$$\begin{pmatrix} 0.9828 \\ 0.9738 \\ 0.9926 \end{pmatrix} \prec \boldsymbol{\nu}^{(10)} \prec \begin{pmatrix} 0.9829 \\ 0.9739 \\ 0.9927 \end{pmatrix} .$$

Further, it holds $\left\|\boldsymbol{\nu}^{(10)} - \boldsymbol{\nu}^{(9)}\right\|_\infty \leq 2 \cdot 10^{-6}$*. So we have*

$$\frac{\left\|\boldsymbol{\nu}^{(10)} - \boldsymbol{\nu}^{(9)}\right\|_\infty}{\left(c_{min} \cdot \min\{\nu^{(10)}_{min}, 1\}\right)^3} \leq \frac{2 \cdot 10^{-6}}{(0.3 \cdot 0.97)^3} \leq 0.00009$$

and hence by Theorem 1.43

$$\boldsymbol{\nu}^{(10)} \leq \boldsymbol{\mu} \leq \boldsymbol{\nu}^{(10)} + [0.00009] \leq \begin{pmatrix} 0.983 \\ 0.974 \\ 0.993 \end{pmatrix} .$$

In particular we know that $\mu_s < 1$ *for all three components* s*.*

Example 1.46. *Consider again the SPP* $\boldsymbol{f}$ *from Example 1.45. Setting*

$$\boldsymbol{u}^{(k)} := \boldsymbol{\nu}^{(k)} + \left[\frac{\left\|\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}\right\|_\infty}{\left(0.3 \cdot \nu^{(k)}_{min}\right)^3} \right] ,$$

Theorem 1.43 guarantees

$$\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu} \leq \boldsymbol{u}^{(k)} .$$

Let us measure the tightness of the bounds $\boldsymbol{\nu}^{(k)}$ *and* $\boldsymbol{u}^{(k)}$ *on* $\boldsymbol{\mu}$ *in the first component. Let*

$$\begin{aligned} p_{lower}(k) &:= -\log_2(\mu_1 - \nu_1^{(k)}) \qquad \text{and} \\ p_{upper}(k) &:= -\log_2(u_1^{(k)} - \mu_1) . \end{aligned}$$

Roughly speaking, $\nu_1^{(k)}$ *and* $u_1^{(k)}$ *have* $p_{lower}(k)$ *and* $p_{upper}(k)$ *valid bits of* μ_1*, respectively. Figure 1.3 shows* $p_{lower}(k)$ *and* $p_{upper}(k)$ *for* $k \in \{1, \ldots, 11\}$*.*

It can be seen that the slope of $p_{lower}(k)$ is approximately 1 for $k = 2, \dots, 6$. This corresponds to the linear convergence of Newton's method according to Theorem 1.37. Since $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is non-singular[2], Newton's method actually has, asymptotically, an exponential convergence order, cf. Theorem 1.24. This behavior can be observed in Figure 1.3 for $k \geq 7$. For p_{upper}, we roughly have (using $\boldsymbol{\nu}^{(k)} \approx \boldsymbol{\mu}$):

$$p_{upper}(k) \approx p_{lower}(k-1) + \log\left(0.3 \cdot \nu_{min}^{(k)}\right)^3 \approx p_{lower}(k-1) - 5\ .$$

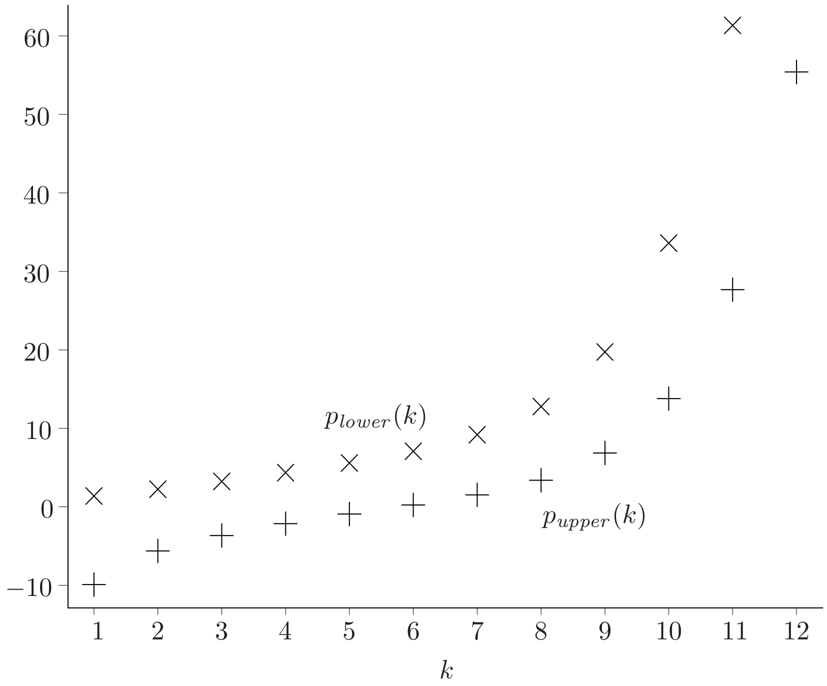

Figure 1.3: Number of valid bits of the lower (shape: ×) and upper (shape: +) bounds on μ_1, see Example 1.46.

The proof of Theorem 1.42 uses similar techniques as the proof of Theorem 1.37, in particular Lemma 1.39.

Proof of Theorem 1.42. By Proposition 1.32, $\boldsymbol{f}$ has a cone vector $\boldsymbol{d}$. Let d_{min} and d_{max} be the smallest and the largest component of $\boldsymbol{d}$, respectively. Let $\lambda_{max} := \max_j\{\frac{\mu_j - x_j}{d_j}\}$, and let w.l.o.g. $\lambda_{max} = \frac{\mu_1 - x_1}{d_1}$. We have $\boldsymbol{x} \geq \boldsymbol{\mu} - \lambda_{max}\boldsymbol{d}$, so we can apply Lemma 1.34 to obtain $\mathcal{N}(\boldsymbol{x}) \geq \boldsymbol{\mu} - \frac{1}{2}\lambda_{max}\boldsymbol{d}$. Thus

$$\|\mathcal{N}(\boldsymbol{x}) - \boldsymbol{x}\|_\infty \geq (\mathcal{N}(\boldsymbol{x}) - \boldsymbol{x})_1 \geq \mu_1 - \frac{1}{2}\lambda_{max}d_1 - x_1 = \frac{1}{2}\lambda_{max}d_1 \geq \frac{1}{2}\lambda_{max}d_{min}\ .$$

On the other hand, with Lemma 1.16 we have $\boldsymbol{0} \leq \boldsymbol{\mu} - \mathcal{N}(\boldsymbol{x}) \leq \frac{1}{2}\lambda_{max}\boldsymbol{d}$ and so $\|\boldsymbol{\mu} - \mathcal{N}(\boldsymbol{x})\|_\infty \leq \frac{1}{2}\lambda_{max}d_{max}$. Combining those inequalities we obtain

$$\frac{\|\mathcal{N}(\boldsymbol{x}) - \boldsymbol{x}\|_\infty}{\|\boldsymbol{\mu} - \mathcal{N}(\boldsymbol{x})\|_\infty} \geq \frac{d_{min}}{d_{max}}\ .$$

Now the statement follows from Lemma 1.39. □

[2] In fact, the matrix $I - \boldsymbol{f}'(\boldsymbol{\mu})$ is "almost" singular, with a determinant of about 0.006.

1.5 General SPPs

In § 1.4 we considered *strongly connected* SPPs, see Definition 1.7. However, it is not always guaranteed that the SPP $\boldsymbol{f}$ is strongly connected. In this section we analyze the convergence speed of two variants of Newton's method that both compute approximations of $\boldsymbol{\mu}$, where $\boldsymbol{f}$ is an SPP that is not necessarily strongly connected ("general SPPs").

The first one was suggested by Etessami and Yannakakis [EY09] and is called *Decomposed Newton Method (DNM)*. It works by running Newton's method separately on each SCC, see § 1.5.1. The second one is the regular Newton's method from § 1.3. We will analyze its convergence speed in § 1.5.2.

The reason why we first analyze DNM is that our convergence speed results about Newton's method for general SPPs (Theorem 1.51) build on our results about DNM (Theorem 1.48). Moreover, from an efficiency point of view it actually may be advantageous to run Newton's method separately on each SCC. For those reasons DNM deserves a separate treatment.

1.5.1 Convergence Speed of the Decomposed Newton Method (DNM)

DNM, originally suggested in [EY09], works as follows. It starts by using Newton's method for each bottom SCC, say S, of the SPP $\boldsymbol{f}$. Then the corresponding variables $\boldsymbol{X}_S$ are substituted for the obtained approximation for $\boldsymbol{\mu}_S$, and the corresponding equations $\boldsymbol{X}_S = \boldsymbol{f}_S(\boldsymbol{X})$ are removed. The same procedure is then applied to the new bottom SCCs, until all SCCs have been processed.

Etessami and Yannakakis did not provide a particular criterion for the number of Newton iterations to be applied in each SCC. Consequently, they did not analyze the convergence speed of DNM. We will treat those issues in this section, thereby taking advantage of our previous analysis of scSPPs.

We fix a quadratic SPP $\boldsymbol{f}$ for this section. We assume that we have already computed the DAG (directed acyclic graph) of SCCs. This can be done in linear time in the size of $\boldsymbol{f}$. To each SCC S we can associate its *depth* t: it is the longest path in the DAG of SCCs from S to a top SCC. Notice that $0 \leq t \leq n-1$. We write $\mathcal{SCC}(t)$ for the set of SCCs of depth t. We define the height $h(\boldsymbol{f})$ as the largest depth of an SCC and the width $w(\boldsymbol{f}) := \max_t |\mathcal{SCC}(t)|$ as the largest number of SCCs of the same depth. Notice that $\boldsymbol{f}$ has at most $(h(\boldsymbol{f})+1) \cdot w(\boldsymbol{f})$ SCCs. Further we define the component sets $[t] := \bigcup_{S \in \mathcal{SCC}(t)} S$ and $[{>}t] := \bigcup_{t' > t} [t']$ and similarly $[{<}t]$.

Algorithm 1.1 shows our version of DNM. We suggest to run Newton's method in each SCC S for a number of steps that depends (exponentially) on the depth of S and (linearly) on a parameter i that controls the precision.

The number of Newton iterations in one call of DNM can be bounded as follows.

Proposition 1.47. *The procedure* DNM$(\boldsymbol{f}, i)$ *of Algorithm 1.1 runs at most* $i \cdot w(\boldsymbol{f}) \cdot 2^{h(\boldsymbol{f})+1} \leq i \cdot n \cdot 2^n$ *iterations of Newton's method.*

Algorithm 1.1 Decomposed Newton Method (DNM)

```
procedure DNM $(\boldsymbol{f}, i)$                /* The parameter $i$ controls the precision. */
assumes: $\boldsymbol{f}$ is an SPP
returns: an approximation $\boldsymbol{\rho}^{(i)}$ of $\boldsymbol{\mu f}$
    for $t$ from $h(\boldsymbol{f})$ downto $0$
        forall $S \in \mathcal{SCC}(t)$                          /* for all SCCs $S$ of depth $t$ */
            $\boldsymbol{\rho}_S^{(i)} \leftarrow \mathcal{N}_{\boldsymbol{f}_S}^{i \cdot 2^t}(\mathbf{0})$          /* perform $i \cdot 2^t$ Newton iterations */
            $\boldsymbol{f}_{[<t]} \leftarrow \boldsymbol{f}_{[<t]}[S/\boldsymbol{\rho}_S^{(i)}]$      /* apply $\boldsymbol{\rho}_S^{(i)}$ in the upper SCCs */
    return $\boldsymbol{\rho}^{(i)}$
```

Proof. The number of iterations is $\sum_{t=0}^{h(\boldsymbol{f})} |\mathcal{SCC}(t)| \cdot i \cdot 2^t$. This can be estimated as follows.

$$\begin{aligned}
\sum_{t=0}^{h(\boldsymbol{f})} |\mathcal{SCC}(t)| \cdot i \cdot 2^t &\leq w(\boldsymbol{f}) \cdot i \cdot \sum_{t=0}^{h(\boldsymbol{f})} 2^t \\
&\leq w(\boldsymbol{f}) \cdot i \cdot 2^{h(\boldsymbol{f})+1} \\
&\leq i \cdot n \cdot 2^n \qquad\qquad (\text{as } w(\boldsymbol{f}) \leq n \text{ and } h(\boldsymbol{f}) < n) \qquad \square
\end{aligned}$$

The following theorem states that DNM has linear convergence order.

Theorem 1.48. *Let $\boldsymbol{f}$ be a quadratic SPP. Let $\boldsymbol{\rho}^{(i)}$ denote the result of calling* DNM$(\boldsymbol{f}, i)$ *(see Algorithm 1.1). Let $\beta_{\boldsymbol{\rho}}$ denote the convergence order of $(\boldsymbol{\rho}^{(i)})_{i \in \mathbb{N}}$. Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\beta_{\boldsymbol{\rho}}(k_{\boldsymbol{f}} + i) \geq i$ for all $i \in \mathbb{N}$.*

Theorem 1.48 can be interpreted as follows: Increasing i by one yields asymptotically at least one additional bit in each component and, by Proposition 1.47, costs at most $n \cdot 2^n$ additional Newton iterations. Notice that for simplicity we do not take into account here that the cost of performing a Newton step on a single SCC is not uniform, but rather depends on the size of the SCC (e.g. cubically if Gaussian elimination is used for solving the linear systems).

For the proof of Theorem 1.48, let $\boldsymbol{\Delta}^{(i)}$ denote the error when running DNM with parameter i, i.e., $\boldsymbol{\Delta}^{(i)} := \boldsymbol{\mu} - \boldsymbol{\rho}^{(i)}$. Observe that the error $\boldsymbol{\Delta}^{(i)}$ can be understood as the sum of two errors:

$$\boldsymbol{\Delta}^{(i)} := \boldsymbol{\mu} - \boldsymbol{\rho}^{(i)} = (\boldsymbol{\mu} - \widetilde{\boldsymbol{\mu}}^{(i)}) + (\widetilde{\boldsymbol{\mu}}^{(i)} - \boldsymbol{\rho}^{(i)}) \,,$$

where $\widetilde{\boldsymbol{\mu}}_{[t]}^{(i)} := \boldsymbol{\mu}\big(\boldsymbol{f}_{[t]}[[>t]/\boldsymbol{\rho}_{[>t]}^{(i)}]\big)$, i.e., $\widetilde{\boldsymbol{\mu}}_{[t]}^{(i)}$ is the least fixed point of $\boldsymbol{f}_{[t]}$ after the approximations from the lower SCCs have been applied. So, $\boldsymbol{\Delta}_{[t]}^{(i)}$ consists of the *propagation error* $(\boldsymbol{\mu}_{[t]} - \widetilde{\boldsymbol{\mu}}_{[t]}^{(i)})$ (resulting from the error at lower SCCs) and the *approximation error* $(\widetilde{\boldsymbol{\mu}}_{[t]}^{(i)} - \boldsymbol{\rho}_{[t]}^{(i)})$ (resulting from the newly added error of Newton's method on level t).

The following lemma gives a bound on the propagation error.

Lemma 1.49 (Propagation error)**.** *There is a constant $C_{\boldsymbol{f}} > 0$ such that*

$$\left\| \boldsymbol{\mu}_{[t]} - \widetilde{\boldsymbol{\mu}}_{[t]} \right\| \leq C_{\boldsymbol{f}} \cdot \sqrt{\left\| \boldsymbol{\mu}_{[>t]} - \boldsymbol{\rho}_{[>t]} \right\|}$$

holds for all $\boldsymbol{\rho}_{[>t]}$ with $\mathbf{0} \leq \boldsymbol{\rho}_{[>t]} \leq \boldsymbol{\mu}_{[>t]}$, where $\widetilde{\boldsymbol{\mu}}_{[t]} = \boldsymbol{\mu}\big(\boldsymbol{f}_{[t]}[[>t]/\boldsymbol{\rho}_{[>t]}]\big)$.

Roughly speaking, Lemma 1.49 states that if $\boldsymbol{\rho}^{(i)}_{[>t]}$ has k valid bits of $\boldsymbol{\mu}_{[>t]}$, then $\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}$ has at least about $k/2$ valid bits of $\boldsymbol{\mu}_{[t]}$. In other words, (at most) one half of the valid bits are lost on each level of the DAG due to the propagation error. The proof of Lemma 1.49 is technically involved and, unfortunately, not constructive in that we know nothing about $C_{\boldsymbol{f}}$ except for its existence. The proof can be found in Appendix A.1.

The following lemma gives a bound on the error $\left\|\boldsymbol{\Delta}^{(i)}_{[t]}\right\|$ on level t, taking both the propagation error and the approximation error into account.

Lemma 1.50. *There is a $C_{\boldsymbol{f}} > 0$ such that $\left\|\boldsymbol{\Delta}^{(i)}_{[t]}\right\| \leq 2^{C_{\boldsymbol{f}} - i \cdot 2^t}$ for all $i \in \mathbb{N}$.*

Proof. Let $\widetilde{\boldsymbol{f}}^{(i)}_{[t]} := \boldsymbol{f}_{[t]}[[>t]/\boldsymbol{\rho}^{(i)}_{[>t]}]$. Observe that the coefficients of $\widetilde{\boldsymbol{f}}^{(i)}_{[t]}$ and thus its least fixed point $\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}$ are monotonically increasing with i, because $\boldsymbol{\rho}^{(i)}_{[>t]}$ is monotonically increasing as well. Consider an arbitrary depth t and choose real numbers $c_{min} > 0$ and $\mu_{min} > 0$ and an integer i_0 such that, for all $i \geq i_0$, c_{min} and μ_{min} are lower bounds on the smallest nonzero coefficient of $\widetilde{\boldsymbol{f}}^{(i)}_{[t]}$ and the smallest coefficient of $\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}$, respectively. Let μ_{max} be the largest component of $\boldsymbol{\mu}_{[t]}$. Let $\widetilde{k} := \left\lceil n \cdot \log \frac{\mu_{max}}{c_{min} \cdot \mu_{min} \cdot \min\{\mu_{min}, 1\}} \right\rceil$. Then it follows from Theorem 1.37 that performing $\widetilde{k} + j$ Newton iterations ($j \geq 0$) on depth t yields j valid bits of $\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}$ for any $i \geq i_0$. In particular, $\widetilde{k} + i \cdot 2^t$ Newton iterations give $i \cdot 2^t$ valid bits of $\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}$ for any $i \geq i_0$. So there exists a constant $c_1 > 0$ such that, for all $i \geq i_0$,

$$\left\|\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]} - \boldsymbol{\rho}^{(i)}_{[t]}\right\| \leq 2^{c_1 - i \cdot 2^t}\,, \tag{1.13}$$

because DNM (see Algorithm 1.1) performs $i \cdot 2^t$ iterations to compute $\boldsymbol{\rho}^{(i)}_S$ where S is an SCC of depth t. Choose c_1 large enough such that Equation (1.13) holds for all $i \geq 0$ and all depths t.

Now we can prove the theorem by induction on t. In the base case ($t = h(\boldsymbol{f})$) there is no propagation error, so the claim of the lemma follows from (1.13). Let $t < h(\boldsymbol{f})$. Then

$$\begin{aligned}
\left\|\boldsymbol{\Delta}^{(i)}_{[t]}\right\| &= \left\|\boldsymbol{\mu}_{[t]} - \widetilde{\boldsymbol{\mu}}^{(i)}_{[t]} + \widetilde{\boldsymbol{\mu}}^{(i)}_{[t]} - \boldsymbol{\rho}^{(i)}_{[t]}\right\| \\
&\leq \left\|\boldsymbol{\mu}_{[t]} - \widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}\right\| + \left\|\widetilde{\boldsymbol{\mu}}^{(i)}_{[t]} - \boldsymbol{\rho}^{(i)}_{[t]}\right\| \\
&\leq \left\|\boldsymbol{\mu}_{[t]} - \widetilde{\boldsymbol{\mu}}^{(i)}_{[t]}\right\| + 2^{c_1 - i \cdot 2^t} && \text{(by (1.13))} \\
&\leq c_2 \cdot \sqrt{\left\|\boldsymbol{\Delta}^{(i)}_{[>t]}\right\|} + 2^{c_1 - i \cdot 2^t} && \text{(Lemma 1.49)} \\
&\leq c_2 \cdot \sqrt{2^{c_3 - i \cdot 2^{t+1}}} + 2^{c_1 - i \cdot 2^t} && \text{(induction hypothesis)} \\
&\leq 2^{c_4 - i \cdot 2^t}
\end{aligned}$$

for some constants $c_2, c_3, c_4 > 0$. □

Now Theorem 1.48 follows easily.

Proof of Theorem 1.48. From Lemma 1.50 we deduce that for each component $j \in [t]$ there is a c_j such that

$$(\mu_j - \rho^{(i)}_j)/\mu_j \leq 2^{c_j - i \cdot 2^t} \leq 2^{c_j - i}\,.$$

Let $k_{\boldsymbol{f}} \geq c_j$ for all $1 \leq j \leq n$. Then

$$(\mu_j - \rho^{(i+k_{\boldsymbol{f}})}_j)/\mu_j \leq 2^{c_j - (i + k_{\boldsymbol{f}})} \leq 2^{-i}\,.$$

□

It is an open problem to give bounds for $k_{\boldsymbol{f}}$ in Theorem 1.48. The results of this thesis do not immediately lead to such a bound because the proof of existence of the $C_{\boldsymbol{f}}$ in Lemma 1.49 is not constructive.

1.5.2 Convergence Speed of Newton's Method

We use the Theorem 1.48 to prove the following theorem for the regular (i.e. not decomposed) Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$.

Theorem 1.51. *Let $\boldsymbol{f}$ be a quadratic SPP. There is a threshold $k_{\boldsymbol{f}} \in \mathbb{N}$ such that*

$$\beta(k_{\boldsymbol{f}} + i \cdot n \cdot 2^n) \geq \beta(k_{\boldsymbol{f}} + i \cdot (h(\boldsymbol{f}) + 1) \cdot 2^{h(\boldsymbol{f})}) \geq i \textit{ for all } i \in \mathbb{N}.$$

In the rest of the section we prove this theorem by a sequence of lemmata. The following lemma states that a Newton step is not faster on an SCC, if the values of the lower SCCs are fixed.

Lemma 1.52. *Let $\boldsymbol{f}$ be an SPP. Let $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{\mu}$ such that $\boldsymbol{f}'(\boldsymbol{x})^*$ exists. Let S be an SCC of $\boldsymbol{f}$ and let L denote the set of components that are not in S, but on which a variable in S depends. Then $(\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}))_S \geq \mathcal{N}_{\boldsymbol{f}_S[L/\boldsymbol{x}_L]}(\boldsymbol{x}_S)$.*

Proof.

$$\begin{aligned}
(\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}))_S &= \big(\boldsymbol{f}'(\boldsymbol{x})^*(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})\big)_S \\
&= \boldsymbol{f}'(\boldsymbol{x})^*_{SS}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})_S + \boldsymbol{f}'(\boldsymbol{x})^*_{SL}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})_L \\
&\geq \boldsymbol{f}'(\boldsymbol{x})^*_{SS}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x})_S \\
&= \big((\boldsymbol{f}_S[L/\boldsymbol{x}_L])'(\boldsymbol{x}_S)\big)^*(\boldsymbol{f}_S[L/\boldsymbol{x}_L](\boldsymbol{x}_S) - \boldsymbol{x}_S) \\
&= \mathcal{N}_{\boldsymbol{f}_S[L/\boldsymbol{x}_L]}(\boldsymbol{x}_S)
\end{aligned}$$

□

Recall Lemma 1.20 which states that the Newton operator $\mathcal{N}$ is monotone. This fact and Lemma 1.52 can be combined to the following lemma stating that $i \cdot (h(\boldsymbol{f}) + 1)$ iterations of the regular Newton's method "dominate" a decomposed Newton method that performs i Newton steps in each SCC.

Lemma 1.53. *Let $\widetilde{\boldsymbol{\nu}}^{(i)}$ denote the result of a decomposed Newton method which performs i iterations of Newton's method in each SCC. Let $\boldsymbol{\nu}^{(i)}$ denote the result of i iterations of the regular Newton's method. Then $\boldsymbol{\nu}^{(i\cdot(h(\boldsymbol{f})+1))} \geq \widetilde{\boldsymbol{\nu}}^{(i)}$.*

Proof. Let $h = h(\boldsymbol{f})$. Let $[t]$ and $[{>}t]$ again denote the set of components of depth t and $> t$, respectively. We show by induction on the depth t:

$$\boldsymbol{\nu}^{(i\cdot(h+1-t))}_{[t]} \geq \widetilde{\boldsymbol{\nu}}^{(i)}_{[t]}$$

The induction base ($t = h$) is clear, because for bottom SCCs the two methods are identical. Let now $t < h$. Then we have:

$$\begin{aligned}
\boldsymbol{\nu}^{(i\cdot(h+1-t))}_{[t]} &= \mathcal{N}^i_{\boldsymbol{f}}(\boldsymbol{\nu}^{(i\cdot(h-t))})_{[t]} \\
&\geq \mathcal{N}^i_{\boldsymbol{f}_{[t]}[[>t]/\boldsymbol{\nu}^{(i\cdot(h-t))}_{[>t]}]}(\boldsymbol{\nu}^{(i\cdot(h-t))}_{[t]}) && \text{(Lemma 1.52)} \\
&\geq \mathcal{N}^i_{\boldsymbol{f}_{[t]}[[>t]/\widetilde{\boldsymbol{\nu}}^{(i)}_{[>t]}]}(\boldsymbol{\nu}^{(i\cdot(h-t))}_{[t]}) && \text{(induction hypothesis)} \\
&\geq \mathcal{N}^i_{\boldsymbol{f}_{[t]}[[>t]/\widetilde{\boldsymbol{\nu}}^{(i)}_{[>t]}]}(\mathbf{0}_{[t]}) && \text{(Lemma 1.20)} \\
&= \widetilde{\boldsymbol{\nu}}^{(i)}_{[t]} && \text{(definition of } \widetilde{\boldsymbol{\nu}}^{(i)}\text{)}
\end{aligned}$$

Now, the lemma itself follows by using Lemma 1.20 once more. □

As a side note, observe that above proof of Lemma 1.53 implicitly benefits from the fact that SCCs of the same depth are independent. So, SCCs with the same depth are handled in parallel by the regular Newton's method. Therefore, $w(\boldsymbol{f})$, the width of $\boldsymbol{f}$, is irrelevant here (cf. Proposition 1.47).

Now we can prove Theorem 1.51.

Proof of Theorem 1.51. Let k_2 be the $k_{\boldsymbol{f}}$ of Theorem 1.48, and let $k_1 = k_2 \cdot (h(\boldsymbol{f}) + 1) \cdot 2^{h(\boldsymbol{f})}$. Then we have

$$\begin{aligned} \boldsymbol{\nu}^{(k_1 + i \cdot (h(\boldsymbol{f})+1) \cdot 2^{h(f)})} &= \boldsymbol{\nu}^{((k_2+i) \cdot (h(\boldsymbol{f})+1) \cdot 2^{h(\boldsymbol{f})})} \\ &\geq \widetilde{\boldsymbol{\nu}}^{((k_2+i) \cdot 2^{h(\boldsymbol{f})})} && \text{(Lemma 1.53)} \\ &\geq \boldsymbol{\rho}^{(k_2+i)} , \end{aligned}$$

where the last step follows from the fact that $\mathrm{DNM}(\boldsymbol{f}, k_2 + i)$ runs at most $(k_2 + i) \cdot 2^{h(\boldsymbol{f})}$ iterations in every SCC. By Theorem 1.48, $\boldsymbol{\rho}^{(k_2+i)}$ and hence $\boldsymbol{\nu}^{(k_1 + i \cdot (h(\boldsymbol{f})+1) \cdot 2^{h(f)})}$ have i valid bits of $\boldsymbol{\mu}$. Therefore, Theorem 1.51 holds with $k_{\boldsymbol{f}} = k_1$. □

1.6 Upper Bounds on the Convergence

In this section we show that the lower bounds on the convergence order of Newton's method that we obtained in the previous section are essentially tight, meaning that an exponential (in n) number of iterations may be needed per bit.

More precisely, we expose a family $\left(\boldsymbol{f}^{(n)}\right)_{n \geq 1}$ of SPPs with n variables, such that more than $k \cdot 2^{n-1}$ iterations are needed for k valid bits. Consider the following system.

$$\boldsymbol{X} = \boldsymbol{f}^{(n)}(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{2} + \frac{1}{2} X_1^2 \\ \frac{1}{4} X_1^2 + \frac{1}{2} X_1 X_2 + \frac{1}{4} X_2^2 \\ \vdots \\ \frac{1}{4} X_{n-1}^2 + \frac{1}{2} X_{n-1} X_n + \frac{1}{4} X_n^2 \end{pmatrix} \tag{1.14}$$

The only solution of (1.14) is $\boldsymbol{\mu f}^{(n)} = (1, \ldots, 1)^\top$. Notice that each component of $\boldsymbol{f}^{(n)}$ is an SCC. We prove the following theorem.

Theorem 1.54. *The convergence order of Newton's method applied to the SPP* $\boldsymbol{f}^{(n)}$ *from* (1.14) *(with* $n \geq 2$*) satisfies*

$$\beta(k \cdot 2^{n-1}) < k \textit{ for all } k \in \{1, 2, \ldots\}.$$

In particular, $\beta(2^{n-1}) = 0$.

Proof. We write $\boldsymbol{f} := \boldsymbol{f}^{(n)}$ for simplicity. Let

$$\boldsymbol{\Delta}^{(i)} := \boldsymbol{\mu} - \boldsymbol{\nu}^{(i)} = (1, \ldots, 1)^\top - \boldsymbol{\nu}^{(i)} .$$

Notice that $(\nu_1^{(i)})_{i \in \mathbb{N}} = (0, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots)$ which is the same sequence as obtained by applying Newton's method to the 1-dimensional system $X_1 = \frac{1}{2} + \frac{1}{2} X_1^2$. So we have $\Delta_1^{(i)} = 2^{-i}$, i.e., after i iterations we have exactly i valid bits in the first component.

We know from Theorem 1.12 that for all j with $1 \le j \le n-1$ we have $\nu_{j+1}^{(i)} \le f_{j+1}(\boldsymbol{\nu}^{(i)}) = \frac{1}{4}(\nu_j^{(i)})^2 + \frac{1}{2}\nu_j^{(i)}\nu_{j+1}^{(i)} + \frac{1}{4}(\nu_{j+1}^{(i)})^2$ and $\nu_{j+1}^{(i)} \le 1$. It follows that $\nu_{j+1}^{(i)}$ is at most the least solution of $X_{j+1} = \frac{1}{4}(\nu_j^{(i)})^2 + \frac{1}{2}\nu_j^{(i)} X_{j+1} + \frac{1}{4}(X_{j+1})^2$, and so $\Delta_{j+1}^{(i)} \ge 2\sqrt{\Delta_j^{(i)}} - \Delta_j^{(i)} > \sqrt{\Delta_j^{(i)}}$.

By induction it follows that $\Delta_{j+1}^{(i)} > (\Delta_1^{(i)})^{2^{-j}}$. In particular,

$$\Delta_n^{(k\cdot 2^{n-1})} > \left(\Delta_1^{(k\cdot 2^{n-1})}\right)^{2^{-(n-1)}} = 2^{-k\cdot 2^{n-1}\cdot 2^{-(n-1)}} = 2^{-k}.$$

Hence, after $k \cdot 2^{n-1}$ iterations we have less than k valid bits. □

Notice that the proof exploits that an error in the first component gets "amplified" along the DAG of SCCs. One can also show along those lines that computing $\boldsymbol{\mu}$ is an *ill-conditioned* problem: Consider the SPP $\boldsymbol{g}^{(n,\varepsilon)}$ obtained from $\boldsymbol{f}^{(n)}$ by replacing the first component by $1 - \varepsilon$ where $0 \le \varepsilon < 1$. If $\varepsilon = 0$ then $(\boldsymbol{\mu g}^{(n,\varepsilon)})_n = 1$, whereas if $\varepsilon = \frac{1}{2^{2^{n-1}}}$ then $(\boldsymbol{\mu g}^{(n,\varepsilon)})_n < \frac{1}{2}$. In other words, to get 1 bit of precision of $\boldsymbol{\mu g}$ one needs exponentially (in n) many bits in $\boldsymbol{g}$. Note that this observation is independent from any particular method to compute or approximate the least fixed point.

1.7 Conclusions

We have studied the convergence order and convergence rate of Newton's method for fixed-point equations of systems of positive polynomials (SPP equations). These equations appear naturally in the analysis of several stochastic computational models that have been intensely studied in recent years, and they also play a central role in the theory of stochastic branching processes.

The restriction to positive coefficients leads to strong results. For arbitrary polynomial equations Newton's method may not converge or converge only locally, i.e., when started at a point sufficiently close to the solution. We have extended a result by Etessami and Yannakakis [EY09], and shown that for SPP equations the method always converges starting at $\mathbf{0}$. Moreover, we have proved that the method has at least linear convergence order, and have determined the asymptotic convergence rate. To the best of our knowledge, this is the first time that a lower bound on the convergence order is proved for a significant class of equations with a trivial membership test.[3] Finally, we have also obtained upper bounds on the threshold for strongly connected SPPs, i.e., the number of iterations necessary to reach the "steady state" in which valid bits are computed at the asymptotic rate. These results lead to practical tests for checking whether the least fixed point of an SPP exceeds a given bound.

There are still at least three important open problems.

- We would like to have bounds on the threshold $k_{\boldsymbol{f}}$ not only for strongly connected SPPs, but also for general SPPs.
- The behavior of Newton's method when arithmetic operations only have a fixed accuracy should be further investigated. We wish to develop tests allowing to decide whether the result of applying Newton's method with a certain fixed accuracy is reliable or not.

[3] Notice the contrast with the classical result stating that if $(I - \boldsymbol{f}'(\boldsymbol{\mu}))$ is non-singular, then Newton's method has exponential convergence order; here the membership test is highly non-trivial, and, for what we know, as hard as computing $\boldsymbol{\mu}$ itself.

- Say that Newton's method is *polynomial* for a class of SPP equations if there is a polynomial $p(x, y, z)$ such that for every $k \geq 0$ and for every system in the class with n equations and coefficients of size m, the $p(n, m, k)$-th Newton approximant $\boldsymbol{\nu}^{(p(n,m,k))}$ has k valid bits. We have proved in Theorem 1.40 that Newton's method is polynomial for SPPs $\boldsymbol{f}$ satisfying $\boldsymbol{f}(\mathbf{0}) \succ \mathbf{0}$; for this class one can take $p(n, m, k) = 7mn + k$. We have also exhibited in § 1.6 a class for which computing the first bit of the least solution takes 2^n iterations. The members of this class, however, are non-strongly-connected, and this is the fact we have exploited to construct them. So the following question remains open: Is Newton's method polynomial for strongly connected SPPs?

Chapter 2

Systems of Positive Min-Max-Polynomials

In this chapter we study systems of positive min-max-polynomials (min-max-SPPs) and two variants of Newton's method to compute the least fixed point of min-max-SPPs. In § 2.1 we introduce basic concepts and state some important facts about min-max-SPPs. A class of games which can be analyzed using our techniques is presented in § 2.2. The main contribution of this chapter, the two approximation methods, is presented and analyzed in § 2.3 and § 2.4. In § 2.5 we study the relation between our two approaches and compare them to previous work. We conclude in § 2.6.

2.1 Preliminaries and a Fundamental Theorem

In § 2.1.1 we prove some more properties of SPPs (without min- or max-operators) that were not necessary for the first part of this thesis, but are crucial for our results on min-max-SPPs. Roughly speaking, those SPP properties are extensions of Lemma 1.2 in that they follow from the "convexity" of SPPs.

In § 2.1.2 we formally introduce the concepts of min-max-SPPs and strategies for them. We also show some basic properties of strategies. In particular, Theorem 2.10 is a fundamental theorem on max-SPPs. The proof of this theorem is the main reason for our deeper investigation of SPPs in § 2.1.1.

2.1.1 Power Series and Some Convexity Properties of SPPs

Let $\boldsymbol{f}$ be a function with $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$. As in Chapter 1, we call a vector $\boldsymbol{x}$ a *fixed point* (resp. *prefixed point* resp. *postfixed point*) if $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}$ (resp. $\boldsymbol{f}(\boldsymbol{x}) \geq \boldsymbol{x}$ resp. $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{x}$). Again, functions $\boldsymbol{f}$ that have a fixed point are called *feasible*, and the least fixed point of $\boldsymbol{f}$ is denoted by $\boldsymbol{\mu f}$ or $\boldsymbol{\mu}$.

We generalize SPPs to positive power series in the obvious way: A function $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ is said to be a *positive power series* if each component is a power series with coefficients from $\mathbb{R}_{\geq 0}$. We need power series in the following lemmata leading to the fundamental Theorem 2.10. Notice that "Taylor's theorem" (Lemma 1.2 on page 17) applies also to positive power series (provided that the power series converges at the involved points). Loosely speaking, this lemma expresses that SPPs are "convex". The following lemmata are consequences of this "convexity"-Lemma 1.2.

Lemma 2.1. *Let $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ be an SPP, $S \subseteq \{1, \dots, n\}$ and $k := |S|$. Assume that $\boldsymbol{\mu}(\boldsymbol{f}[S/\boldsymbol{b}])$ exists for some $\boldsymbol{b} \in \mathbb{R}^k_{\geq 0}$. Then $\boldsymbol{f}^* : [\boldsymbol{0}, \boldsymbol{b}] \to \mathbb{R}^{n-k}_{\geq 0}$ defined by $\boldsymbol{f}^*(\boldsymbol{x}) := \boldsymbol{\mu}(\boldsymbol{f}[S/\boldsymbol{x}])$ is a positive power series.*

Proof. W.l.o.g. we can assume that $\boldsymbol{b} \succ \boldsymbol{0}$ and $S = \{1, \dots, k\}$. Let $T := \{k+1, \dots, n\}$. Let $\mathbb{R}^{n-k}_{\geq 0}[\boldsymbol{X}_S]$ denote the set of polynomials over the variables $X_1, \dots, X_k$ with coefficients from $\mathbb{R}^{n-k}_{\geq 0}$. For every $i \in \mathbb{N}$, $(\boldsymbol{f}[S/\boldsymbol{X}_S])^i(\boldsymbol{0})$ can be considered as a polynomial from $\mathbb{R}^{n-k}_{\geq 0}[\boldsymbol{X}_S]$. Moreover, by Kleene's theorem we have $\lim_{i\to\infty}(\boldsymbol{f}[S/\boldsymbol{x}])^i(\boldsymbol{0}) = \boldsymbol{f}^*(\boldsymbol{x})$ for $\boldsymbol{x} \in [\boldsymbol{0}, \boldsymbol{b}]$. For $\alpha \in \mathbb{N}^k$, let $\boldsymbol{c}^{(i)}_\alpha \in \mathbb{R}^{n-k}_{\geq 0}$ denote the coefficient of $\boldsymbol{X}^\alpha_S = X_1^{\alpha_1} \cdot \dots \cdot X_k^{\alpha_k}$ in the polynomial $(\boldsymbol{f}[S/\boldsymbol{X}_S])^i(\boldsymbol{0})$. We show:

(1) $(\boldsymbol{c}^{(i)}_\alpha)_{i\in\mathbb{N}}$ is increasing for every $\alpha \in \mathbb{N}^k$; and

(2) $(\boldsymbol{c}^{(i)}_\alpha)_{i\in\mathbb{N}}$ is bounded for every $\alpha \in \mathbb{N}^k$.

In order to show the first statement, we consider the set $\mathbb{R}^{n-k}_{\geq 0}[\boldsymbol{X}_S]$ of polynomials as partially ordered by setting

$$\sum_{\alpha\in\mathbb{N}^k} \boldsymbol{u}_\alpha \cdot \boldsymbol{X}^\alpha_S \sqsubseteq \sum_{\alpha\in\mathbb{N}^k} \boldsymbol{v}_\alpha \cdot \boldsymbol{X}^\alpha_S \quad \text{if } \boldsymbol{u}_\alpha \leq \boldsymbol{v}_\alpha \text{ for all } \alpha \in \mathbb{N}^k \;.$$

In those terms we need to show that $(\boldsymbol{f}[S/\boldsymbol{X}_S])^i(\boldsymbol{0}) \sqsubseteq (\boldsymbol{f}[S/\boldsymbol{X}_S])^{i+1}(\boldsymbol{0})$ for all $i \in \mathbb{N}$. Notice that the map from $\mathbb{R}^{n-k}_{\geq 0}[\boldsymbol{X}_S]$ to $\mathbb{R}^{n-k}_{\geq 0}[\boldsymbol{X}_S]$ defined by $\boldsymbol{p} \mapsto \boldsymbol{f}[S/\boldsymbol{X}_S](\boldsymbol{p})$ is monotone, i.e., $\boldsymbol{p} \sqsubseteq \boldsymbol{q}$ implies $\boldsymbol{f}[S/\boldsymbol{X}_S](\boldsymbol{p}) \sqsubseteq \boldsymbol{f}[S/\boldsymbol{X}_S](\boldsymbol{q})$. Now we get the first statement by induction on i, i.e., $\boldsymbol{0} \sqsubseteq \boldsymbol{f}[S/\boldsymbol{X}_S](\boldsymbol{0})$ and

$$\begin{aligned}
&(\boldsymbol{f}[S/\boldsymbol{X}_S])^{i+1}(\boldsymbol{0}) \\
&= \boldsymbol{f}[S/\boldsymbol{X}_S]((\boldsymbol{f}[S/\boldsymbol{X}_S])^{i}(\boldsymbol{0})) \\
&\sqsubseteq \boldsymbol{f}[S/\boldsymbol{X}_S]((\boldsymbol{f}[S/\boldsymbol{X}_S])^{i+1}(\boldsymbol{0})) \qquad \text{(monotonicity, induction hypothesis)} \\
&= (\boldsymbol{f}[S/\boldsymbol{X}_S])^{i+2}(\boldsymbol{0}) \;.
\end{aligned}$$

For the second statement, we have $\boldsymbol{f}^*(\boldsymbol{b}) \geq (\boldsymbol{f}[S/\boldsymbol{b}])^i(\boldsymbol{0}) \geq \boldsymbol{c}^{(i)}_\alpha \cdot \boldsymbol{b}^\alpha$, so $(\boldsymbol{c}^{(i)}_\alpha)_{i\in\mathbb{N}}$ must be bounded because $\boldsymbol{b} \succ \boldsymbol{0}$.

As the statements (1) and (2) are now established, it follows that $(\boldsymbol{c}^{(i)}_\alpha)_{i\in\mathbb{N}}$ converges for all $\alpha \in \mathbb{N}^k$. Let $\boldsymbol{c}_\alpha := \lim_{i\to\infty} \boldsymbol{c}^{(i)}_\alpha \in \mathbb{R}^{n-k}_{\geq 0}$. Consider the power series $\sum_{\alpha\in\mathbb{N}^k} \boldsymbol{c}_\alpha \cdot \boldsymbol{x}^\alpha$. By (absolute) convergence on $[\boldsymbol{0}, \boldsymbol{b}]$, we have

$$\boldsymbol{f}^*(\boldsymbol{x}) = \lim_{i\to\infty} (\boldsymbol{f}[S/\boldsymbol{x}])^i(\boldsymbol{0}) = \lim_{i\to\infty} \sum_{\alpha\in\mathbb{N}^k} \boldsymbol{c}^{(i)}_\alpha \cdot \boldsymbol{x}^\alpha = \sum_{\alpha\in\mathbb{N}^k} \lim_{i\to\infty} \boldsymbol{c}^{(i)}_\alpha \cdot \boldsymbol{x}^\alpha = \sum_{\alpha\in\mathbb{N}^k} \boldsymbol{c}_\alpha \cdot \boldsymbol{x}^\alpha$$

for $\boldsymbol{x} \in [\boldsymbol{0}, \boldsymbol{b}]$. Thus, $\boldsymbol{f}^*$ is a positive power series that converges on $[\boldsymbol{0}, \boldsymbol{b}]$. □

We use this lemma for the proof of the following lemma that has been proved implicitly in [EY05c]. However, since we are not restricted to the case of *1-exit recursive simple stochastic games* as in [EY05c], we need a more general statement for our setting.

Lemma 2.2. *Let $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ be a feasible SPP and $\{i\} \dot\cup T = \{1, \dots, n\}$. Let $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ with $x_i < f_i(\boldsymbol{x})$ and $\boldsymbol{x}_T \leq \boldsymbol{\mu}(\boldsymbol{f}[i/x_i])$. Assume that there exists a postfixed point $\boldsymbol{y} \geq \boldsymbol{x}$ of $\boldsymbol{f}$. Then $\boldsymbol{x} \leq \boldsymbol{\mu}$.*

Proof. Assume w.l.o.g. that $i = 1$. Let $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be defined by

$$g(z) = f_1(z, \boldsymbol{\mu}(\boldsymbol{f}[1/z])). \tag{2.1}$$

Note that by assumption and monotonicity of f_1

$$x_1 < f_1(\boldsymbol{x}) = f_1(x_1, \boldsymbol{x}_T) \leq f_1(x_1, \boldsymbol{\mu}(\boldsymbol{f}[1/x_1])) = g(x_1). \tag{2.2}$$

Furthermore, since g is the composition of two positive power series (Lemma 2.1), g is also a positive power series. Let $\boldsymbol{x}^* := \boldsymbol{\mu}$. Since $\boldsymbol{\mu}(\boldsymbol{f}[1/x_1^*]) = \boldsymbol{x}_T^*$ we get

$$g(x_1^*) = f_1(x_1^*, \boldsymbol{\mu}(\boldsymbol{f}[1/x_1^*])) = f_1(x_1^*, \boldsymbol{x}_T^*) = f_1(\boldsymbol{x}^*) = x_1^* \,. \tag{2.3}$$

Case 1: $g'(x_1) \leq 1$. We first show that $x_1^* > x_1$. Suppose for a contradiction $x_1^* \leq x_1$. We get

$$\begin{aligned} x_1 &< g(x_1) && \text{(by (2.2))} \\ &\leq g(x_1^*) + g'(x_1)(x_1 - x_1^*) && \text{(Lemma 1.2)} \\ &\leq g(x_1^*) - x_1^* + x_1 && (g'(x_1) \leq 1) \\ &= x_1 && \text{(by (2.3))} \end{aligned}$$

which is a contradiction. Thus, $x_1^* > x_1$. By monotonicity of $\boldsymbol{\mu}(\boldsymbol{f}[1/\cdot])$ we get $\boldsymbol{x}_T^* = \boldsymbol{\mu}(\boldsymbol{f}[1/x_1^*]) \geq \boldsymbol{\mu}(\boldsymbol{f}[1/x_1]) \geq \boldsymbol{x}_T$. Summarizing we have $\boldsymbol{x}^* = (x_1^*, \boldsymbol{x}_T^*) \geq (x_1, \boldsymbol{x}_T) = \boldsymbol{x}$, so we are done in this case.

Case 2: $g'(x_1) > 1$. We show that this case does not occur. We get:

$$\begin{aligned} g(y_1) &\geq g(x_1) + g'(x_1)(y_1 - x_1) && \text{(Lemma 1.2)} \\ &\geq g(x_1) + y_1 - x_1 && (g'(x_1) \geq 1) \\ &> y_1 && \text{(by (2.2))} \end{aligned}$$

Since $\boldsymbol{y}_T$ is a postfixed point of $\boldsymbol{f}[1/y_1]$, it holds $\boldsymbol{\mu}(\boldsymbol{f}[1/y_1]) \leq \boldsymbol{y}_T$. Thus, by monotonicity of f_1, we get $y_1 < g(y_1) = \boldsymbol{f}_1(y_1, \boldsymbol{\mu}(\boldsymbol{f}[1/y_1])) \leq f_1(y_1, \boldsymbol{y}_T) = f_1(\boldsymbol{y})$, contradicting the assumption that $\boldsymbol{y}$ is a postfixed point of $\boldsymbol{f}$. □

Lemma 2.2 can be generalized by induction as follows.

Lemma 2.3. *Let $\boldsymbol{f}$ be a feasible SPP, $S \dot\cup T = \{1, \ldots, n\}$. Let $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ with $\boldsymbol{x}_S \prec \boldsymbol{f}_S(\boldsymbol{x})$ and $\boldsymbol{x}_T \leq \boldsymbol{\mu}(\boldsymbol{f}[S/\boldsymbol{x}_S])$. Assume that there exists a postfixed point $\boldsymbol{y} \geq \boldsymbol{x}$ of $\boldsymbol{f}$. Then $\boldsymbol{x} \leq \boldsymbol{\mu}$.*

Proof. Let w.l.o.g. $S = \{n - k + 1, \ldots, n\}$. We proceed by induction on k. The base case $(k = 0)$ is trivial.

Let $\widehat{\boldsymbol{f}} := \boldsymbol{f}[n/x_n]$, $\widehat{\boldsymbol{x}} := \boldsymbol{x}_{\{1,\ldots,n-1\}}$, $\widehat{\boldsymbol{y}} := \boldsymbol{y}_{\{1,\ldots,n-1\}}$ and $\widehat{S} := \{n - k + 1, \ldots, n - 1\}$. Then $\widehat{\boldsymbol{x}}_{\widehat{S}} = \boldsymbol{x}_{\widehat{S}} \prec \boldsymbol{f}_{\widehat{S}}(\boldsymbol{x}) = \widehat{\boldsymbol{f}}_{\widehat{S}}(\widehat{\boldsymbol{x}})$ and $\widehat{\boldsymbol{x}}_T = \boldsymbol{x}_T \leq \boldsymbol{\mu}(\boldsymbol{f}[S/\boldsymbol{x}_S]) = \boldsymbol{\mu}(\widehat{\boldsymbol{f}}[\widehat{S}/\widehat{\boldsymbol{x}}_{\widehat{S}}])$. Moreover, $\widehat{\boldsymbol{y}}$ is a postfixed point of $\widehat{\boldsymbol{f}}$, as $\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{y}}) = \boldsymbol{f}_{\{1,\ldots,n-1\}}(\widehat{\boldsymbol{y}}, x_n) \leq \boldsymbol{f}_{\{1,\ldots,n-1\}}(\boldsymbol{y}) \leq \widehat{\boldsymbol{y}}$. By induction hypothesis it follows $\widehat{\boldsymbol{x}} \leq \boldsymbol{\mu}\widehat{\boldsymbol{f}}$. By definition, this amounts to $\boldsymbol{x}_{\{1,\ldots,n-1\}} \leq \boldsymbol{\mu}(\boldsymbol{f}[n/x_n])$. Moreover, $x_n < f_n(\boldsymbol{x})$ and $\boldsymbol{y} \geq \boldsymbol{x}$ is a postfixed point of $\boldsymbol{f}$. So we get $\boldsymbol{x} \leq \boldsymbol{\mu}$ by Lemma 2.2. □

2.1.2 Min-Max-SPPs

The operators $\wedge$ and $\vee$ are defined by $x \wedge y := \min\{x, y\}$ and $x \vee y := \max\{x, y\}$ for $x, y \in \mathbb{R}$. These operators are also extended component-wise to $\mathbb{R}^n$ and point-wise to $\mathbb{R}^n$-valued functions. Given polynomials $f_1, \ldots, f_k$ we call $f_1 \wedge \cdots \wedge f_k$ a *min-polynomial* and $f_1 \vee \cdots \vee f_k$ a *max-polynomial*. Min- and max-polynomials are also called *min-max-polynomials*. We call $\boldsymbol{f} = (f_1, \ldots, f_n)^\top$ a *system of min-max-polynomials* if every component f_i is a min-max-polynomial. A system of min-max-polynomials is called *linear* (resp. *quadratic*) if all occurring polynomials are linear (resp. *quadratic*), i.e., they are of degree at most 1 (resp. degree at most 2). By introducing auxiliary variables every system of min-max-polynomials can be transformed into a *quadratic* one in a time linear in the size of the system (as in § 1.3.4 in the first part of the thesis). A system of min-max-polynomials where all coefficients are from $\mathbb{R}^n_{\geq 0}$ is called *system of positive min-max-polynomials*, or *min-max-SPP* for short. The terms *min-SPP* and *max-SPP* are defined analogously.

Example 2.4. *Consider the quadratic 2-dimensional min-max-SPP* $\boldsymbol{f}$ *with*

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} f_1(X_1, X_2) \\ f_2(X_1, X_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}X_2^2 + \frac{1}{2} \ \wedge \ 3 \\ X_1 \ \vee \ 2 \end{pmatrix} .$$

The graphs of the corresponding SPP equations $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ *and the least fixed point* $\boldsymbol{\mu} = (3, 3)^\top$ *are shown in Figure 2.1.*

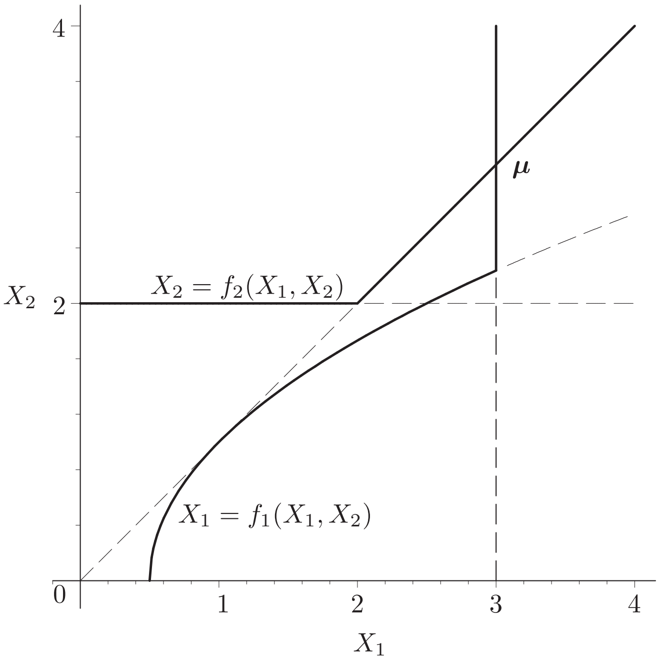

Figure 2.1: Graphs of the equations $X_1 = f_1(X_1, X_2)$ and $X_2 = f_2(X_1, X_2)$ in Example 2.4. The least fixed point $\boldsymbol{\mu} = (3, 3)^\top$ is also shown.

Min-max-SPPs, like SPPs, can be considered as monotone continuous mappings from $\mathbb{R}^n_{\geq 0}$ to $\mathbb{R}^n_{\geq 0}$, so Kleene's fixed point theorem is again applicable (cf. Theorem 1.3):

Theorem 2.5 (Kleene's fixed point theorem for min-max-SPPs)**.** *Every feasible min-max-SPP* $\boldsymbol{f}$ *has a least fixed point* $\boldsymbol{\mu}$ *in* $\mathbb{R}^n_{\geq 0}$. *Moreover, the* Kleene sequence $(\boldsymbol{\kappa}_{\boldsymbol{f}}^{(k)})_{k \in \mathbb{N}}$ *with* $\boldsymbol{\kappa}_{\boldsymbol{f}}^{(k)} = \boldsymbol{f}^k(\boldsymbol{0})$ *is monotonically increasing and converges to* $\boldsymbol{\mu}$*.*

Example 2.6. *The Kleene sequence for the min-max-SPP from Example 2.4 is:*

$$\boldsymbol{\kappa}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(1)} = \begin{pmatrix} 1/2 \\ 2 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(2)} = \begin{pmatrix} 5/2 \\ 2 \end{pmatrix},$$

$$\boldsymbol{\kappa}^{(3)} = \begin{pmatrix} 5/2 \\ 5/2 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(4)} = \begin{pmatrix} 3 \\ 5/2 \end{pmatrix}, \quad \boldsymbol{\kappa}^{(i)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \text{ for } i \geq 5$$

In this particular example, the Kleene sequence does not only converge to $\boldsymbol{\mu}$, but even reaches $\boldsymbol{\mu}$ after finitely many iterations. This is by no means always so, as we have seen in Chapter 1 even for SPPs without minimum or maximum operator.

Strategies. Let $\boldsymbol{f}$ denote a system of min-max-polynomials. A $\vee$-*strategy* σ for $\boldsymbol{f}$ picks for each max-polynomial a polynomial occurring in it. Formally, a $\vee$-strategy maps each component i of $\boldsymbol{f}$ $(1 \leq i \leq n)$ to a min-polynomial such that

$$\sigma(i) = \begin{cases} f_i & \text{if } f_i \text{ is a (min-)polynomial} \\ f_{i,j} \text{ (where } 1 \leq j \leq k_i) & \text{if } f_i \text{ is a max-polynomial } f_i = f_{i,1} \vee \cdots \vee f_{i,k_i} \end{cases}$$

The $\wedge$-strategies π are defined accordingly. We also write f_i^σ for $\sigma(i)$ and f_i^π for $\pi(i)$. We denote the set of $\vee$-strategies for $\boldsymbol{f}$ by $\Sigma_{\boldsymbol{f}}$ and the set of $\wedge$-strategies for $\boldsymbol{f}$ by $\Pi_{\boldsymbol{f}}$. For $s \in \Sigma_{\boldsymbol{f}} \cup \Pi_{\boldsymbol{f}}$, we write $\boldsymbol{f}^s$ for $(f_1^s, \ldots, f_n^s)^\top$. We define $\Pi^*_{\boldsymbol{f}} := \{\pi \in \Pi_{\boldsymbol{f}} \mid \boldsymbol{f}^\pi \text{ is feasible}\}$ where we drop the subscript when it is understood.

Example 2.7. *Consider the min-max-SPP $\boldsymbol{f}$ from Example 2.4. Then the map π with $\pi(1) = 3$ and $\pi(2) = X_1 \vee 2$ is a $\wedge$-strategy. The max-SPP $\boldsymbol{f}^\pi$ is given by*

$$\boldsymbol{f}^\pi(\boldsymbol{X}) = \begin{pmatrix} 3 \\ X_1 \vee 2 \end{pmatrix}.$$

We collect some elementary facts concerning strategies:

Lemma 2.8. *Let $\boldsymbol{f}$ be a feasible min-max-SPP. Then*

(1) $\boldsymbol{\mu f}^\sigma \leq \boldsymbol{\mu f}$ for every $\sigma \in \Sigma$;

(2) $\boldsymbol{\mu f}^\pi \geq \boldsymbol{\mu f}$ for every $\pi \in \Pi^*$;

(3) $\boldsymbol{\mu f}^\pi = \boldsymbol{\mu f}$ for some $\pi \in \Pi^*$.

Proof. Observe that, for $\sigma \in \Sigma$, $\boldsymbol{\mu f}$ is a postfixed point of $\boldsymbol{f}^\sigma$. Thus, Knaster-Tarski's theorem implies the first statement. Similarly, the fact that, for $\pi \in \Pi^*$, $\boldsymbol{\mu f}^\pi$ is a postfixed point of $\boldsymbol{f}$ implies the second statement. For the third statement observe that there exists some $\pi \in \Pi$ such that $\boldsymbol{\mu f}$ is a fixed point of $\boldsymbol{f}^\pi$. Thus $\pi \in \Pi^*$ and $\boldsymbol{\mu f}^\pi \leq \boldsymbol{\mu f}$. Since $\boldsymbol{\mu f}^\pi \geq \boldsymbol{\mu f}$ by statement (2), we obtain $\boldsymbol{\mu f} = \boldsymbol{\mu f}^\pi$. □

In [EY05c] the authors consider a class of stochastic games (so-called 1-exit recursive simple stochastic games), for which they prove that a positional optimal strategy exists for the player who wants to maximize the outcome (Theorem 2 of [EY05c]). The outcome of such a game is the least fixed point of some min-max-SPP $\boldsymbol{f}$. In our terminology, Theorem 2 of [EY05c] states that there exists a $\vee$-strategy σ such that $\boldsymbol{\mu f}^\sigma = \boldsymbol{\mu f}$ if $\boldsymbol{f}$ is derived from such a recursive stochastic game. The following example shows that this does not hold for arbitrary min-max-SPPs.

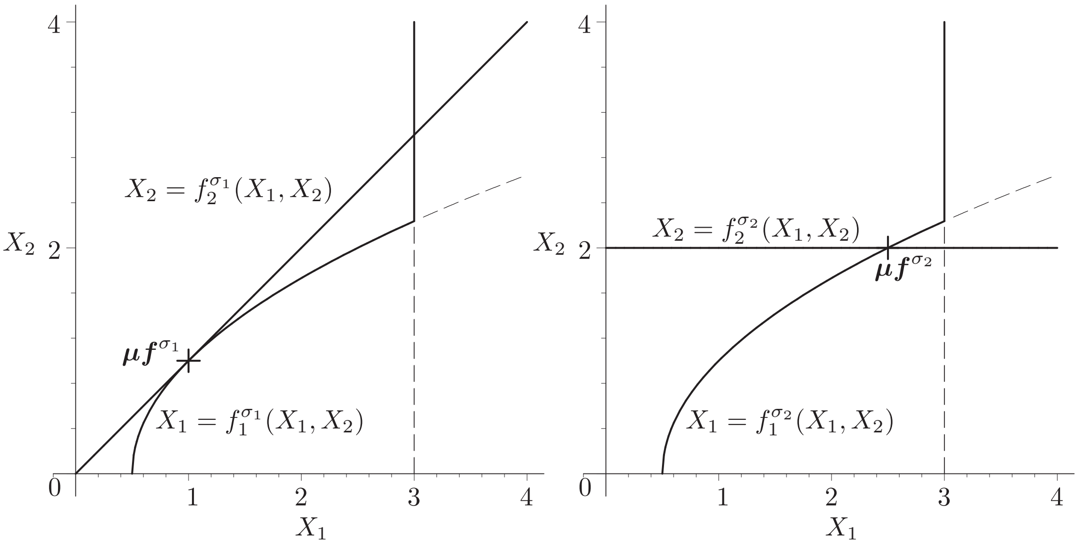

Figure 2.2: Two different $\vee$-strategies are applied to $\boldsymbol{f}$. The left and the right side show plots of $\boldsymbol{X} = \boldsymbol{f}^{\sigma_1}(\boldsymbol{X})$ and $\boldsymbol{X} = \boldsymbol{f}^{\sigma_2}(\boldsymbol{X})$, respectively.

Example 2.9. *Consider the min-max-SPP $\boldsymbol{f}$ from Example 2.4. Let the $\vee$-strategies $\sigma_1, \sigma_2 \in \Sigma$ be defined by $\sigma_1(2) = X_1$ and $\sigma_2(2) = 2$. Figure 2.2 shows the graphs of $\boldsymbol{X} = \boldsymbol{f}^{\sigma_1}(\boldsymbol{X})$ and $\boldsymbol{X} = \boldsymbol{f}^{\sigma_2}(\boldsymbol{X})$. We have $\boldsymbol{\mu f}^{\sigma_1} = (1,1)^\top$, $\boldsymbol{\mu f}^{\sigma_2} = (\frac{5}{2}, 2)^\top$, but $\boldsymbol{\mu} = (3,3)^\top$. Note that no $\vee$-strategy σ exists such that $\boldsymbol{\mu f}^{\sigma} = \boldsymbol{\mu}$.*

But for feasible max-SPPs the following fundamental result, Theorem 2.10, is retained. It generalizes Theorem 2 of [EY05c]. Although the statement of Theorem 2.10 looks very natural, we need the machinery developed in § 2.1.1 for the proof.

Theorem 2.10. *Let $\boldsymbol{f}$ be a feasible max-SPP. Then $\boldsymbol{\mu f}^{\sigma} = \boldsymbol{\mu f}$ for some $\sigma \in \Sigma$.*

Proof. The proof is inspired by a proof of [EY05c]. Suppose for a contradiction that $\boldsymbol{\mu f}^{\sigma} < \boldsymbol{\mu}$ for every $\vee$-strategy $\sigma \in \Sigma$. Let σ be any strategy and $\boldsymbol{x} := \boldsymbol{\mu f}^{\sigma}$. We have $\boldsymbol{x} = \boldsymbol{f}^{\sigma}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{x})$. Since by assumption $\boldsymbol{x} < \boldsymbol{\mu}$, there exists some $i \in \{1, \ldots, n\}$ such that $x_i < f_i(\boldsymbol{x})$. Let $S = \{i\}$ and $T := \{1, \ldots, n\} \setminus S$. Choose a strategy σ' such that $f_i^{\sigma'}(\boldsymbol{x}) = f_i(\boldsymbol{x}) > x_i$ and $\sigma'(j) = \sigma(j)$ for every $j \in \{1, \ldots, n\} \setminus \{i\}$. We will apply Lemma 2.3. Observe that by construction $\boldsymbol{f}^{\sigma'}[i/y] = \boldsymbol{f}^{\sigma}[i/y]$ for $y \in \mathbb{R}_{\geq 0}$ and thus in particular $\boldsymbol{\mu f}^{\sigma'}[i/\boldsymbol{x}_S] = \boldsymbol{\mu f}^{\sigma}[i/\boldsymbol{x}_S] = \boldsymbol{x}_T$. We get $\boldsymbol{\mu} \geq \boldsymbol{f}^{\sigma'}(\boldsymbol{\mu})$ by Lemma 2.8.1, i.e., $\boldsymbol{\mu} \geq \boldsymbol{x}$ is a postfixed point of $\boldsymbol{f}^{\sigma'}$. Thus, Lemma 2.3 implies $\boldsymbol{\mu f}^{\sigma'} \geq \boldsymbol{x} = \boldsymbol{\mu f}^{\sigma}$. Since $\boldsymbol{x}$ is not a fixed point of $\boldsymbol{f}^{\sigma'}$ we have $\boldsymbol{\mu f}^{\sigma'} > \boldsymbol{x} = \boldsymbol{\mu f}^{\sigma}$. Thus, we have shown that, for every strategy σ, there exists a strategy σ' such that $\boldsymbol{\mu f}^{\sigma'} > \boldsymbol{\mu f}^{\sigma}$. This contradicts the fact that there are only finitely many different strategies. □

2.2 A Class of Applications: Extinction Games

In order to illustrate the interest of min-max-SPPs we consider *extinction games*, which are special stochastic games. Consider a world of n different species $s_1, \ldots, s_n$. Each species s_i is controlled by one of two adversarial players. For each s_i there is a non-empty set A_i of actions. An action $a \in A_i$ replaces a single individual of species s_i by other individuals specified by the action a. The actions can be probabilistic. E.g., an action could transform an adult rabbit to zero individuals with probability 0.2, to an adult rabbit with probability 0.3 and to an adult and a baby rabbit with probability 0.5. Another action could transform an adult rabbit to a fat rabbit. The terminator (savior) wants to maximize (minimize) the probability that some initial population is extinguished. During the game each player continuously chooses an individual of a species s_i controlled by her/him and applies an action from A_i to it. Note that actions on different species are never in conflict and the execution order is irrelevant. What is the probability that the population is extinguished if the players follow optimal strategies?

To answer those questions we set up a min-max-SPP $\boldsymbol{f}$ with one min-max-polynomial for each species, thereby following [Har63, EY05c]. The variables X_i represent the probability that a population with only a single individual of species s_i is extinguished. In the rabbit example we have $X_{\text{adult}} = 0.2 + 0.3X_{\text{adult}} + 0.5X_{\text{adult}}X_{\text{baby}} \vee X_{\text{fat}}$, assuming that the adult rabbits are controlled by the terminator. The probability that an initial population with p_i individuals of species s_i is extinguished is given by $\prod_{i=1}^{n}(\mu_i)^{p_i}$. The stochastic termination games of [EY05c, EY06, WE07] can be considered as extinction games.

In Example 0.8 (page 11) we already gave an example of an extinction game. In that example, a doctor has two different treatment options. This leads to a max-polynomial, because the doctor will choose her action to maximize the probability of extinguishing the flu. Since the doctor wishes to base her decision on a worst-case assumption on the type of the flu, the flu was modeled as another "player" which chooses the flu type. This leads to a min-polynomial, because the flu player will pick the flu type that minimizes the probability of extinguishing the disease.

The notions of *individuals* and *species* can be interpreted quite broadly. This is illustrated by the following example where each species corresponds to a certain problem, and the numbers of individuals of each species model the severity of the corresponding problem.

Example 2.11 (The primaries game). *In the primaries of the 2008 elections of the US president, the candidates of the Democratic Party are Hillary Clinton and Barack Obama. Hillary Clinton has to decide her strategy. Her team estimates that undecided voters have not yet decided to vote for her for three possible reasons: they consider her (a) cold and calculating, (b) too much part of Washington's establishment, or (c) they listen to Obama's campaign. So the team decides to model those problems as species in an extinction game. The larger the population of a species, the more influenced is an undecided voter by the problem. The goal of Clinton's team is to maximize the extinction probabilities.*

Clinton's possible actions for problem (a) are showing emotions *or* concentrating on her program. *If she shows emotions, her team estimates that the individual of problem (a) is removed with probability* 0.3*, but with probability* 0.7 *the action backfires and produces yet another individual of (a). This and the effect of concentrating on her program can be read off from Equation* (2.4) *below. For problem (b), Clinton can choose between concentrating on her voting record or her statement "I'll be ready from day 1". Her team estimates the effect as given in Equation* (2.5)*. Problem (c) is controlled by Obama, who has the choice between his "change" message, or attacking Clinton for her position on Iraq, see Equation* (2.6)*.*

$$\begin{aligned} X_a &= 0.3 + 0.7X_a^2 \quad \vee \quad 0.1 + 0.9X_c && (2.4)\\ X_b &= 0.1 + 0.9X_c \quad \vee \quad 0.4X_b + 0.3X_c + 0.3 && (2.5)\\ X_c &= 0.5X_b + 0.3X_b^2 + 0.2 \quad \wedge \quad 0.5X_a + 0.4X_aX_b + 0.1X_b && (2.6)\end{aligned}$$

What should Clinton and Obama do? What are the extinction probabilities, assuming perfect strategies? In the next sections we show how to efficiently solve these problems.

2.3 The $\boldsymbol{\tau}$-Method

Let $\boldsymbol{f}$ denote a feasible min-max-SPP. In this section we present our first method for computing $\boldsymbol{\mu}$ approximatively. We call it $\boldsymbol{\tau}$-method. This method computes, for each approximant $\boldsymbol{\tau}^{(k)}$, the next approximant $\boldsymbol{\tau}^{(k+1)}$ as the least fixed point of a piecewise linear approximation $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{\tau}^{(k)}\right) \vee \boldsymbol{\tau}^{(k)}$ (see below) of $\boldsymbol{f}$ at $\boldsymbol{\tau}^{(k)}$. This approximation is a system of *linear* min-max-polynomials where all coefficients of monomials of degree 1 are nonnegative. We call such a system a *monotone linear min-max-system* (*min-max-SML* for short). Note that a min-max-SML $\boldsymbol{f}$ is not necessarily a min-max-SPP, since negative coefficients of monomials of degree 0 are allowed, e.g., the min-max-SML $f(X_1) = X_1 - 1$ is not a min-max-SPP.

[GS07b] considers equation systems of the form $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ where $\boldsymbol{f}$ is a min-max-SML.[1] We identify a min-max-SML $\boldsymbol{f}$ with its interpretation as a function from $\overline{\mathbb{R}}^n$ to $\overline{\mathbb{R}}^n$ ($\overline{\mathbb{R}}$ denotes the complete lattice $\mathbb{R} \cup \{-\infty, \infty\}$). Since $\boldsymbol{f}$ is monotone on $\overline{\mathbb{R}}^n$, it has a least fixed point $\boldsymbol{\mu} \in \overline{\mathbb{R}}^n$ which can be computed using the strategy improvement algorithm from [GS07b].

We are going to use an analogue of Lemma 2.8 for min-max-SMLs. For completeness, we state and prove it here:

Lemma 2.12. *Let $\boldsymbol{f}$ be a min-max-SML. Then*

(1) $\boldsymbol{\mu f}^{\sigma} \leq \boldsymbol{\mu f}$ for every $\sigma \in \Sigma$;

(2) $\boldsymbol{\mu f}^{\pi} \geq \boldsymbol{\mu f}$ for every $\pi \in \Pi$;

(3) $\boldsymbol{\mu f}^{\pi} = \boldsymbol{\mu f}$ for some $\pi \in \Pi$.

Proof. Observe that, for $\sigma \in \Sigma$, $\boldsymbol{\mu f}$ is a postfixed point of $\boldsymbol{f}^{\sigma}$. Thus, Knaster-Tarski's theorem implies the first statement. Similarly, the fact that, for $\pi \in \Pi$, $\boldsymbol{\mu f}^{\pi}$ is a postfixed point of $\boldsymbol{f}$ implies the second statement. For the third statement observe that there exists a $\pi \in \Pi$ such that $\boldsymbol{\mu f}$ is a fixed point of $\boldsymbol{f}^{\pi}$. Thus $\boldsymbol{\mu f}^{\pi} \leq \boldsymbol{\mu f}$. Since $\boldsymbol{\mu f}^{\pi} \geq \boldsymbol{\mu f}$ by statement (2), we obtain $\boldsymbol{\mu f} = \boldsymbol{\mu f}^{\pi}$. □

Given a min-max-SPP $\boldsymbol{f}$ we now define the min-max-SML $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y})$, a piecewise linear approximation of $\boldsymbol{f}$ at $\boldsymbol{y}$. In a first step, consider a multivariate polynomial $f : \mathbb{R}^n \to \mathbb{R}$. Given some approximant $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$, a linear approximation $\mathcal{L}(f, \boldsymbol{y}) : \mathbb{R}^n \to \mathbb{R}$ of f at $\boldsymbol{y}$ is given by the first-order Taylor approximation at $\boldsymbol{y}$, i.e.,

$$\mathcal{L}(f, \boldsymbol{y})(\boldsymbol{x}) := f(\boldsymbol{y}) + f'(\boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y}), \qquad \boldsymbol{x} \in \mathbb{R}^n.$$

This is precisely the linear approximation which is used for Newton's method. Now consider a max-polynomial $f = f_1 \vee \cdots \vee f_k$. We define the approximation $\mathcal{L}(f, \boldsymbol{y}) : \mathbb{R}^n \to \mathbb{R}$ of f at $\boldsymbol{y}$ by

$$\mathcal{L}(f, \boldsymbol{y}) := \mathcal{L}(f_1, \boldsymbol{y}) \vee \cdots \vee \mathcal{L}(f_k, \boldsymbol{y}).$$

Notice that in this case, $\mathcal{L}(f, \boldsymbol{y})$ is in general not a linear function but a linear max-polynomial. Similarly, for a min-polynomial $f = f_1 \wedge \cdots \wedge f_k$, we define

$$\mathcal{L}(f, \boldsymbol{y}) := \mathcal{L}(f_1, \boldsymbol{y}) \wedge \cdots \wedge \mathcal{L}(f_k, \boldsymbol{y}).$$

In this case $\mathcal{L}(f, \boldsymbol{y})$ is a linear min-polynomial. Finally, for a min-max-SPP $\boldsymbol{f}$, we define the approximation $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y}) : \mathbb{R}^n \to \mathbb{R}^n$ of $\boldsymbol{f}$ at $\boldsymbol{y}$ by

$$\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y}) := (\mathcal{L}(f_1, \boldsymbol{y}), \ldots, \mathcal{L}(f_n, \boldsymbol{y}))^\top$$

which is a min-max-SML.

[1] Such equation systems are called *system of rational equations* in [GS07b].

Example 2.13. *Consider the 1-dimensional min-max-SPP f with $f(X) = g(X) \wedge h(X)$ and*

$$g(X) = 0.8X^2 + 0.4X + 0.1 \quad \text{and} \quad h(X) = 0.6X^2 + 0.4\,.$$

We have

$$\mathcal{L}\,(g,0)\,(X) = 0.4X + 0.1 \quad \text{and} \quad \mathcal{L}\,(h,0)\,(X) = 0.4\,.$$

It follows

$$\mathcal{L}\,(f,0)\,(X) = 0.4X + 0.1 \;\wedge\; 0.4\,.$$

Figure 2.3 shows graphs of these functions.

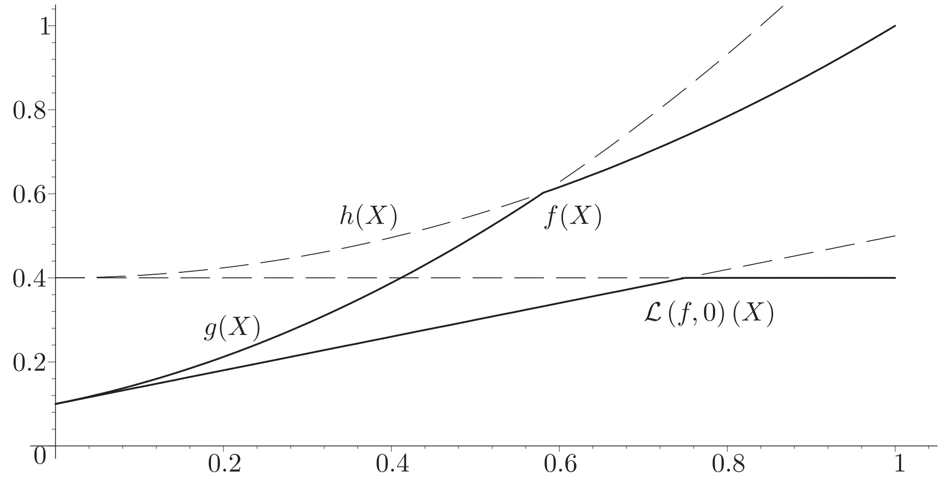

Figure 2.3: In this example we have $f(X) = g(X) \wedge h(X)$. Consequently, we have $\mathcal{L}\,(f,0)\,(X) = \mathcal{L}\,(g,0)\,(X) \wedge \mathcal{L}\,(h,0)\,(X)$.

Example 2.14. *Consider the min-max-SPP $\boldsymbol{f}$ from Example 2.4 with*

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} f_1(X_1, X_2) \\ f_2(X_1, X_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}X_2^2 + \frac{1}{2} \;\wedge\; 3 \\ X_1 \;\vee\; 2 \end{pmatrix}\,.$$

The approximation $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, (1/2,\, 1/2)^\top\right)$ is given by

$$\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}\right)(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{2}X_2 + \frac{3}{8} \wedge 3 \\ X_1 \vee 2 \end{pmatrix}\,.$$

Now we can define the *Newton operator* $\mathcal{N}_{\boldsymbol{f}} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ for min-max-SPPs as follows:

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) := \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}\,(\boldsymbol{f}, \boldsymbol{x}) \vee \boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^n_{\geq 0}.$$

Observe that $\boldsymbol{\mathcal{L}}\,(\boldsymbol{f}, \boldsymbol{x}) \vee \boldsymbol{x}$ is a min-max-SML (that is, after introducing auxiliary variables in order to eliminate components which contain both $\vee$- and $\wedge$-operators, cf. § 1.3.4). To compute the least fixed point of a min-max-SML, one can use the strategy improvement algorithm from [GS07b].

Example 2.15. *Let $\boldsymbol{f}$ be the min-max-SPP from Example 2.14. We wish to apply the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ to the point $\boldsymbol{v} = (1/2,\, 1/2)^\top$. For that we need to find the least solution of $\boldsymbol{X} = \overline{\boldsymbol{f}}(\boldsymbol{X})$ with*

$$\overline{\boldsymbol{f}}(\boldsymbol{X}) = (\boldsymbol{\mathcal{L}}\,(\boldsymbol{f}, \boldsymbol{v}) \vee \boldsymbol{v})\,(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{2}X_2 + \frac{3}{8} \;\wedge\; 3 \\ X_1 \vee 2 \end{pmatrix} \vee \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \left(\frac{1}{2}X_2 + \frac{3}{8} \;\wedge\; 3\right) \;\vee\; \frac{1}{2} \\ X_1 \vee 2 \end{pmatrix}\,. \quad (2.7)$$

Strictly speaking, $\overline{\boldsymbol{f}}$ is not a min-max-SML, because the first component contains both a minimum and a maximum operator. This could be fixed by introducing an auxiliary variable Y:

$$\begin{pmatrix} X_1 \\ X_2 \\ Y \end{pmatrix} = \begin{pmatrix} Y \vee \frac{1}{2} \\ X_1 \vee 2 \\ \frac{1}{2}X_2 + \frac{3}{8} \wedge 3 \end{pmatrix}$$

For illustration purposes we stick to the 2-dimensional "min-max-SML" $\overline{\boldsymbol{f}}$ from (2.7)*. Figure 2.4 shows the graphs of the equation system $\boldsymbol{X} = \overline{\boldsymbol{f}}(\boldsymbol{X})$ and the least fixed point $\boldsymbol{\mu}\overline{\boldsymbol{f}} = (11/8,\, 2)^\top$, which is, by definition, equal to $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{v})$. Notice that all pieces of the graphs are straight lines, cf. the non-linearized system from Figure 2.1 (page 55). The least fixed point of a min-max-SML can be algorithmically computed with the method from [GS07b].*

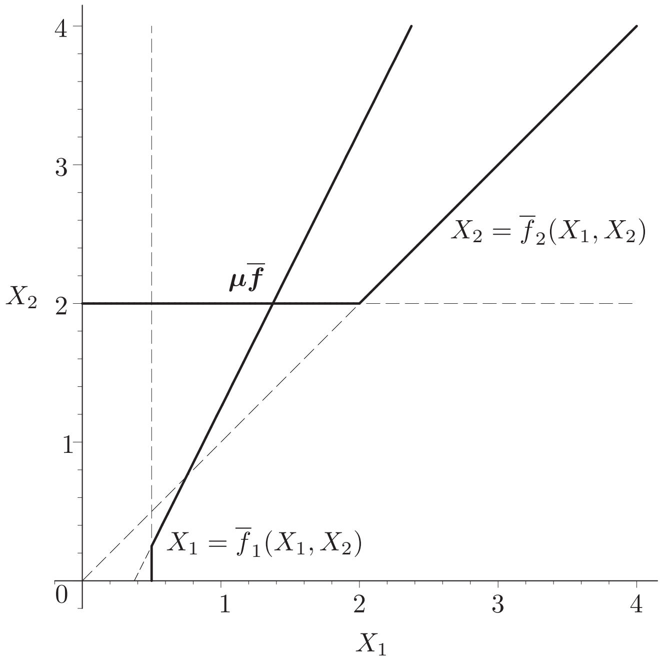

Figure 2.4: Graphs of the equation system $\boldsymbol{X} = \overline{\boldsymbol{f}}(\boldsymbol{X})$ where $\overline{\boldsymbol{f}}$ is a min-max-SML.

We collect basic properties of $\mathcal{N}_{\boldsymbol{f}}$ in the following lemma:

Lemma 2.16. *Let $\boldsymbol{f}$ be a feasible min-max-SPP and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. Then:*

(1) $\boldsymbol{x} \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x}) \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$;

(2) $\boldsymbol{x} = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$ whenever $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$;

(3) (Monotonicity of $\mathcal{N}_{\boldsymbol{f}}$) $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$ whenever $\boldsymbol{x} \leq \boldsymbol{y}$;

(4) $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \leq \boldsymbol{f}(\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}))$ whenever $\boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x})$;

(5) $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \geq \mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{x})$ for every $\vee$-strategy $\sigma \in \Sigma$;

(6) $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \leq \mathcal{N}_{\boldsymbol{f}^\pi}(\boldsymbol{x})$ for every $\wedge$-strategy $\pi \in \Pi$;

(7) $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) = \mathcal{N}_{\boldsymbol{f}^\pi}(\boldsymbol{x})$ for some $\wedge$-strategy $\pi \in \Pi$.

Proof. We show the seven statements in turn.

(1) Let $\boldsymbol{x}^* := \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$. The first inequality holds, as $\boldsymbol{x} \leq \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{x}\right) \vee \boldsymbol{x}) = \boldsymbol{x}^*$. For the second inequality, observe that we have $\boldsymbol{x}^* \geq \boldsymbol{x}$ and $f'_{i,j}(\boldsymbol{x}) \geq 0$ and thus $f'_{i,j}(\boldsymbol{x})(\boldsymbol{x}^* - \boldsymbol{x}) \geq 0$. Hence we have

$$x_i^* = x_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{x}^* - \boldsymbol{x})) \geq x_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} f_{i,j}(\boldsymbol{x}) = x_i \vee f_i(\boldsymbol{x})$$

for $\Box \in \{\bigwedge, \bigvee\}$.

(2) Let $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$ and $i \in \{1, \dots, n\}$. Assume that $f_i = \Box_{j \in \{1,\dots,k\}} f_{i,j}$ where $\Box \in \{\bigvee, \bigwedge\}$. Then

$$\begin{aligned} x_i &= f_i(\boldsymbol{x}) \\ &= x_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} f_{i,j}(\boldsymbol{x}) \\ &= x_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}) \\ &= x_i \vee \mathcal{L}\left(f_i, \boldsymbol{x}\right)(\boldsymbol{x}) . \end{aligned}$$

Hence $\boldsymbol{x}$ is a fixed point of $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{x}\right) \vee \boldsymbol{x}$ and we have $\boldsymbol{x} \geq \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{x}\right) \vee \boldsymbol{x}) = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \geq \boldsymbol{x}$ using statement (1).

(3) Let $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{y}^* := \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$ and $i \in \{1, \dots, n\}$. Assume that $f_i = \Box_{j \in \{1,\dots,k\}} f_{i,j}$ where $\Box \in \{\bigvee, \bigwedge\}$. We have:

$$\begin{aligned} & \boldsymbol{y}_i^* \\ &= y_i \vee \mathcal{L}\left(f_i, \boldsymbol{y}\right)(\boldsymbol{y}^*) \\ &= y_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{y}) + f'_{i,j}(\boldsymbol{y})(\boldsymbol{y}^* - \boldsymbol{y})) \\ &\geq y_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) + f'_{i,j}(\boldsymbol{y})(\boldsymbol{y}^* - \boldsymbol{y})) && \text{(Lemma 1.2)} \\ &\geq y_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{y}^* - \boldsymbol{y})) && (\boldsymbol{y}^* - \boldsymbol{y} \geq \boldsymbol{0}) \\ &= y_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{y}^* - \boldsymbol{x})) && (f'_{i,j}(\boldsymbol{x}) \text{ linear}) \\ &\geq x_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{y}^* - \boldsymbol{x})) \\ &= x_i \vee \mathcal{L}\left(f_i, \boldsymbol{x}\right)(\boldsymbol{y}^*) \end{aligned}$$

So $\boldsymbol{y}^*$ is a postfixed point of $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{x}\right) \vee \boldsymbol{x}$ which implies, using Knaster-Tarski's fixed point theorem, that $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{x}\right) \vee \boldsymbol{x}) \leq \boldsymbol{y}^* = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$.

(4) Let $\boldsymbol{x}^* := \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$. Assume that $f_i = \Box_{j \in \{1,\dots,k\}} f_{i,j}$ where $\Box \in \{\bigvee, \bigwedge\}$. Then:

$$\begin{aligned} & f_i(\boldsymbol{x}^*) \\ &\geq f_i(\boldsymbol{x}) \vee f_i(\boldsymbol{x}^*) && \text{(stmt. (1), } f_i \text{ monotone)} \\ &\geq x_i \vee f_i(\boldsymbol{x}^*) && (\boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x})) \\ &\geq x_i \vee \mathop{\Box}_{j \in \{1,\dots,k\}} (f_{i,j}(\boldsymbol{x}) + f'_{i,j}(\boldsymbol{x})(\boldsymbol{x}^* - \boldsymbol{x})) && \text{(stmt. (1), Lemma 1.2)} \\ &= x_i \vee \mathcal{L}\left(f_i, \boldsymbol{x}\right)(\boldsymbol{x}^*) && (\text{definition of } \mathcal{L}\left(f_i, \boldsymbol{x}\right)) \\ &= x_i^* && (\boldsymbol{x}^* \text{ fixed point of } \boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{x}\right)) \end{aligned}$$

As $i \in \{1, \dots, n\}$ is arbitrary, we have $\boldsymbol{f}(\boldsymbol{x}^*) \geq \boldsymbol{x}^*$.

(5) Let $\sigma \in \Sigma$ and $\boldsymbol{x}^* := \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$. We have to show that $x_i^* = x_i \vee \mathcal{L}(f_i, \boldsymbol{x})(\boldsymbol{x}^*) \geq x_i \vee \mathcal{L}(f_i^\sigma, \boldsymbol{x})(\boldsymbol{x}^*)$ for $i = 1, \ldots, n$, i.e., $\boldsymbol{x}^*$ is a postfixed point of $\mathcal{L}(f^\sigma, \boldsymbol{x}) \vee \boldsymbol{x}$. Then Knaster-Tarski's fixed point theorem implies that $\boldsymbol{x}^* \geq \boldsymbol{\mu}(\mathcal{L}(f^\sigma, \boldsymbol{x}) \vee \boldsymbol{x}) = \mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{x})$. Let $i \in \{1, \ldots, n\}$. Assume that $f_i = \bigvee_{j=1,\ldots,k} f_{i,j}$ and that $f_i^\sigma = f_{i,j_0}$. Then

$$\begin{aligned} x_i^* &= x_i \vee \mathcal{L}(f_i, \boldsymbol{x})(\boldsymbol{x}^*) = x_i \vee \bigvee_{j=1,\ldots,k} \mathcal{L}(f_{i,j}, \boldsymbol{x})(\boldsymbol{x}^*) \\ &\geq x_i \vee \mathcal{L}(f_{i,j_0}, \boldsymbol{x})(\boldsymbol{x}^*) = x_i \vee \mathcal{L}(f_i^\sigma, \boldsymbol{x})(\boldsymbol{x}^*). \end{aligned}$$

Assume now that $f_i = \bigwedge_{j=1,\ldots,k} f_{i,j}$. Then $f_i^\sigma = f_i$, so there is nothing to show.

(6) Is shown analogously.

(7) Let $\boldsymbol{g} := \boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{x}) \vee \boldsymbol{x}$. Then $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{\mu g}$. By Lemma 2.12.3 there is a $\wedge$-strategy $\pi' \in \Pi_{\boldsymbol{g}}$ with $\boldsymbol{\mu g}^{\pi'} = \boldsymbol{\mu g}$. Observe that there exists a $\wedge$-strategy $\pi \in \Pi_{\boldsymbol{f}}$ which corresponds to the $\wedge$-strategy $\pi' \in \Pi_{\boldsymbol{g}}$, i.e., $\boldsymbol{\mathcal{L}}(\boldsymbol{f}^\pi, \boldsymbol{x}) \vee \boldsymbol{x} = \boldsymbol{g}^{\pi'}$. Hence we have

$$\mathcal{N}_{\boldsymbol{f}^\pi}(\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}(\boldsymbol{f}^\pi, \boldsymbol{x}) \vee \boldsymbol{x}) = \boldsymbol{\mu g}^{\pi'} = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}). \qquad \square$$

In particular, Lemma 2.16 implies that the least fixed point of $\mathcal{N}_{\boldsymbol{f}}$ is equal to the least fixed point of $\boldsymbol{f}$. For our *$\boldsymbol{\tau}$-method* we use this operator for computing a sequence of approximants to the least fixed point:

Definition 2.17 ($\boldsymbol{\tau}$-sequence)**.** *We call the sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)})_{k \in \mathbb{N}}$ of approximants defined by $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} := \mathcal{N}_{\boldsymbol{f}}^k(\boldsymbol{0})$ the* $\boldsymbol{\tau}$-sequence *for $\boldsymbol{f}$. We drop the subscript if $\boldsymbol{f}$ is understood.*

The $\boldsymbol{\tau}$-sequence converges to $\boldsymbol{\mu}$ at least as fast as the Kleene sequence:

Proposition 2.18. *Let $\boldsymbol{f}$ be a feasible min-max-SPP. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ is monotonically increasing, bounded from above by $\boldsymbol{\mu}$, and converges to $\boldsymbol{\mu}$. Moreover, $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\tau}^{(k)}$ holds for all $k \in \mathbb{N}$.*

Proof. By Lemma 2.16.1 we have $\boldsymbol{x} \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$ for every $\boldsymbol{x}$, so $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)})$ is monotonically increasing. We show by induction on k that $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\tau}^{(k)} \leq \boldsymbol{\mu}$. The base case $k = 0$ is trivial. For the step we get by Lemma 2.16:

$$\boldsymbol{\kappa}^{(k+1)} = \boldsymbol{f}(\boldsymbol{\kappa}^{(k)}) \leq \boldsymbol{f}(\boldsymbol{\tau}^{(k)}) \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\tau}^{(k)}) = \boldsymbol{\tau}^{(k+1)} \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\mu}) = \boldsymbol{\mu}$$

Thus we get $\boldsymbol{\mu} = \lim_{k\to\infty} \boldsymbol{\kappa}^{(k)} \leq \lim_{k\to\infty} \boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} \leq \boldsymbol{\mu}$ by Theorem 2.5. $\square$

We aim at some more quantitative results on the convergence speed of the $\boldsymbol{\tau}$-sequence. In fact, we will show that some of our results on Newton's method for SPPs from the first part of this thesis can be extended to min-max-SPPs.

Let $\boldsymbol{f}$ be a quadratic SPP (without $\wedge$ or $\vee$). Then the $\boldsymbol{\tau}$-sequence for $\boldsymbol{f}$ coincides with the Newton sequence and thus converges linearly[2]:

[2] In Chapter 1 we required that the SPPs under consideration be *clean*. This restriction was necessary because the Newton sequence was defined using matrix inverses (cf. Definition 1.11 on page 21) which in general only exist if the SPP is clean. In this section we avoid this problem by defining $\boldsymbol{\tau}^{(k+1)}$ as the least fixed point of $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{\tau}^{(k)}) \vee \boldsymbol{\tau}^{(k)}$. As mentioned in § 1.1.2, an SPP $\boldsymbol{f}$ can easily be transformed into an equivalent clean SPP $\boldsymbol{g}$ by detecting and removing the components i with $\mu_i = 0$. Then the approximants $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}$ and $\boldsymbol{\nu}_{\boldsymbol{g}}^{(k)}$ are equal, except for extra components in $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}$ which are 0. So, components i with $\mu_i = 0$ do not cause any harm.

Proposition 2.19 (see Theorem 1.51)**.** *Let $\boldsymbol{f}$ be a feasible quadratic SPP. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ has linear convergence order. More precisely, let β be the convergence order of the $\boldsymbol{\tau}$-sequence. Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\beta(k_{\boldsymbol{f}} + i \cdot n \cdot 2^n) \geq i$ for all $i \in \mathbb{N}$.*

Our goal for the rest of this section is to show that essentially the same holds for min-max-SPPs.

In a first step towards that goal, we consider max-SPPs.

Lemma 2.20. *Let $\boldsymbol{f}$ be a feasible max-SPP. Let $M := \{\sigma \in \Sigma \mid \boldsymbol{\mu f}^\sigma = \boldsymbol{\mu}\}$. The set M is non-empty and $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(k)}$ for all $\sigma \in M$ and $k \in \mathbb{N}$.*

Proof. Theorem 2.10 implies that there exists a $\vee$-strategy $\sigma \in \Sigma$ such that $\boldsymbol{\mu f}^\sigma = \boldsymbol{\mu}$. Thus M is non-empty. Let $\sigma \in M$. By induction on k Lemma 2.16 implies

$$\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} = \mathcal{N}_{\boldsymbol{f}}^k(\mathbf{0}) \geq \mathcal{N}_{\boldsymbol{f}^\sigma}^k(\mathbf{0}) = \boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(k)} \quad \text{for every } k \in \mathbb{N}. \qquad \square$$

A direct consequence of Lemma 2.20 is that the $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)})$ has exponential convergence order whenever there is a $\sigma \in M$ such that $(\boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(k)})$ has exponential convergence order. This is the case if $I - (\boldsymbol{f}^\sigma)'(\boldsymbol{\mu})$ is nonsingular, see Theorem 1.24. The following proposition holds even if nonsingularity cannot be guaranteed:

Proposition 2.21. *Let $\boldsymbol{f}$ be a feasible quadratic max-SPP. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ has linear convergence order. More precisely, let β be the convergence order of the $\boldsymbol{\tau}$-sequence. Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\beta(k_{\boldsymbol{f}} + i \cdot n \cdot 2^n) \geq i$ for all $i \in \mathbb{N}$.*

Proof. By Lemma 2.20 we have $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(k)}$ for all $k \in \mathbb{N}$. By Proposition 2.19, $\boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(k_{\boldsymbol{f}^\sigma} + i \cdot n \cdot 2^n)}$ has i valid bits. So we can choose $k_{\boldsymbol{f}} := k_{\boldsymbol{f}^\sigma}$. $\square$

The following lemma extends our considerations to min-max-SPPs $\boldsymbol{f}$. It relates the sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)})$ to the sequences $(\boldsymbol{\tau}_{\boldsymbol{f}^\pi}^{(k)})$ where $\boldsymbol{\mu f}^\pi = \boldsymbol{\mu}$.

Lemma 2.22. *Let $\boldsymbol{f}$ be a feasible min-max-SPP and m denote the number of strategies $\pi \in \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu f}^\pi$. There is a constant $\overline{k} \in \mathbb{N}$ such that for all $k \in \mathbb{N}$ there exists some strategy $\pi \in \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu f}^\pi$ and $\boldsymbol{\tau}_{\boldsymbol{f}}^{(\overline{k} + m \cdot k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^\pi}^{(k)}$.*

Proof. Using Lemma 2.16.7, we conclude that, for every k, there exists a $\wedge$-strategy $\pi^{(k)}$ for $\boldsymbol{f}$ such that $\mathcal{N}_{\boldsymbol{f}^{\pi^{(k)}}}(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}) = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}) = \boldsymbol{\tau}_{\boldsymbol{f}}^{(k+1)}$. We first show that there exists some $\overline{k} \in \mathbb{N}$ such that

$$\boldsymbol{\mu f}^{\pi^{(k)}} = \boldsymbol{\mu} \text{ for every } k \geq \overline{k}\,. \tag{2.8}$$

Since, by Lemma 2.12.2, $\boldsymbol{\mu f}^{\pi^{(k)}} \geq \boldsymbol{\mu}$ for every k, suppose for a contradiction that $\boldsymbol{\mu f}^{\pi^{(k)}} > \boldsymbol{\mu}$ for infinitely many k. As $\boldsymbol{\mu} \leq \boldsymbol{f}^{\pi^{(k)}}(\boldsymbol{\mu})$, we have $\boldsymbol{\mu} < \boldsymbol{f}^{\pi^{(k)}}(\boldsymbol{\mu})$ for infinitely many k. Since all $\boldsymbol{f}^{\pi^{(k)}}$ (finitely many) are continuous and $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}$ converges to $\boldsymbol{\mu}$, we conclude

$$\boldsymbol{\mu} \not\geq \boldsymbol{f}^{\pi^{(k)}}(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}) \leq \mathcal{N}_{\boldsymbol{f}^{\pi^{(k)}}}(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}) = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)}) = \boldsymbol{\tau}_{\boldsymbol{f}}^{(k+1)} \quad \text{for infinitely many } k\,,$$

contradicting $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} \leq \boldsymbol{\mu}$ for all k. So, (2.8) is shown.

Let $k \in \mathbb{N}$. Consider the $\wedge$-strategies $\pi^{(\overline{k})}, \ldots, \pi^{(\overline{k} + m \cdot k)}$. By (2.8), $\boldsymbol{\mu f}^\pi = \boldsymbol{\mu}$ holds for every $\wedge$-strategy π within this sequence. By the pigeonhole principle there is a strategy $\pi \in \Pi$ that occurs at least k times in $\pi^{(\overline{k})}, \ldots, \pi^{(\overline{k} + m \cdot k)}$. By monotonicity, $\boldsymbol{\tau}_{\boldsymbol{f}}^{(\overline{k} + m \cdot k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^\pi}^{(k)}$. $\square$

Now we can prove the main result of this section which states that the $\boldsymbol{\tau}$-sequence for min-max-SPPs has at least linear convergence order.

Theorem 2.23. *Let $\boldsymbol{f}$ be a feasible quadratic min-max-SPP and let m denote the number of strategies $\pi \in \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu f}^{\pi}$. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ has linear convergence order. More precisely, let β be the convergence order of the $\boldsymbol{\tau}$-sequence. Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\beta(k_{\boldsymbol{f}} + i \cdot m \cdot n \cdot 2^n) \geq i$ for all $i \in \mathbb{N}$.*

Proof. By Lemma 2.22 there exists some $\overline{k} \in \mathbb{N}$ such that for all $k \in \mathbb{N}$ there exists some strategy $\pi \in \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu f}^{\pi}$ and $\boldsymbol{\tau}_{\boldsymbol{f}}^{(\overline{k}+m\cdot k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^{\pi}}^{(k)}$. Let

$$k_{max} := \max \left\{ k_{\boldsymbol{f}^{\pi}} \mid \pi \in \Pi,\ \boldsymbol{\mu f}^{\pi} = \boldsymbol{\mu} \right\}$$

where $k_{\boldsymbol{f}^{\pi}}$ is from Proposition 2.21. Let $k_{\boldsymbol{f}} := \overline{k} + m \cdot k_{max}$. Let $i \in \mathbb{N}$ and $k := k_{max} + i \cdot n \cdot 2^n$. Then:

$$\boldsymbol{\tau}_{\boldsymbol{f}}^{(k_{\boldsymbol{f}}+i\cdot m\cdot n\cdot 2^n)} = \boldsymbol{\tau}_{\boldsymbol{f}}^{(\overline{k}+m\cdot k_{max}+i\cdot m\cdot n\cdot 2^n)} = \boldsymbol{\tau}_{\boldsymbol{f}}^{(\overline{k}+m\cdot k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^{\pi}}^{(k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^{\pi}}^{(k_{\boldsymbol{f}^{\pi}}+i\cdot n\cdot 2^n)}$$

By Proposition 2.21, $\boldsymbol{\tau}_{\boldsymbol{f}^{\pi}}^{(k_{\boldsymbol{f}^{\pi}}+i\cdot n\cdot 2^n)}$ has i valid bits. Hence $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k_{\boldsymbol{f}}+i\cdot m\cdot n\cdot 2^n)}$ has i valid bits as well. □

The upper bound on the convergence rate provided by Theorem 2.23 is by the factor m worse than the upper bound obtained for SPPs and max-SPPs, cf. Proposition 2.19 and Proposition 2.21. As m is the number of strategies $\pi \in \Pi$ with $\boldsymbol{\mu f}^{\pi} = \boldsymbol{\mu}$, this number is trivially bounded by $|\Pi|$ but should usually be much smaller.

In order to determine the approximant $\boldsymbol{\tau}^{(k+1)} = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\tau}^{(k)})$ from $\boldsymbol{\tau}^{(k)}$ we have to compute the least fixed point of the min-max-SML $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{\tau}^{(k)}\right) \vee \boldsymbol{\tau}^{(k)}$. This can be done using the strategy improvement algorithm from [GS07b]. The algorithm iterates over $\vee$-strategies. For each strategy it solves a linear program or, alternatively, iterates over $\wedge$-strategies. For more details see [GS07b]. The number of $\vee$-strategies used by this algorithm is trivially bounded by the number of $\vee$-strategies for $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{\tau}^{(k)}\right) \vee \boldsymbol{\tau}^{(k)}$, which is exponential in the number of $\vee$-expressions occurring in $\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \boldsymbol{\tau}^{(k)}\right) \vee \boldsymbol{\tau}^{(k)}$. So far, no example is known for which the algorithm needs more than linearly many strategy improvement steps, i.e., iterates over more than linearly many strategies. However, a very recent result [Fri09] indicates that, in fact, exponentially many iterations may be needed. This is shown in [Fri09] for the strategy improvement algorithm on parity games. Whether this carries over to min-max-SMLs remains to be seen.

2.4 The $\boldsymbol{\nu}$-Method

In this section we derive an alternative method to approximate $\boldsymbol{\mu f}$ for min-max-SPPs $\boldsymbol{f}$. This method, called *$\boldsymbol{\nu}$-method*, has both advantages and disadvantages compared to the $\boldsymbol{\tau}$-method presented in the previous section.

Advantage: One step of the $\boldsymbol{\nu}$-method is cheaper to compute. The $\boldsymbol{\tau}$-method uses strategy iteration over $\vee$-strategies to compute $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$. This could be expensive, as there may be exponentially many $\vee$-strategies. The $\boldsymbol{\nu}$-method, which we present in this section, is an alternative generalization of Newton's method. In each step it picks the currently most promising $\vee$-strategy directly, without strategy iteration. It turns out that the computation of a single step reduces to solving one instance of a linear programming (LP) problem.

Disadvantage: The $\boldsymbol{\nu}$-method needs more steps. Letting $\boldsymbol{\nu}^{(k)}$ and $\boldsymbol{\tau}^{(k)}$ denote the iterates of the $\boldsymbol{\tau}$-method and the $\boldsymbol{\nu}$-method, respectively, we will show $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\tau}^{(k)} \leq \boldsymbol{\mu}$ holds for all $k \in \mathbb{N}$, where the inequalities may be strict. This means that whereas a single step of the $\boldsymbol{\nu}$-method is cheaper to compute, more steps may be needed to reach an approximation of $\boldsymbol{\mu}$ of a certain precision.

Advantage: The $\boldsymbol{\nu}$-method computes good strategies for extinction games. We will see at the end of the section that the iterates of the $\boldsymbol{\nu}$-method carry information on good strategies for extinction games. More precisely, with each iterate $\boldsymbol{\nu}^{(k)}$ comes a strategy for the terminator that guarantees her/him a termination probability of at least $\boldsymbol{\nu}^{(k)}$, regardless of how the savior plays.

For the $\boldsymbol{\nu}$-method, consider again a fixed feasible min-max-SPP $\boldsymbol{f}$ whose least fixed point we want to approximate. Assume that $\boldsymbol{y}$ is some approximation of $\boldsymbol{\mu}$. Instead of applying $\mathcal{N}_{\boldsymbol{f}}$ to $\boldsymbol{y}$, as in the $\boldsymbol{\tau}$-method, we now choose a strategy $\sigma \in \Sigma$ such that $\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{f}^{\sigma}(\boldsymbol{y})$, and compute $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y})$, where $\mathcal{N}_{\boldsymbol{f}^{\sigma}}$ was defined in § 2.3 as $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y}) := \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y}) \vee \boldsymbol{y})$. In the following we write $\mathcal{N}_{\sigma}$ instead of $\mathcal{N}_{\boldsymbol{f}^{\sigma}}$ if $\boldsymbol{f}$ is understood.

Assume for a moment that $\boldsymbol{f}$ is a max-SPP and that there is a unique $\sigma \in \Sigma$ such that $\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{f}^{\sigma}(\boldsymbol{y})$. The approximant $\mathcal{N}_{\sigma}(\boldsymbol{y})$ is the result of applying one iteration of Newton's method, because $\boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})$ is not only a linearization of $\boldsymbol{f}^{\sigma}$, but the first-order Taylor approximation of $\boldsymbol{f}$ at $\boldsymbol{y}$. More precisely, $\boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{y}) + \boldsymbol{f}'(\boldsymbol{y}) \cdot (\boldsymbol{x} - \boldsymbol{y})$, and $\mathcal{N}_{\sigma}(\boldsymbol{y})$ is obtained by solving $\boldsymbol{x} = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})(\boldsymbol{x})$. In this sense, the $\boldsymbol{\nu}$-method is a more direct generalization of Newton's method than the $\boldsymbol{\tau}$-method.

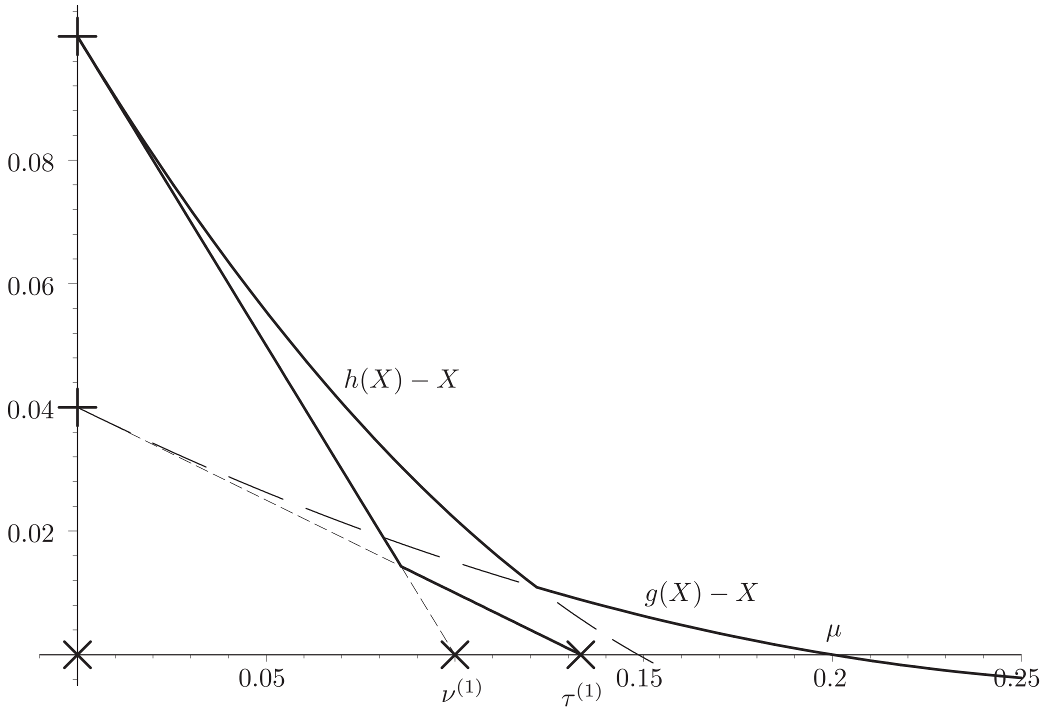

Figure 2.5: The $\boldsymbol{\tau}$-method and the $\boldsymbol{\nu}$-method produce different iterates.

Example 2.24. *Consider again the 1-dimensional max-SPP f from Example 0.11 with $f(X) = g(X) \vee h(X)$ where*

$$g(X) = 0.5X^2 + 0.7X + 0.04 \quad \textit{and} \quad h(X) = 0.1 + 2.2X^2\,,$$

see Figure 2.5. Let

$$\overline{g}(X) = g(0) + g'(0) \cdot X = 0.04 + 0.7X \quad \text{and} \quad \overline{h}(X) = h(0) + h'(0) \cdot X = 0.1\,.$$

The $\boldsymbol{\tau}$-method computes $\mathcal{N}_{\boldsymbol{f}}(0) = \tau^{(1)}$ as the least fixed point of $\overline{f}$ where $\overline{f}(X) = \overline{g}(X) \vee \overline{h}(X)$, i.e., $\tau^{(1)}$ is the solution of $\overline{g}(X) - X \ \vee\ \overline{h}(X) - X = 0$, see Figure 2.5. The $\boldsymbol{\nu}$-method proceeds as follows. First it picks the strategy σ with $f(0) = f^{\sigma}(0)$, i.e., it picks $\sigma = h$. Then it computes $\mathcal{N}_{\sigma}(0) = \nu^{(1)}$ as the least fixed point of

$$\mathcal{L}\left(f^{\sigma}, 0\right) = \mathcal{L}\left(h, 0\right) = \overline{h}\,,$$

i.e., $\nu^{(1)}$ is the solution of $\overline{h}(X) - X = 0$, see also Figure 2.5. Notice that $\overline{h}$ is the first-order Taylor approximation of f at 0, so the $\boldsymbol{\nu}$-method is very close to the "classical" Newton method from Chapter 1. But recall from Example 0.9 that the classical Newton method does not work when there are minimum operators. Figure 2.5 shows $\nu^{(1)} < \tau^{(1)}$ which illustrates the disadvantage of the $\boldsymbol{\nu}$-method mentioned at the beginning of the section.

Formally, we define the $\boldsymbol{\nu}$-method by a sequence of approximants, the $\boldsymbol{\nu}$*-sequence*.

Definition 2.25 ($\boldsymbol{\nu}$-sequence)**.** *A sequence $(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})_{k \in \mathbb{N}}$ is called* $\boldsymbol{\nu}$-sequence *of a min-max-SPP $\boldsymbol{f}$ if $\boldsymbol{\nu}_{\boldsymbol{f}}^{(0)} = \mathbf{0}$ and for each k there is a strategy $\sigma^{(k)} \in \Sigma$ with $\boldsymbol{f}(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)}) = \boldsymbol{f}^{\sigma^{(k)}}(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})$ and $\boldsymbol{\nu}_{\boldsymbol{f}}^{(k+1)} = \mathcal{N}_{\sigma^{(k)}}(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})$. We drop the subscript if $\boldsymbol{f}$ is understood.*

Notice the nondeterminism here if there is more than one $\vee$-strategy that attains $\boldsymbol{f}(\boldsymbol{\nu}^{(k)})$. The following proposition is analogous to Proposition 2.18 (also cf. Theorem 1.12) and states some basic properties of $\boldsymbol{\nu}$-sequences.

Proposition 2.26. *Let $\boldsymbol{f}$ be a feasible min-max-SPP. The sequence $(\boldsymbol{\nu}^{(k)})$ is monotonically increasing, bounded from above by $\boldsymbol{\mu}$, and converges to $\boldsymbol{\mu}$. More precisely, we have $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{\nu}^{(k+1)} \leq \boldsymbol{\mu}$ for all $k \in \mathbb{N}$.*

Proof. By induction on k. The base case $k = 0$ is easy. Let $k > 0$. Then:

$$\begin{aligned}
\boldsymbol{\kappa}^{(k)} &= \boldsymbol{f}(\boldsymbol{\kappa}^{(k-1)}) && \text{(definition of } \boldsymbol{\kappa}^{(k)}\text{)} \\
&\leq \boldsymbol{f}(\boldsymbol{\nu}^{(k-1)}) && \text{(induction hyp.: } \boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\nu}^{(k)}\text{)} \\
&= \boldsymbol{f}^{\sigma^{(k-1)}}(\boldsymbol{\nu}^{(k-1)}) && \text{(definition of } \sigma^{(k-1)}\text{)} \\
&\leq \boldsymbol{\nu}^{(k)} && \text{(Lemma 2.16.1)} \\
&\leq \boldsymbol{f}^{\sigma^{(k-1)}}(\boldsymbol{\nu}^{(k)}) && \begin{aligned}&\text{(induction hyp.: } \boldsymbol{\nu}^{(k-1)} \leq \boldsymbol{f}^{\sigma^{(k-1)}}(\boldsymbol{\nu}^{(k-1)}), \\ &\text{ so with Lemma 2.16.4: } \boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}^{\sigma^{(k-1)}}(\boldsymbol{\nu}^{(k)})\text{)}\end{aligned} \\
&\leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) && \text{(}\sigma^{(k-1)} \text{ is a } \vee\text{-strategy)} \\
&= \boldsymbol{f}^{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)}) && \text{(definition of } \sigma^{(k)}\text{)} \\
&\leq \boldsymbol{\nu}^{(k+1)} && \text{(Lemma 2.16.1)} \\
&= \mathcal{N}_{\boldsymbol{f}^{\sigma^{(k)}}}(\boldsymbol{\nu}^{(k)}) && \text{(definition of } \boldsymbol{\nu}^{(k+1)}\text{)} \\
&\leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\nu}^{(k)}) && \text{(Lemma 2.16.5)} \\
&\leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\mu}) && \begin{aligned}&\text{(induction hyp.: } \boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu}, \\ &\text{ so with Lemma 2.16.3: } \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\nu}^{(k)}) \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\mu})\text{)}\end{aligned} \\
&= \boldsymbol{\mu} && \text{(Lemma 2.16.2)}
\end{aligned}$$

□

The goal of this section is again to strengthen Proposition 2.26 towards quantitative convergence results for $\boldsymbol{\nu}$-sequences. To achieve this goal we again relate the convergence of $\boldsymbol{\nu}$-sequences to the convergence of Newton's method for SPPs. If $\boldsymbol{f}$ is an SPP, Proposition 2.19 allows to reason about the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ when applied to approximants $\boldsymbol{x} \le \boldsymbol{\mu}$. To transfer this result to min-max-SPPs $\boldsymbol{f}$ we need an invariant like $\boldsymbol{\nu}^{(k)} \le \boldsymbol{\mu f}^{\sigma^{(k)}}$ for $\boldsymbol{\nu}$-sequences. To obtain such an invariant we need to further restrict the choice of $\sigma^{(k)}$. Roughly speaking, the strategy in a component i is only changed when it is immediate that component i has not yet reached its fixed point.

Definition 2.27 (lazy strategy update). *Let $\boldsymbol{x} \le \boldsymbol{f}^{\sigma}(\boldsymbol{x})$ for a $\sigma \in \Sigma$. We say that $\sigma' \in \Sigma$ is obtained from $\boldsymbol{x}$ and σ by a* lazy strategy update *if $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}^{\sigma'}(\boldsymbol{x})$ and $\sigma'(i) = \sigma(i)$ holds for all components i with $f_i(\boldsymbol{x}) = x_i$. We call a $\boldsymbol{\nu}$-sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$* lazy *if for all k, the strategy $\sigma^{(k)}$ is obtained from $\boldsymbol{\nu}^{(k)}$ and $\sigma^{(k-1)}$ by a lazy strategy update.*

Algorithm 2.1 summarizes the *lazy $\boldsymbol{\nu}$-method* which works by computing lazy $\boldsymbol{\nu}$-sequences.

Algorithm 2.1 lazy $\boldsymbol{\nu}$-method

```
procedure lazy-$\boldsymbol{\nu}(\boldsymbol{f}, k)$
assumes: $\boldsymbol{f}$ is a min-max-SPP
returns: $\boldsymbol{\nu}^{(k)}, \sigma^{(k)}$ obtained by $k$ iterations of the lazy $\boldsymbol{\nu}$-method
    $\boldsymbol{\nu} \leftarrow \mathbf{0}$
    $\sigma \leftarrow$ any $\sigma \in \Sigma$ such that $\boldsymbol{f}(\mathbf{0}) = \boldsymbol{f}^{\sigma}(\mathbf{0})$
    for $i$ from 1 to $k$ do
        $\boldsymbol{\nu} \leftarrow \mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{\nu})$
        $\sigma \leftarrow$ lazy strategy update from $\boldsymbol{\nu}$ and $\sigma$
    od
    return $\boldsymbol{\nu}, \sigma$
```

For our convergence speed analysis of Algorithm 2.1, the following invariant will be crucial:

Lemma 2.28. *Let $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ be a lazy $\boldsymbol{\nu}$-sequence. Then $\boldsymbol{\nu}^{(k)} \le \boldsymbol{\mu f}^{\sigma^{(k)}\pi}$ holds for all $k \in \mathbb{N}$ and for all $\pi \in \Pi^*$ (i.e., for all π such that $\boldsymbol{f}^{\pi}$ is feasible).*

The proof of Lemma 2.28 is non-trivial and, for the sake of readability, has been moved to Appendix B.1. The following example shows that lazy strategy updates are essential to Lemma 2.28 even for max-SPPs.

Example 2.29. *Consider the max-SPP $\boldsymbol{f}$ with*

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} f_1(X_1, X_2) \\ f_2(X_1, X_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \vee X_1 \\ X_1 X_2 + \frac{1}{2} \end{pmatrix} .$$

Let $\sigma^{(0)}(1) = \frac{1}{2}$ and $\sigma^{(1)}(1) = X_1$. Then there is a $\boldsymbol{\nu}$-sequence $(\boldsymbol{\nu}^{(k)})$ with

$$\boldsymbol{\nu}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\nu}^{(1)} = \mathcal{N}_{\sigma^{(0)}}(\boldsymbol{\nu}^{(0)}) = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \quad \boldsymbol{\nu}^{(2)} = \mathcal{N}_{\sigma^{(1)}}(\boldsymbol{\nu}^{(1)}) .$$

However, the conclusion of Lemma 2.28 does not hold, because

$$\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \boldsymbol{\nu}^{(1)} \not\le \boldsymbol{\mu f}^{\sigma^{(1)}} = \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} .$$

Notice that $\sigma^{(1)}$ is not obtained by a lazy strategy update, as $f_1(\boldsymbol{\nu}^{(1)}) = \nu_1^{(1)}$.

Lemma 2.28 falls short of our subgoal to establish $\boldsymbol{\nu}^{(k)} \le \boldsymbol{\mu f}^{\sigma^{(k)}}$, because $\Pi \setminus \Pi^*$ might be non-empty. In fact, the following example shows that $\boldsymbol{\nu}^{(k)} \le \boldsymbol{\mu f}^{\sigma^{(k)}\pi}$ does not always hold for all $\pi \in \Pi$, even when $\boldsymbol{f}^{\sigma^{(k)}\pi}$ is feasible. Luckily, Lemma 2.28 will suffice for our convergence speed result.

Example 2.30. *Consider again the min-max-SPP $\boldsymbol{f}$ from Example 2.4 with*

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} f_1(X_1, X_2) \\ f_2(X_1, X_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}X_2^2 + \frac{1}{2} \ \wedge \ 3 \\ X_1 \ \vee \ 2 \end{pmatrix} .$$

The unique $\boldsymbol{\nu}$-sequence of $\boldsymbol{f}$ is given by

$$\begin{aligned} &\boldsymbol{\nu}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, && \sigma^{(0)}(2) = 2, && \boldsymbol{\nu}^{(1)} = \begin{pmatrix} 1/2 \\ 2 \end{pmatrix}, && \sigma^{(1)}(2) = 2, \\ &\boldsymbol{\nu}^{(2)} = \begin{pmatrix} 5/2 \\ 2 \end{pmatrix}, && \sigma^{(2)}(2) = X_1, && \boldsymbol{\nu}^{(i)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}, && \sigma^{(i)}(2) = X_1 \qquad (i \ge 3) . \end{aligned}$$

We have (see also Figure 2.1 on page 55)

$$\boldsymbol{\nu}^{(3)} = \boldsymbol{\mu f}^{\sigma^{(3)}\pi} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \quad \text{for } \pi \in \Pi^* \text{ with } \pi(1) = 3, \text{ but}$$

$$\boldsymbol{\nu}^{(3)} \not\le \boldsymbol{\mu f}^{\sigma^{(3)}\pi'} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{for } \pi' \in \Pi \setminus \Pi^* \text{ with } \pi'(1) = \frac{1}{2}X_2^2 + \frac{1}{2}.$$

Note that $\boldsymbol{f}^{\sigma^{(3)}\pi'}$ is feasible and $\boldsymbol{f}^{\pi'}$ is not.

The following lemma relates the $\boldsymbol{\nu}$-method for min-max-SPPs to Newton's method for SPPs.

Lemma 2.31. *Let $\boldsymbol{f}$ be a feasible min-max-SPP and $(\boldsymbol{\nu}^{(k)})$ a lazy $\boldsymbol{\nu}$-sequence. Let m be the number of strategy pairs $(\sigma, \pi) \in \Sigma \times \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu f}^{\sigma\pi}$. Then $m \ge 1$ and there is a constant $\overline{k} \in \mathbb{N}$ such that, for all $k \in \mathbb{N}$, there exist strategies $\sigma \in \Sigma$, $\pi \in \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu f}^{\sigma\pi}$ and $\boldsymbol{\nu}_{\boldsymbol{f}}^{(\overline{k}+m\cdot k)} \ge \boldsymbol{\tau}_{\boldsymbol{f}^{\sigma\pi}}^{(k)}$.*

Before proving Lemma 2.31 we show the following lemma, a consequence of Lemma 2.28. Let us define, for all $k \in \mathbb{N}$, a strategy $\pi^{(k)} \in \Pi$ such that $\boldsymbol{\nu}^{(k+1)} = \mathcal{N}_{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)}) = \mathcal{N}_{\sigma^{(k)}\pi^{(k)}}(\boldsymbol{\nu}^{(k)})$. Such a $\pi^{(k)}$ exists by Lemma 2.16.7.

Lemma 2.32. *There is a $\overline{k} \in \mathbb{N}$ such that $\boldsymbol{\mu} = \boldsymbol{\mu f}^{\sigma^{(k)}\pi^{(k)}}$ for all $k \ge \overline{k}$.*

Proof. As the $\pi^{(k)}$ are $\wedge$-strategies, we have $\boldsymbol{f} \le \boldsymbol{f}^{\pi^{(k)}}$. Hence $\boldsymbol{\mu} \le \boldsymbol{\mu f}^{\pi^{(k)}}$ if $\boldsymbol{\mu f}^{\pi^{(k)}}$ is defined. We show that there is a $k_0 \ge 0$ such that

$$\boldsymbol{\mu} = \boldsymbol{\mu f}^{\pi^{(k)}} \text{ for all } k \ge k_0 . \tag{2.9}$$

Assume for a contradiction that $\boldsymbol{f}^{\pi^{(k)}}$ is infeasible or $\boldsymbol{\mu} < \boldsymbol{\mu f}^{\pi^{(k)}}$ holds for infinitely many k. Then $\boldsymbol{\mu} < \boldsymbol{f}^{\pi^{(k)}}(\boldsymbol{\mu})$ for infinitely many k. By Proposition 2.26, the $\boldsymbol{\nu}^{(k)}$ converge to $\boldsymbol{\mu}$, so, by continuity, we also have $\boldsymbol{\mu} \not\ge \boldsymbol{f}^{\pi^{(k)}}(\boldsymbol{\nu}^{(k)})$ for some (actually infinitely many) k. With Lemma 2.16.1 we get $\boldsymbol{\mu} \not\ge \boldsymbol{f}^{\pi^{(k)}}(\boldsymbol{\nu}^{(k)}) \le \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) = \boldsymbol{f}^{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)}) \le \mathcal{N}_{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)}) = \boldsymbol{\nu}^{(k+1)}$ which contradicts Proposition 2.26 stating $\boldsymbol{\nu}^{(k+1)} \le \boldsymbol{\mu}$. So, (2.9) holds in fact, which implies $\pi^{(k)} \in \Pi^*$, i.e., $\boldsymbol{f}^{\pi^{(k)}}$ is feasible for all $k \ge k_0$. Therefore, Lemma 2.28 implies that

$$\boldsymbol{\nu}^{(k)} \le \boldsymbol{\mu f}^{\sigma^{(k)}\pi^{(k)}} \le \boldsymbol{\mu f}^{\pi^{(k)}} = \boldsymbol{\mu} \text{ holds for all } k \ge k_0. \tag{2.10}$$

Assume for a contradiction that $\boldsymbol{\mu}\boldsymbol{f}^{\sigma^{(k)}\pi^{(k)}} < \boldsymbol{\mu}$ holds for infinitely many k. There are only finitely many different strategy pairs $(\sigma^{(k)}, \pi^{(k)})$, so there is an $\varepsilon > 0$ such that $\left\| \boldsymbol{\mu} - \boldsymbol{\mu}\boldsymbol{f}^{\sigma^{(k)}\pi^{(k)}} \right\| \geq \varepsilon$ holds for infinitely many k. With (2.10), $\left\| \boldsymbol{\mu} - \boldsymbol{\nu}^{(k)} \right\| \geq \varepsilon$ holds for infinitely many k, contradicting Proposition 2.26 which assures that the $\boldsymbol{\nu}^{(k)}$ converge to $\boldsymbol{\mu}$. □

Now we can show Lemma 2.31.

Proof of Lemma 2.31. Lemma 2.32 implies $m \geq 1$. Take the $\overline{k}$ from Lemma 2.32 and consider $\boldsymbol{\nu}_{\boldsymbol{f}}^{(\overline{k}+m\cdot k)}$ for any $k \in \mathbb{N}$. It is obtained by applying $m \cdot k$ iterations of the lazy $\boldsymbol{\nu}$-method to $\boldsymbol{\nu}^{(\overline{k})}$. By the pigeonhole principle, there are strategies $\sigma \in \Sigma$, $\pi \in \Pi$ such that $\mathcal{N}_{\boldsymbol{f}^{\sigma\pi}}$ has been applied at least k times. Hence, we have:

$$
\begin{aligned}
\boldsymbol{\mu}\boldsymbol{f}^{\sigma\pi} &= \boldsymbol{\mu} && \text{(Lemma 2.32)} \\
&\geq \boldsymbol{\nu}_{\boldsymbol{f}}^{(\overline{k}+m\cdot k)} && \text{(Proposition 2.26)} \\
&\geq \left(\mathcal{N}_{\boldsymbol{f}^{\sigma\pi}}\right)^k \left(\boldsymbol{\nu}_{\boldsymbol{f}}^{(\overline{k})}\right) && \begin{array}{l}\text{(pigeonhole principle (see above),} \\ \text{ Lemma 2.16.3: monotonicity of } \mathcal{N}_{\boldsymbol{f}^{\sigma\pi}})\end{array} \\
&\geq \left(\mathcal{N}_{\boldsymbol{f}^{\sigma\pi}}\right)^k (\mathbf{0}) && \text{(Lemma 2.16.3: monotonicity of } \mathcal{N}_{\boldsymbol{f}^{\sigma\pi}}) \\
&= \boldsymbol{\tau}_{\boldsymbol{f}^{\sigma\pi}}^{(k)} && \text{(definition of } \boldsymbol{\tau}_{\boldsymbol{f}^{\sigma\pi}}^{(k)})
\end{aligned}
$$

□

Again, in typical cases, i.e., if $I - (\boldsymbol{f}^{\sigma\pi})'(\boldsymbol{\mu})$ is nonsingular for all $\sigma \in \Sigma$ and $\pi \in \Pi$ with $\boldsymbol{\mu}\boldsymbol{f}^{\sigma\pi} = \boldsymbol{\mu}$, the lazy $\boldsymbol{\nu}$-sequence has exponential convergence order. The following theorem captures the worst-case, in which it still converges linearly.

Theorem 2.33. *Let $\boldsymbol{f}$ be a feasible quadratic min-max-SPP and let m be the number of strategy pairs $(\sigma, \pi) \in \Sigma \times \Pi$ with $\boldsymbol{\mu} = \boldsymbol{\mu}\boldsymbol{f}^{\sigma\pi}$. The lazy $\boldsymbol{\nu}$-sequence $(\boldsymbol{\nu}^{(k)})$ has linear convergence order. More precisely, let β be the convergence order of the lazy $\boldsymbol{\nu}$-sequence. Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\beta(k_{\boldsymbol{f}} + i \cdot m \cdot n \cdot 2^n) \geq i$ for all $i \in \mathbb{N}$.*

Proof. Set $k_{max} = \max\{k_{\boldsymbol{f}^{\sigma\pi}} \mid \boldsymbol{\mu} = \boldsymbol{\mu}\boldsymbol{f}^{\sigma\pi}\}$, where the maximum ranges over the $k_{\boldsymbol{f}^{\sigma\pi}}$ from Proposition 2.19. Let $i \in \mathbb{N}$. We have:

$$
\begin{aligned}
\boldsymbol{\nu}_{\boldsymbol{f}}^{(\overline{k}+m\cdot(k_{max}+i\cdot n\cdot 2^n))} &\geq \boldsymbol{\tau}_{\boldsymbol{f}^{\sigma\pi}}^{(k_{max}+i\cdot n\cdot 2^n)} && \text{(Lemma 2.31)} \\
&\geq \boldsymbol{\tau}_{\boldsymbol{f}^{\sigma\pi}}^{(k_{\boldsymbol{f}^{\sigma\pi}}+i\cdot n\cdot 2^n)} && (k_{max} \geq k_{\boldsymbol{f}^{\sigma\pi}})
\end{aligned}
$$

The last approximant has, by Proposition 2.19, i valid bits of $\boldsymbol{\mu}\boldsymbol{f}^{\sigma\pi} = \boldsymbol{\mu}$. So we can choose $k_{\boldsymbol{f}} := \overline{k} + m \cdot k_{max}$. □

Algorithm 2.2 shows how to compute $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y})$ by solving a single linear programming (LP) problem.

Example 2.34. *We illustrate Algorithm 2.2 using the min-max-SPP $\boldsymbol{f}$ from Example 2.4 with*

$$
\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} f_1(X_1, X_2) \\ f_2(X_1, X_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}X_2^2 + \frac{1}{2} \ \wedge \ 3 \\ X_1 \ \vee \ 2 \end{pmatrix} .
$$

As in Example 2.15 (page 60), let $\boldsymbol{v} := (1/2,\, 1/2)^\top$. The strategy σ with $\sigma(2) = 2$ yields

$$
\boldsymbol{f}^{\sigma}(\boldsymbol{v}) = \boldsymbol{f}(\boldsymbol{v}) = \begin{pmatrix} 5/8 \\ 2 \end{pmatrix} ,
$$

Algorithm 2.2 $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$

```
procedure $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$
assumes: $\boldsymbol{f}$ is a min-SPP, $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$
returns: $\boldsymbol{\mu}(\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y}) \vee \boldsymbol{y})$
    $\boldsymbol{g} \leftarrow$ linear min-SPP with $\boldsymbol{g}(\boldsymbol{X}) = \boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{X}) - \boldsymbol{y}$
    $\boldsymbol{u} \leftarrow \boldsymbol{\kappa}_{\boldsymbol{g}}^{(n)}$
    $\widetilde{\boldsymbol{g}} \leftarrow (\widetilde{g}_1, \ldots, \widetilde{g}_n)^\top$ where $\widetilde{g}_i = \begin{cases} 0 & \text{if } u_i = 0 \\ g_i & \text{if } u_i > 0 \end{cases}$
    $\boldsymbol{d} \leftarrow$ maximize $x_1 + \cdots + x_n$ subject to $\boldsymbol{0} \leq \boldsymbol{x} \leq \widetilde{\boldsymbol{g}}(\boldsymbol{x})$
    return $\boldsymbol{y} + \boldsymbol{d}$
```

so, if $\boldsymbol{v}$ were an iterate of the $\boldsymbol{\nu}$-sequence, Algorithm 2.1 would call Algorithm 2.2 with $\mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{v})$. Algorithm 2.2 first computes the linearization $\boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{v})$:

$$\boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{v})(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{2}X_2 + \frac{3}{8} \ \wedge \ 3 \\ 2 \end{pmatrix} .$$

Then the function $\boldsymbol{g}$ in Algorithm 2.2 is computed as follows:

$$\boldsymbol{g}(\boldsymbol{X}) = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{v})(\boldsymbol{v} + \boldsymbol{X}) - \boldsymbol{v} = \begin{pmatrix} \frac{1}{2} \cdot (\frac{1}{2} + X_2) + \frac{3}{8} - \frac{1}{2} \ \wedge \ 3 - \frac{1}{2} \\ 2 - \frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}X_2 + \frac{1}{8} \ \wedge \ \frac{5}{2} \\ \frac{3}{2} \end{pmatrix}$$

Since $\boldsymbol{\kappa}_{\boldsymbol{g}}^{(2)} \geq \boldsymbol{\kappa}_{\boldsymbol{g}}^{(1)} = (1/8,\ 3/2)^\top$, we have $\widetilde{\boldsymbol{g}} = \boldsymbol{g}$. As the next step, Algorithm 2.2 solves the following linear programming problem:

$$\text{maximize} \quad x_1 + x_2 \quad \text{subject to} \begin{cases} x_1 \geq 0, & x_1 \leq \frac{1}{2}x_2 + \frac{1}{8}, \quad x_1 \leq \frac{5}{2} \\ x_2 \geq 0, & x_2 \leq \frac{3}{2} \end{cases}$$

Its solution is $\boldsymbol{d} = (7/8,\ 3/2)^\top$, so Algorithm 2.2 returns $\mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{v}) = \boldsymbol{v} + \boldsymbol{d} = (11/8,\ 2)^\top$. Note that, in this particular instance, the vector $\mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{v}) = (11/8,\ 2)^\top$ equals the vector $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{v})$ as computed in Example 2.15 (page 60).

For the correctness of Algorithm 2.2, we show the following proposition which states that $\mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{y})$ can be determined by computing the least fixed point of a certain linear min-SPP.

Proposition 2.35. *Let $\boldsymbol{y} \leq \boldsymbol{f}^\sigma(\boldsymbol{y}) \leq \boldsymbol{\mu}$. Then $\mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{y}) = \boldsymbol{y} + \boldsymbol{\mu g}$ for the linear min-SPP $\boldsymbol{g}$ with $\boldsymbol{g}(\boldsymbol{X}) = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{X}) - \boldsymbol{y}$.*

Proof. We have for all vectors $\boldsymbol{d}$:

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{d}) &:= \boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{d}) - \boldsymbol{y} && \text{(by definition of } \boldsymbol{g}\text{)} \\ &= \bigwedge_{\pi \in \Pi} \boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma\pi}, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{d}) - \boldsymbol{y} && \text{(by definition of } \boldsymbol{\mathcal{L}}(,)\text{)} \\ &= \bigwedge_{\pi \in \Pi} \boldsymbol{f}^{\sigma\pi}(\boldsymbol{y}) - \boldsymbol{y} + (\boldsymbol{f}^{\sigma\pi})'(\boldsymbol{y}) \cdot \boldsymbol{d} && \text{(by definition of } \boldsymbol{\mathcal{L}}(,)\text{)} \end{aligned}$$

Notice that $\boldsymbol{g}$ is a min-*SPP*, because $\boldsymbol{f}^{\sigma\pi}(\boldsymbol{y}) \geq \boldsymbol{y}$ by assumption. Let $\boldsymbol{x}^{(1)} = \mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{y}) = \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{y}) \vee \boldsymbol{y})$. As $\boldsymbol{y} \leq \boldsymbol{f}^\sigma(\boldsymbol{y})$, $\boldsymbol{x}^{(1)}$ is the least point $\boldsymbol{x} \geq \boldsymbol{y}$ such that $\boldsymbol{x} = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{y})(\boldsymbol{x})$. In other words, $\boldsymbol{x}^{(1)} = \boldsymbol{y} + \boldsymbol{d}^{(0)}$, where $\boldsymbol{d}^{(0)}$ is the least *nonnegative* point $\boldsymbol{d}$ such that $\boldsymbol{y} + \boldsymbol{d} = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^\sigma, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{d})$, which is equivalent to $\boldsymbol{d} = \boldsymbol{g}(\boldsymbol{d})$. So, $\boldsymbol{x}^{(1)} = \boldsymbol{y} + \boldsymbol{\mu g}$. □

After having computed the linear min-SPP $\boldsymbol{g}$, Algorithm 2.2 determines the 0-components of $\boldsymbol{\mu g}$. This can be done by performing n Kleene steps, since $(\boldsymbol{\mu g})_i = 0$ whenever $(\boldsymbol{\kappa}_{\boldsymbol{g}}^{(n)})_i = 0$ (cf. § 1.1.2). Let $\widetilde{\boldsymbol{g}}$ be the linear min-SPP obtained from $\boldsymbol{g}$ by substituting the constant 0 for all components g_i with $(\boldsymbol{\mu g})_i = 0$. The least fixed point of $\widetilde{\boldsymbol{g}}$ can be computed by solving a single linear programming (LP) problem, as implied by the following lemma.

Lemma 2.36. *Let $\boldsymbol{g}$ be a linear min-SPP such that $g_i = 0$ whenever $(\boldsymbol{\mu g})_i = 0$ for all components i. Then $\boldsymbol{\mu g}$ is the greatest vector $\boldsymbol{x}$ with $\boldsymbol{x} \leq \boldsymbol{g}(\boldsymbol{x})$.*

Proof. Let S denote the set of the components i with $g_i = 0$. Define the SPP $\boldsymbol{f} := \boldsymbol{g}[S/\boldsymbol{0}]$. Now we know by assumption that $\boldsymbol{f}$ is clean, i.e., $\boldsymbol{\mu} \succ \boldsymbol{0}$, and it suffices to show that $\boldsymbol{y} := \boldsymbol{\mu}$ is the greatest prefixed point of $\boldsymbol{f}$. By Lemma 2.8 there exists a $\pi \in \Pi$ with $\boldsymbol{y} = \boldsymbol{\mu f}^{\pi}$. In particular we have $\boldsymbol{y} = \boldsymbol{f}^{\pi}(\boldsymbol{y})$. Let $\boldsymbol{z}$ be any prefixed point of $\boldsymbol{f}$. As $\boldsymbol{f}(\boldsymbol{z}) \leq \boldsymbol{f}^{\pi}(\boldsymbol{z})$, it follows $\boldsymbol{z} \leq \boldsymbol{f}^{\pi}(\boldsymbol{z})$. Since $\boldsymbol{y} \succ \boldsymbol{0}$, there is an $\varepsilon > 0$ such that $\boldsymbol{x} := \boldsymbol{y} + \varepsilon(\boldsymbol{y} - \boldsymbol{z}) \succ \boldsymbol{0}$. We have:

$$\begin{aligned}
\boldsymbol{x} &= \boldsymbol{y} + \varepsilon(\boldsymbol{y} - \boldsymbol{z}) && \text{(definition of } \boldsymbol{x}\text{)} \\
&= \boldsymbol{y} + \varepsilon(\boldsymbol{f}^{\pi}(\boldsymbol{y}) - \boldsymbol{z}) && (\boldsymbol{y} = \boldsymbol{f}^{\pi}(\boldsymbol{y})) \\
&= \boldsymbol{y} + \varepsilon(\boldsymbol{f}^{\pi}(\boldsymbol{z}) + (\boldsymbol{f}^{\pi})' \cdot (\boldsymbol{y} - \boldsymbol{z}) - \boldsymbol{z}) && (\boldsymbol{f}^{\pi} \text{ is linear}) \\
&\geq \boldsymbol{y} + \varepsilon(\boldsymbol{f}^{\pi})' \cdot (\boldsymbol{y} - \boldsymbol{z}) && (\boldsymbol{z} \leq \boldsymbol{f}^{\pi}(\boldsymbol{z})) \\
&= \boldsymbol{f}^{\pi}(\boldsymbol{y}) + (\boldsymbol{f}^{\pi})' \cdot \varepsilon(\boldsymbol{y} - \boldsymbol{z}) && (\boldsymbol{y} = \boldsymbol{f}^{\pi}(\boldsymbol{y})) \\
&= \boldsymbol{f}^{\pi}(\boldsymbol{y} + \varepsilon(\boldsymbol{y} - \boldsymbol{z})) && (\boldsymbol{f}^{\pi} \text{ is linear}) \\
&= \boldsymbol{f}^{\pi}(\boldsymbol{x}) && \text{(definition of } \boldsymbol{x}\text{)}\ ,
\end{aligned}$$

i.e., $\boldsymbol{x}$ is a postfixed point of $\boldsymbol{f}^{\pi}$. By Knaster-Tarski's theorem, $\boldsymbol{y} = \boldsymbol{\mu f}^{\pi}$ is the least postfixed point of $\boldsymbol{f}^{\pi}$, hence $\boldsymbol{y} \leq \boldsymbol{x}$. But this implies with the definition of $\boldsymbol{x}$ that $\boldsymbol{z} \leq \boldsymbol{y}$. As the prefixed point $\boldsymbol{z}$ of $\boldsymbol{f}$ was chosen arbitrarily, $\boldsymbol{y}$ is the greatest prefixed point of $\boldsymbol{f}$. □

The correctness of Algorithm 2.2 follows from Proposition 2.35 and Lemma 2.36.

The following theorem shows the second major advantage of the lazy $\boldsymbol{\nu}$-method, namely, that the strategies $\sigma^{(k)}$ are meaningful in terms of games.

Theorem 2.37. *Let $\Pi = \Pi^*$. Let $(\boldsymbol{\nu}^{(k)})_{k \in \mathbb{N}}$ be a lazy $\boldsymbol{\nu}$-sequence. Then $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu f}^{\sigma^{(k)}}$ holds for all $k \in \mathbb{N}$.*

Proof. Immediate from Lemma 2.28. □

In terms of extinction games, the inequality $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu f}^{\sigma^{(k)}}$ of Theorem 2.37 means that, no matter how the savior plays, the terminator achieves an extinction probability of at least $\boldsymbol{\nu}^{(k)}$ by using the strategy $\sigma^{(k)}$. As $(\boldsymbol{\nu}^{(k)})$ converges to $\boldsymbol{\mu}$ by Proposition 2.26, these lower bounds on the terminator's winning chances come arbitrarily close to the winning probability in optimal play. We say, the $\vee$-strategies $\sigma^{(k)}$ are *ε-optimal.* Moreover, since $(\boldsymbol{\nu}^{(k)})$ converges to $\boldsymbol{\mu}$ and there are only finitely many strategies, there is an $i \in \mathbb{N}$ such that $\boldsymbol{\mu f}^{\sigma^{(i+j)}} = \boldsymbol{\mu}$ holds for all $j \geq 0$, i.e., ultimately, the $\sigma^{(k)}$ are optimal. It is not clear, however, how to compute the i for which $\sigma^{(i+j)}$ is optimal for all j.

It is an open question whether the $\boldsymbol{\tau}$-method can be modified to yield ε-optimal strategies.

Example 2.38 (Application to the primaries game)**.** *We solved the equation system of Example 2.11 (page 58) approximatively by performing* 5 *iterations of the lazy* $\boldsymbol{\nu}$*-method. Using Theorem 2.37 we found that Clinton can extinguish an individual of problem (a) with a probability of at least* $X_a = 0.492$ *by concentrating on her program and her "ready from day 1" message. (More than 70 Kleene iterations would be needed to infer that* X_a *is at least* 0.49*.) As* $\boldsymbol{\nu}^{(5)}$ *seems to solve above equation system quite well in the sense that* $\|\boldsymbol{f}(\boldsymbol{\nu}^{(5)}) - \boldsymbol{\nu}^{(5)}\|$ *is small, we recommend Obama to talk about Iraq. Since* $\nu^{(2)}_{X_1} > 0.38$ *and* $\sigma^{(2)}(1) = 0.3 + 0.7X_1^2$*, Clinton's team can use Theorem 2.37 after only 2 iterations to infer that* $X_a \geq 0.38$ *by showing emotions and using her "ready from day 1" message.*

As commented above, we cannot guarantee Obama or Clinton that the recommended strategies are optimal, and we do not know how many iterations of the $\boldsymbol{\nu}$*-method are needed to yield an optimal strategy for Clinton. In a pragmatic sense, it seems plausible that* $\boldsymbol{\nu}^{(k)}$ *is close to* $\boldsymbol{\mu}$ *when* $\|\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}\|$ *is small, but proofs of such claims would have to be constructed case-by-case.*

We illustrate the $\boldsymbol{\nu}$-method by two more examples of extinction games.

Example 2.39. *Consider again the flu example from Example 0.8 (page 11) which gives rise to the following equation system:*

$$\begin{aligned} U &= 0.3 + 0.7UU \ \vee \ 0.9T + 0.1U \\ T &= 0.35 + 0.65TU \ \wedge \ 0.5 + 0.2TU + 0.3TUU \end{aligned}$$

Recall that μ_U *(resp.* μ_T*) are the probabilities that the doctor succeeds in extinguishing the flu that may spread from a patient who is not treated (resp. is treated) with Muniflu.*

The doctor uses the lazy $\boldsymbol{\nu}$*-method to compute the extinction probabilities and obtains the following sequences:*

k	0	1	2	3	4	5
$\boldsymbol{\nu}_U^{(k)}$	0	0.300	0.409	0.524	0.538	0.538
$\boldsymbol{\nu}_T^{(k)}$	0	0.350	0.465	0.524	0.538	0.538
$\sigma^{(k)}(U)$	p_1	p_1	p_2	p_2	p_2	p_2

Here, p_1, p_2 *stand for the polynomials* $0.3 + 0.7UU$ *and* $0.9T + 0.1U$*, respectively, in other words,* p_1 *stands for the doctor's action* not *to treat the patient, and* p_2 *stands for treating him with Muniflu. By virtue of Theorem 2.37, this lazy* $\boldsymbol{\nu}$*-sequence tells the doctor after two iterations that Muniflu achieves a cure with a probability of at least* 0.4*. After four iterations, she knows that Muniflu succeeds with a probability of at least* 0.538*. As* $\sigma^{(k)}(U) = p_2$ *holds for (at least)* $k \in \{2, 3, 4, 5\}$*, she should treat the patient with Muniflu.*

Example 2.40. *Consider the following card game between two players, Black and Red. Initially, a dealer places one card between the players, face up. We call it the current card. If the current card is black (spades or clubs), then it is Black's turn, otherwise it is Red's turn. The player in turn is dealt a new card, and chooses, without looking at it, between "swap" and "play". By "swap" the current card is replaced by the new one. By "play" the new card is uncovered; if it is higher than the current card, the new card is placed on top of the current card; else, both the new and the current card are removed from the game.*

Black wins if the pile of cards between the players becomes empty. Red wins if the game goes on forever.

The position of the game is given by the current pile of cards between the players. Since it can become arbitrarily large, the game has a potentially infinite number of positions. Denote by B_i *(*R_i*) the probability that if the current card is black (red) with number* i *the card is* *eventually removed from the pile. Assuming 13 ranks (as in Poker), the probability that Black wins by optimal play is given by the following equations for each* $1 \leq i \leq 13$*:*

$$\begin{aligned} B_i &= X \ \vee \ \frac{i}{13} + Y_i \cdot B_i & \qquad X &= \sum_{j=1}^{13} \frac{1}{26} B_j + \frac{1}{26} R_j \\ R_i &= X \ \wedge \ \frac{i}{13} + Y_i \cdot R_i & \qquad Y_i &= \sum_{j=i+1}^{13} \frac{1}{26} B_j + \frac{1}{26} R_j \end{aligned}$$

We performed 5 *iterations of the lazy* $\boldsymbol{\nu}$*-method to determine (using Theorem 2.37) that if Black "swaps" at* B_1, B_2, B_3, B_4 *and "plays" at the other black cards, then Black wins the game starting with a random card* X *with a probability of at least* 0.86*. As* $\boldsymbol{\nu}^{(5)}$ *seems to solve above system quite well, we read off from* $\boldsymbol{\nu}^{(5)}$ *the recommendation for Red to "play" at* R_1, R_2, R_3, R_4 *and otherwise "swap". There is no guarantee that these strategies are optimal.*

2.5 Comparisons

We have seen that the $\boldsymbol{\nu}$-method computes ε-optimal strategies for the terminator in extinction games, whereas it is open whether the $\boldsymbol{\tau}$-method can be used for that as well.

We have also seen that one step of the $\boldsymbol{\nu}$-method is cheaper to compute than one step of the $\boldsymbol{\tau}$-method, because the $\boldsymbol{\tau}$-method requires strategy iteration in each step, whereas each step of the $\boldsymbol{\nu}$-method reduces to one linear programming problem.

On the other hand, by Lemma 2.16.5, we have $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \geq \mathcal{N}_{\boldsymbol{f}^\sigma}$, and so it follows that $\boldsymbol{\tau}^{(k)} \geq \boldsymbol{\nu}^{(k)}$ holds for all $k \in \mathbb{N}$. This means that, counting the number of approximation steps, the $\boldsymbol{\tau}$-method is at least as "fast" as the $\boldsymbol{\nu}$-method. In the following (slightly contrived) example, the $\boldsymbol{\tau}$-method is, in fact, much "faster" than the $\boldsymbol{\nu}$-method.

Example 2.41. *Let* $\boldsymbol{f}$ *be the following min-max-SPP* $\boldsymbol{f}$ *which is parameterized by an arbitrary* $k \in \mathbb{N}$*:*

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} X_2 \wedge 2 \\ X_1^2 + 0.25 \ \vee \ X_1 + 2^{-2(k+1)} \end{pmatrix} .$$

Since the constant $2^{-2(k+1)}$ *is represented using* $\mathcal{O}(k)$ *bits,* $\boldsymbol{f}$ *is of a size linear in* k*. The lazy* $\boldsymbol{\nu}$*-method needs at least* k *steps, in fact, we have* $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu} - (1.5, 3.75)$*. The* $\boldsymbol{\tau}$*-method needs exactly* 2 *steps. These claims are proved in Appendix B.2.*

To give some intuition why the $\boldsymbol{\nu}$*-method performs badly here, we condense the 2-dimensional min-max-SPP* $\boldsymbol{f}$ *in a 1-dimensional version* g *with*

$$g(X) = \big(g_1(X) \vee g_2(X)\big) \ \wedge \ 2$$

where

$$g_1(X) = X^2 + 0.25 \quad \text{ and } \quad g_2(X) = X + 2^{-2(k+1)} .$$

Figure 2.6 shows the graph of $g(X) - X$ *for* $k = 2$*. Strictly speaking,* g *is not a min-max-SPP because it contains both a minimum and a maximum operator, but the* $\boldsymbol{\tau}$*- and the* $\boldsymbol{\nu}$*-method can be applied analogously in this example. The* $\boldsymbol{\tau}$*-method finds the least fixed point* $\mu = 2$ *in only one step: The linearization of* g *at* 0 *is*

$$\mathcal{L}\,(g, 0)\,(X) = \big(0.25 \ \vee \ X + 2^{-2(k+1)}\big) \ \wedge \ 2 \,,$$

and the least fixed point of this linearization is 2*, as can easily seen by computing the Kleene sequence for* $\mathcal{L}\,(g, 0)\,(X)$*.*

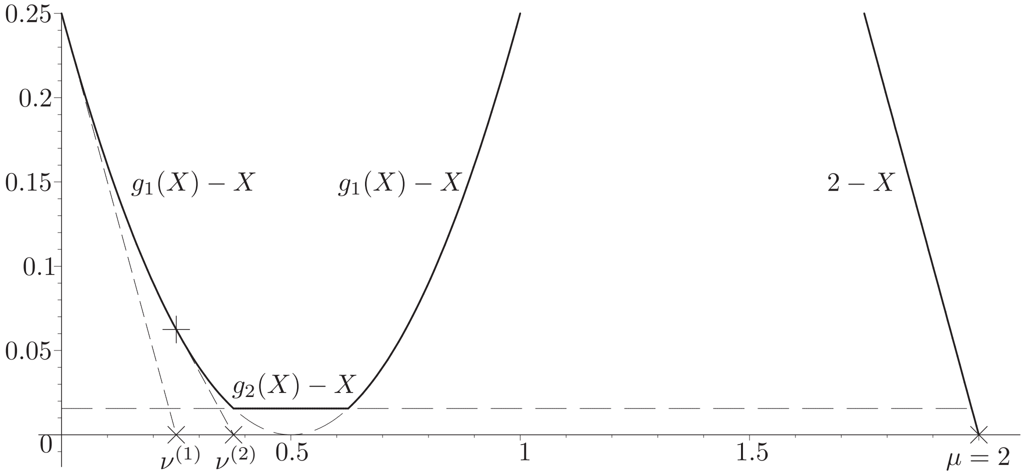

Figure 2.6: In this example the $\boldsymbol{\nu}$-method performs badly.

The $\boldsymbol{\nu}$-method, on the other hand, needs more steps. Figure 2.6 shows the first few iterates. Since $g_1(x) > g_2(x)$ for small x, the $\boldsymbol{\nu}$-method must take $\sigma^{(i)} = g_1(X)$ at the beginning. In the figure (which shows the case $k = 2$) this applies to $\sigma^{(0)}$ and $\sigma^{(1)}$. So, the $\boldsymbol{\nu}$-method does not linearize g_2 at the beginning. In some sense, it "tries" to approximate the least fixed point of g_1 (which is $1/2$) until it enters the region where $g_2(x) > g_1(x)$ (in the figure this region is the interval $(3/8, 5/8)$). Once this region is reached, the $\boldsymbol{\nu}$-method takes $\sigma = g_2(X)$ and needs only one more step. The phase in which the $\boldsymbol{\nu}$-method produces iterates that are less than $1/2$ can be made arbitrarily long by moving down the graph of $g_2(X) - X$, i.e., by increasing k.

Comparison with PReMo

We now compare our approaches with the PReMo tool. PReMo [WE07] is a tool for analyzing probabilistic models with recursion. In particular, it can analyze a class of stochastic 2-player games, the so-called *1-exit recursive simple stochastic games* [EY05c]. PReMo computes the outcome of such games under optimal play by translating the game into a min-max-SPP $\boldsymbol{f}$ and computing its least fixed-point $\boldsymbol{\mu}$. PReMo employs 4 different techniques to approximate $\boldsymbol{\mu}$ for min-max-SPPs $\boldsymbol{f}$: It uses Newton's method only for SPPs without min or max. In this case both of our methods coincide with Newton's method. For min-max-SPPs, PReMo uses Kleene iteration, round-robin iteration (called Gauss-Seidel in [WE07]), and an "optimistic" variant of Kleene which is not guaranteed to converge. In the following we compare our algorithms only with Kleene iteration, as our algorithms are guaranteed to converge and one round-robin step is not faster than n Kleene steps.

Our methods improve on Kleene iteration in the sense that we have both $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\tau}^{(k)}$ and $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\nu}^{(k)}$ for all $k \in \mathbb{N}$, and our methods converge linearly, whereas Kleene iteration does not converge linearly in general. For example, consider the SPP $f(X) = \frac{1}{2}X^2 + \frac{1}{2}$ with $\mu f = 1$. Kleene iteration needs exponentially many iterations for i bits (see Example 0.4), whereas Newton's method gives exactly 1 bit per iteration. For the slightly modified SPP $\tilde{f}(X) = f(X) \wedge 1$ which has the same fixed point, PReMo no longer uses Newton's method, as $\tilde{f}$ contains a minimum. Our algorithms still produce exactly 1 bit per iteration.

In the case of linear min-max systems our methods compute the precise solution and not only an approximation. This applies, for example, to the max-linear system of [WE07] describing the expected time of termination of a nondeterministic variant of Quicksort.

Notice that Kleene iteration usually does not compute the precise solution, even for linear SPPs without minimum or maximum.

We implemented our algorithms prototypically in Maple and ran them on the quadratic nonlinear min-max-SPP describing the termination probabilities of a recursive simple stochastic game. This game stems from the example suite of PReMo (`rssg2.c`) and we used PReMo to produce the equations. Both of our algorithms reached the least fixed point after 2 iterations. So we could compute the precise $\boldsymbol{\mu}$ and optimal strategies for both players, whereas PReMo computes only approximations of $\boldsymbol{\mu}$.

2.6 Conclusions

Computing the least fixed point of min-max-SPPs is a central problem in the analysis of certain two-player stochastic games such as extinction games. We have presented the first methods for approximatively computing the least fixed point of min-max-SPPs, which are guaranteed to converge at least linearly. Both of them are generalizations of Newton's method. Whereas the $\boldsymbol{\tau}$-method converges faster in terms of the number of approximation steps, one approximation step of the $\boldsymbol{\nu}$-method is cheaper. Furthermore, we have shown that the $\boldsymbol{\nu}$-method computes ε-optimal strategies for the terminator in extinction games.

There are several open problems.

- One would like to know how many iterations of our methods are necessary to reach $\boldsymbol{\mu}$ within a certain precision. We have established the convergence order of our methods (linear), but do not yet have bounds on the threshold $k_{\boldsymbol{f}}$. Alternatively, one may look for sufficient criteria guaranteeing that the current approximant is close to $\boldsymbol{\mu}$.
- Our methods need to be evaluated in practice. In particular, the influence of imprecise computation through floating point arithmetic should be studied.
- Can the $\boldsymbol{\tau}$-method, like the $\boldsymbol{\nu}$-method, be used to compute ε-optimal strategies?
- How can optimal strategies be computed? More precisely, we know that the strategies computed by the lazy $\boldsymbol{\nu}$-method are eventually optimal. But it is open how to determine how many iterations are needed.

Chapter 3

Generalizing Newton's Method: An Epilogue

In this thesis, we have studied fixed-point equations $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ and algorithms, based on Newton's method, that compute the least solution $\boldsymbol{\mu}$. In the first part of the thesis, the function $\boldsymbol{f}$ was a vector of polynomials with nonnegative real coefficients, and we applied $\boldsymbol{f}$ mainly to vectors of nonnegative reals. In particular, $\boldsymbol{\mu}$ was a vector of nonnegative reals.

In this chapter we sketch an abstraction from the nonnegative reals. We argue that the notions of a polynomial vector $\boldsymbol{f}$, its least fixed point $\boldsymbol{\mu}$, and even Newton's method to compute $\boldsymbol{\mu}$ can all be generalized from the nonnegative reals to many more domains. Several theorems of this thesis are instances of more general theorems. Since interprocedural program analysis can be seen as the art of computing least fixed points of polynomial vectors, this generalization of Newton's method leads to new program analysis algorithms. The results mentioned in this chapter are described and proved in [EKL09] and have been published in [EKL07b, EKL07a, EKL08], see also Michael Luttenberger's recent PhD thesis [Lut09].

If the set of nonnegative reals is extended with $+\infty$, it can be seen as a *semiring* with the operations product and sum. This means, sum and product are associative and have neutral elements 0 and 1, respectively. Moreover, sum is commutative, and product distributes over sum. We call the semiring over the extended reals the *real semiring*. Like most naturally occurring semirings, the real semiring is *ω-continuous*, which means that the sum operator can be extended to an infinite summation operator that satisfies some natural properties, see [Kui97]. We look only at ω-continuous semirings in this chapter.

The real semiring is special in that it is *commutative*, which means that the product operation is commutative. Many semirings (but not the real semiring) are *idempotent*, which means that the sum operation is idempotent.

Here are some examples for semirings apart from the real semiring:

(1) The real interval $[0, 1]$ constitutes an idempotent and commutative semiring, where the sum operation is maximum, and the product operation is multiplication.

(2) The set 2^{Σ^*} contains the languages over an alphabet Σ. It constitutes a semiring, where the sum operation is language union, and the product operation is language concatenation. This semiring is idempotent but not commutative.

(3) For some fixed $s \in \mathbb{N}$, the set $2^{\mathbb{N}^s}$ contains the sets of vectors whose s components are natural numbers. It constitutes a semiring, where the sum operation is set union, and the product operation is given as follows:

$$U \cdot V = \{(u_1 + v_1, \ldots, u_s + v_s) \mid (u_1, \ldots, u_s) \in U, (v_1, \ldots, v_s) \in V\}$$

This semiring is idempotent and commutative.

(4) For some fixed domain set D, the set $2^{D \times D}$ contains the binary relations over D. It constitutes a semiring, where the the sum operation is set union, and the product operation is the join of relations, i.e.

$$R \cdot S = \{(a, c) \mid \exists b \text{ with } (a, b) \in R,\ (b, c) \in S\}\,.$$

This semiring is idempotent but not commutative.

In the following, let S be any semiring. The concept of *polynomials* can be extended to S in a straightforward way: A polynomial f is any expression over variables and constants, using sum and product as operators. For instance, if we have $\boldsymbol{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$ as variables, then $f(\boldsymbol{X}) = a \cdot X_1 \cdot X_2 + b \cdot X_1$ is a polynomial, where $a, b \in S$. Note that any polynomial can be written as a sum of products, because semirings are distributive.

A *vector* $\boldsymbol{x} \in S^n$ is a *fixed point* of a *polynomial vector* $\boldsymbol{f}$ if $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}$. Those notions are defined as expected. One can define an order $\leq$ on S by setting $a \leq a + b$ for all $a, b \in S$. Extending this order component-wise on vectors allows to speak about the *least fixed point* $\boldsymbol{\mu}$ of a polynomial vector $\boldsymbol{f}$: it is the least solution (with respect to $\leq$) of $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$. Kleene's fixed-point theorem guarantees the existence of $\boldsymbol{\mu}$:

Theorem 3.1 (Kleene's fixed-point theorem)**.** *Every polynomial vector $\boldsymbol{f}$ has a least fixed point $\boldsymbol{\mu}$ in S^n, i.e., $\boldsymbol{\mu} = \boldsymbol{f}(\boldsymbol{\mu})$ and, in addition, $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{y})$ implies $\boldsymbol{\mu} \leq \boldsymbol{y}$. Moreover, the sequence $(\boldsymbol{\kappa}^{(k)})_{k \in \mathbb{N}}$ with $\boldsymbol{\kappa}^{(k)} = \boldsymbol{f}^k(\boldsymbol{0})$ is monotonically increasing with respect to $\leq$ (i.e., $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\kappa}^{(k+1)})$) and converges to $\boldsymbol{\mu}$.*

Note that the statement of this theorem is almost identical to the statements in Theorem 1.3 (page 17) and Theorem 2.5 (page 55), with the exception that the requirement that $\boldsymbol{f}$ be feasible is omitted. This is because it is implied by the ω-continuity of S, which we assume throughout this chapter. For the real semiring (which is ω-continuous like all mentioned semirings) this means that some components of $\boldsymbol{\mu}$ may be ∞ (recall that ∞ is an element of the real semiring).

The least fixed point $\boldsymbol{\mu}$ plays a central role for the mentioned application to interprocedural program analysis. This is illustrated in the following example.

Example 3.2. *Consider a simple program with a procedure X that either calls itself or directly returns. Assume for a moment that it calls itself with probability $a = 2/3$ and returns with probability $b = 1/3$. Such a program can be modeled using a probabilistic pushdown automaton (see § 1.1.4 on page 19) with exactly one control state (which we omit for brevity), one stack symbol X, and the following rules:*

$$X \overset{a}{\hookrightarrow} XX \qquad X \overset{b}{\hookrightarrow} \varepsilon$$

In § 1.1.4 we stated that the termination probability, i.e., the probability that the program eventually terminates, is given as the least solution μ of

$$X = f(X) = a \cdot X \cdot X + b\,. \tag{3.1}$$

With $a = 2/3$ and $b = 1/3$ we have $\mu = 1/2$.

Equation (3.1) *can be used to compute much more information on the program.*

(1) *Assume that we are interested in the probability of the most likely terminating execution of the program. To determine this probability, we reinterpret* (3.1) *as an equation over the semiring* (1) *in the above list. That is, we obtain the equation $X = f(X)$ with $f(X) = aXX \vee b$. The probability of the most likely terminating execution is the least fixed point μ of f. With $a = 2/3$ and $b = 1/3$ we have $\mu = 1/3$ (i.e., the most likely terminating execution is that X directly returns).*

(2) *Assume that the procedure X outputs the letter "a" whenever it calls itself and outputs "b" whenever it returns. In order to determine the language of possible output strings of terminating executions, we reinterpret* (3.1) *as an equation over the semiring* (2) *in the above list with $\Sigma = \{a, b\}$. That is, we obtain the language equation $X = f(X)$ with $f(X) = \{a\} \cdot X \cdot X \,\cup\, \{b\}$. The least solution is the language of output strings. It can be equivalently described as the language of the context-free grammar with the rules $X \to aXX$ and $X \to b$.*

(3) *Assume that we are only interested in how many letters "a" and "b" are output, not in which order. More precisely, we wish to determine the following set:*

$$M := \{(i, j) \mid \exists \text{ terminating execution that outputs } i \text{ letters "a" and } j \text{ letters "b"}\}$$

To determine M, we reinterpret (3.1) *as an equation over the semiring* (3) *in the above list with $s = 2$. That is, we obtain the equation $X = f(X)$ with*

$$f(X) = \{(1, 0)\} \cdot X \cdot X \;\cup\; \{(0, 1)\},$$

where we write $\cdot$ for the product operator defined for the semiring (3)*. The set M is the least fixed point of f.*

As in Chapter 1 and 2, the Kleene sequence $(\boldsymbol{\kappa}^{(k)})$ can be used to approximate $\boldsymbol{\mu}$, but the convergence may be slow.

Example 3.3. *We wish to compute the set M from the previous example, part (3). As explained there, we need to compute the least fixed point μ of the polynomial (or 1-dimensional polynomial vector) f with*

$$f(X) = \{(1, 0)\} \cdot X \cdot X \;\cup\; \{(0, 1)\}\,.$$

The Kleene sequence is given by:

$$\begin{array}{llll}
\kappa^{(0)} = \emptyset & \kappa^{(2)} = \{(0,1),(1,2)\} & \kappa^{(4)} = \{(0,1),(1,2),\ldots,(7,8)\} & \cdots \\
\kappa^{(1)} = \{(0,1)\} & \kappa^{(3)} = \{(0,1),(1,2),(2,3),(3,4)\} & \kappa^{(5)} = \{(0,1),(1,2),\ldots,(15,16)\} & \cdots
\end{array}$$

So we have $\kappa^{(k)} = \{(j-1, j) \mid 1 \le j \le 2^{k-1}\}$ for all $k \ge 0$. By Theorem 3.1 it follows

$$\mu = \lim_{k\to\infty} \kappa^{(k)} = \{(j, j+1) \mid j \in \mathbb{N}\}\,.$$

Note that μ is an infinite set, whereas each $\kappa^{(k)}$ is finite. So, in some sense, the Kleene sequence converges "slowly".

We sketch in the following how to generalize Newton's method in order to obtain a sequence that converges faster to $\boldsymbol{\mu}$ than the Kleene sequence. Recall Theorem 1.12 from Chapter 1 (page 22):

Theorem 1.12 (weaker version). *Let $\boldsymbol{f}$ be an SPP. The Newton sequence $(\boldsymbol{\nu}^{(k)})$ with*

$$\boldsymbol{\nu}^{(0)} = \mathbf{0} \quad \text{and} \quad \boldsymbol{\nu}^{(k+1)} = \boldsymbol{\nu}^{(k)} + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^* \cdot \Big(\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}\Big)$$

is monotonically increasing, bounded from above by $\boldsymbol{\mu}$ (i.e. $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{\nu}^{(k+1)} \leq \boldsymbol{\mu}$), and converges to $\boldsymbol{\mu}$.

We are going to generalize Theorem 1.12 to semirings. So far, Theorem 1.12 seems to make little sense in terms of semirings:

(a) It is not clear what the derivative $\boldsymbol{f}'$ means in semirings.

(b) It is not clear what the matrix star $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*$ means in semirings.

(c) It is not clear what $\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}$ means, because the sum operator need not have an inverse in semirings.

All those obstacles can be overcome:

(a) For the derivative we take the algebraic definition, in other words, we apply the usual sum and product rules to calculate the derivative. For instance, for the polynomial $f(\boldsymbol{X}) = a \cdot X_1 \cdot X_2 + b \cdot X_1$ we have

$$\frac{\delta}{\delta X_1} f(\boldsymbol{X}) = a \cdot X_2 + b \quad \text{and} \quad \frac{\delta}{\delta X_2} f(\boldsymbol{X}) = a \cdot X_1 \,.$$

Just like in the real case, the partial derivatives of a polynomial vector are collected in the Jacobian matrix $\boldsymbol{f}'(\boldsymbol{X})$. For non-commutative semirings, the definition of derivatives is slightly more delicate [EKL07a].

(b) We have $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)}) \in S^{n \times n}$. For any square matrix $A \in S^{n \times n}$ we can define

$$A^* \;:=\; I \;+\; A \;+\; A \cdot A \;+\; A \cdot A \cdot A \;+\; \cdots$$

The concepts of the identity matrix I, matrix addition, matrix multiplication, matrix-vector multiplication, etc. can all be defined as expected. The assumption of ω-continuity gives the infinite sum A^* a well-defined meaning. Having already replaced the matrix inverse $\left(I - \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})\right)^{-1}$ from the original formulation of Theorem 1.12 by the matrix star $\boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^*$, we have avoided the seemingly even harder problem of computing matrix inverses in semirings.

(c) Note that Theorem 1.12 states $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)})$. If this inequality also held in semirings, we would have, by definition of the order $\leq$, for each $k \in \mathbb{N}$ a vector $\boldsymbol{\delta}^{(k)}$ such that

$$\boldsymbol{\nu}^{(k)} + \boldsymbol{\delta}^{(k)} = \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \,.$$

Consequently, we could replace $\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) - \boldsymbol{\nu}^{(k)}$ by $\boldsymbol{\delta}^{(k)}$. As it turns out, this approach works.

Having overcome problems (a), (b) and (c) as outlined, the following theorem can be proved:

Theorem 3.4. *Let $\boldsymbol{f}$ be a polynomial vector over any ω-continuous semiring. Define a Newton sequence $(\boldsymbol{\nu}^{(k)})$ by*

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{\nu}^{(k+1)} = \boldsymbol{\nu}^{(k)} + \boldsymbol{f}'(\boldsymbol{\nu}^{(k)})^* \cdot \boldsymbol{\delta}^{(k)} \,,$$

where $\boldsymbol{\delta}^{(k)}$ is any vector satisfying $\boldsymbol{\nu}^{(k)} + \boldsymbol{\delta}^{(k)} = \boldsymbol{f}(\boldsymbol{\nu}^{(k)})$. Then there is exactly one Newton sequence, and the Newton sequence $(\boldsymbol{\nu}^{(k)})$ is monotonically increasing, bounded from above by $\boldsymbol{\mu}$ (i.e. $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{\nu}^{(k+1)} \leq \boldsymbol{\mu}$), converges to $\boldsymbol{\mu}$, and does so at least as fast as the Kleene sequence (i.e. $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\nu}^{(k)}$).

Notice that the statement of Theorem 3.4 is very similar to Proposition 2.18 (page 63) and Proposition 2.26 (page 67).

Example 3.5. *Consider again the polynomial f from Example 3.3 with*

$$f(X) = \{(1,0)\} \cdot X \cdot X \ \cup \ \{(0,1)\} \,.$$

We wish to compute the Newton sequence $(\nu^{(k)})$ for f. First we compute the derivative f':

$$f'(X) \ = \ \{(1,0)\} \cdot X \ \cup \ \{(1,0)\} \cdot X \ = \ \{(1,0)\} \cdot X$$

Clearly, $\nu^{(0)} = \emptyset$. Further, we have $f(\nu^{(0)}) = \{(0,1)\}$, so we have to take $\delta^{(0)} = \{(0,1)\}$ to achieve $\nu^{(0)} \cup \delta^{(0)} = f(\nu^{(0)})$. As $\{(0,0)\}$ is the neutral element of the product operator, we obtain

$$\begin{aligned}
\nu^{(1)} &= \nu^{(0)} \ \cup \ f'(\nu^{(0)})^* \cdot \delta^{(0)} \\
&= \emptyset \ \cup \ (\{(1,0)\} \cdot \emptyset)^* \cdot \{(0,1)\} \\
&= \emptyset^* \cdot \{(0,1)\} = \big(\{(0,0)\} \cup \emptyset \cup \emptyset^2 \cup \cdots\big) \cdot \{(0,1)\} \\
&= \{(0,1)\} \,.
\end{aligned}$$

We have $f(\nu^{(1)}) = \{(1,0)\} \cdot \{(0,1)\} \cdot \{(0,1)\} \ \cup \ \{(0,1)\} = \{(0,1),(1,2)\}$, so we can take $\delta^{(1)} = \{(1,2)\}$ to achieve $\nu^{(1)} \cup \delta^{(1)} = f(\nu^{(1)})$. To compute $\nu^{(1)}$, we first compute the matrix star:

$$f'(\nu^{(1)})^* = (\{(1,0)\} \cup \{(0,1)\})^* = \{(1,1)\}^* = \{(j,j) \mid j \geq 0\}$$

This yields:

$$\begin{aligned}
\nu^{(2)} &= \nu^{(1)} \ \cup \ f'(\nu^{(1)})^* \cdot \delta^{(1)} \\
&= \{(0,1)\} \ \cup \ \{(j,j) \mid j \geq 0\} \cdot \{(1,2)\} \\
&= \{(0,1)\} \ \cup \ \{(j+1, j+2) \mid j \geq 0\} \\
&= \{(j, j+1) \mid j \geq 0\}
\end{aligned}$$

So, $\nu^{(2)}$ equals the least fixed point μ, which we have computed in Example 3.3. We conclude that in this example, the Newton sequence reaches μ in two iterations, whereas the Kleene sequence never reaches μ (and only converges to μ).

In the previous example, the Kleene sequence never reaches μ, whereas the Newton sequence reaches μ after finitely many steps. In fact, one can show the following theorem for arbitrary *commutative and idempotent* semirings:

Theorem 3.6. *Let $\boldsymbol{f}$ be a polynomial vector over a commutative and idempotent semiring. Let n be the number of components of $\boldsymbol{f}$. Then we have $\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\mu}$, i.e., the Newton sequence reaches $\boldsymbol{\mu}$ after $n+1$ iterations.*

In [EKL09] the Newton sequence is analyzed in much greater detail. The analysis includes also non-commutative and non-idempotent semirings.

We conclude that computing the least fixed point of polynomial vectors is a very general task for solving problems in various computer science areas, from stochastic models of web-surfers, over extinction games, to interprocedural program analysis. Newton's method provides a generic and efficient algorithm to perform this task. In particular, it vastly accelerates Kleene iteration, which is the traditional way of approximating fixed points.

Appendix A

Proofs of Chapter 1

A.1 Proof of Lemma 1.49

The proof of Lemma 1.49 is by a sequence of lemmata. The following two Lemmata A.1 and A.2 provide a lower bound on $\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}\|$ for an "almost-fixed-point" $\boldsymbol{x}$.

Lemma A.1. *Let $\boldsymbol{f}$ be a quadratic SPP without linear terms, i.e., $\boldsymbol{f}(\boldsymbol{X}) = B(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{c}$ where B is a bilinear map, and $\boldsymbol{c}$ is a constant vector. Let $\boldsymbol{f}(\boldsymbol{X})$ be non-constant in every component. Let $R \mathbin{\dot\cup} S = \{1, \ldots, n\}$ with $S \neq \emptyset$. Let every component depend on every S-component and not on any R-component. Then there is a constant $C_{\boldsymbol{f}} > 0$ such that*

$$\|\boldsymbol{f}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta})\| \geq C_{\boldsymbol{f}} \cdot \|\boldsymbol{\delta}\|^2$$

for all $\boldsymbol{\delta}$ with $\boldsymbol{0} \leq \boldsymbol{\delta} \leq \boldsymbol{\mu}$.

Proof. With the given component dependencies we can write $\boldsymbol{f}(\boldsymbol{X})$ as follows:

$$\boldsymbol{f}_R(\boldsymbol{X}) = \begin{pmatrix} \boldsymbol{f}_R(\boldsymbol{X}) \\ \boldsymbol{f}_S(\boldsymbol{X}) \end{pmatrix} = \begin{pmatrix} B_R(\boldsymbol{X}_S, \boldsymbol{X}_S) + \boldsymbol{c}_R \\ B_S(\boldsymbol{X}_S, \boldsymbol{X}_S) + \boldsymbol{c}_S \end{pmatrix}$$

A straightforward calculation shows

$$\boldsymbol{e}(\boldsymbol{\delta}) := \boldsymbol{f}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta}) = (I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{\delta} + B(\boldsymbol{\delta}, \boldsymbol{\delta}) \; .$$

Furthermore, $\partial_{\boldsymbol{X}_R} \boldsymbol{f}$ is constant zero in all entries, so

$$\begin{aligned} \boldsymbol{e}_R(\boldsymbol{\delta}) &= \boldsymbol{\delta}_R - \partial_{\boldsymbol{X}_S} \boldsymbol{f}_R(\boldsymbol{\mu}) \cdot \boldsymbol{\delta}_S + B_R(\boldsymbol{\delta}_S, \boldsymbol{\delta}_S) \qquad \text{and} \\ \boldsymbol{e}_S(\boldsymbol{\delta}) &= \boldsymbol{\delta}_S - \partial_{\boldsymbol{X}_S} \boldsymbol{f}_S(\boldsymbol{\mu}) \cdot \boldsymbol{\delta}_S + B_S(\boldsymbol{\delta}_S, \boldsymbol{\delta}_S) \; . \end{aligned}$$

Notice that for every real number $r > 0$ we have

$$\min_{\boldsymbol{0} \leq \boldsymbol{\delta} \leq \boldsymbol{\mu}, \|\boldsymbol{\delta}\| \geq r} \frac{\|\boldsymbol{e}(\boldsymbol{\delta})\|^2}{\|\boldsymbol{\delta}\|} > 0 \; ,$$

because otherwise $\boldsymbol{\mu} - \boldsymbol{\delta} < \boldsymbol{\mu}$ would be a fixed point of $\boldsymbol{f}$. We have to show:

$$\inf_{\boldsymbol{0} \leq \boldsymbol{\delta} \leq \boldsymbol{\mu}, \|\boldsymbol{\delta}\| > 0} \frac{\|\boldsymbol{e}(\boldsymbol{\delta})\|^2}{\|\boldsymbol{\delta}\|} > 0$$

Assume, for a contradiction, that this infimum equals zero. Then there exists a sequence $(\boldsymbol{\delta}^{(i)})_{i\in\mathbb{N}}$ with $\mathbf{0} \leq \boldsymbol{\delta}^{(i)} \leq \boldsymbol{\mu}$ and $\left\|\boldsymbol{\delta}^{(i)}\right\| > 0$ such that

$$\lim_{i\to\infty} \left\|\boldsymbol{\delta}^{(i)}\right\| = 0 \quad \text{and} \quad \lim_{i\to\infty} \frac{\left\|\boldsymbol{e}(\boldsymbol{\delta}^{(i)})\right\|}{\left\|\boldsymbol{\delta}^{(i)}\right\|^2} = 0 .$$

Define $r^{(i)} := \left\|\boldsymbol{\delta}^{(i)}\right\|$ and $\boldsymbol{d}^{(i)} := \frac{\boldsymbol{\delta}^{(i)}}{\left\|\boldsymbol{\delta}^{(i)}\right\|}$. Notice that $\boldsymbol{d}^{(i)} \in \{\boldsymbol{d} \in \mathbb{R}^n_{\geq 0} \mid \|\boldsymbol{d}\| = 1\} =: D$ where D is compact. So some subsequence of $(\boldsymbol{d}^{(i)})_{i\in\mathbb{N}}$, say w.l.o.g. the sequence $(\boldsymbol{d}^{(i)})_{i\in\mathbb{N}}$ itself, converges to some vector $\boldsymbol{d}^*$. By our assumption we have

$$\boldsymbol{e}(\boldsymbol{\delta}^{(i)}) / \left\|\boldsymbol{\delta}^{(i)}\right\|^2 = \left\|\frac{1}{r^{(i)}}(I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{d}^{(i)} + B(\boldsymbol{d}^{(i)}, \boldsymbol{d}^{(i)})\right\| \longrightarrow 0 . \tag{A.1}$$

As $B(\boldsymbol{d}^{(i)}, \boldsymbol{d}^{(i)})$ is bounded, $\frac{1}{r^{(i)}}(I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{d}^{(i)}$ must be bounded, too. Since $r^{(i)}$ converges to 0, $\left\|(I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{d}^{(i)}\right\|$ must converge to 0, so

$$(I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{d}^* = \mathbf{0} .$$

In particular, $\left((I - \boldsymbol{f}'(\boldsymbol{\mu}))\boldsymbol{d}^*\right)_R = \boldsymbol{d}^*_R - \partial_{\boldsymbol{X}_S}\boldsymbol{f}_R(\boldsymbol{\mu}) \cdot \boldsymbol{d}^*_S = \mathbf{0}$. So we have $\boldsymbol{d}^*_S > \mathbf{0}$, because $\boldsymbol{d}^*_S = \mathbf{0}$ would imply $\boldsymbol{d}^*_R = \mathbf{0}$ which would contradict $\boldsymbol{d}^* > \mathbf{0}$.

In the remainder of the proof we focus on $\boldsymbol{f}_S$. Define the scSPP $\boldsymbol{g}(\boldsymbol{X}_S) := \boldsymbol{f}_S(\boldsymbol{X})$. Notice that $\boldsymbol{\mu g} = \boldsymbol{\mu}_S$. We can apply Lemma 1.31 to $\boldsymbol{g}$ and $\boldsymbol{d}^*_S$ and obtain $\boldsymbol{d}^*_S \succ \mathbf{0}$. As $\boldsymbol{f}_S(\boldsymbol{X})$ is non-constant we get $B_S(\boldsymbol{d}^*_S, \boldsymbol{d}^*_S) \succ \mathbf{0}$. By (A.1), $\frac{1}{r^{(i)}}(I - \boldsymbol{g}'(\boldsymbol{\mu g}))\boldsymbol{d}^{(i)}_S$ converges to $-B_S(\boldsymbol{d}^*_S, \boldsymbol{d}^*_S) \prec \mathbf{0}$. So there is a $j \in \mathbb{N}$ such that $(I - \boldsymbol{g}'(\boldsymbol{\mu g}))\boldsymbol{d}^{(j)}_S \prec \mathbf{0}$. Let $\widetilde{\boldsymbol{\delta}} := r\boldsymbol{d}^{(j)}$ for some small enough $r > 0$ such that $\mathbf{0} < \widetilde{\boldsymbol{\delta}}^*_S \leq \boldsymbol{\mu g}$ and

$$\begin{aligned} \boldsymbol{e}_S(\widetilde{\boldsymbol{\delta}}) &= (I - \boldsymbol{g}'(\boldsymbol{\mu g}))\widetilde{\boldsymbol{\delta}}_S + B_S(\widetilde{\boldsymbol{\delta}}_S, \widetilde{\boldsymbol{\delta}}_S) \\ &= r(I - \boldsymbol{g}'(\boldsymbol{\mu g}))\boldsymbol{d}^{(j)}_S + r^2 B_S(\boldsymbol{d}^{(j)}_S, \boldsymbol{d}^{(j)}_S) \prec \mathbf{0} . \end{aligned}$$

So we have $\boldsymbol{g}(\boldsymbol{\mu g} - \widetilde{\boldsymbol{\delta}}_S) \prec \boldsymbol{\mu g} - \widetilde{\boldsymbol{\delta}}_S$. However, $\boldsymbol{\mu g}$ is the least point $\boldsymbol{x}$ with $\boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{x}$. Thus we get the desired contradiction. □

Lemma A.2. *Let $\boldsymbol{f}$ be a quadratic strongly connected SPP. Then there is a constant $C_{\boldsymbol{f}} > 0$ such that*

$$\|\boldsymbol{f}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta})\| \geq C_{\boldsymbol{f}} \cdot \|\boldsymbol{\delta}\|^2$$

for all $\boldsymbol{\delta}$ with $\mathbf{0} \leq \boldsymbol{\delta} \leq \boldsymbol{\mu}$.

Proof. Write $\boldsymbol{f}(\boldsymbol{X}) = B(\boldsymbol{X}, \boldsymbol{X}) + L\boldsymbol{X} + \boldsymbol{c}$ for a bilinear map B, a matrix L and a constant vector $\boldsymbol{c}$. By Theorem 1.12.2. the matrix $L^* = (I - L)^{-1} = (I - \boldsymbol{f}'(\mathbf{0}))^{-1}$ exists. Define the SPP $\widetilde{\boldsymbol{f}}(\boldsymbol{X}) := L^* B(\boldsymbol{X}, \boldsymbol{X}) + L^*\boldsymbol{c}$. A straightforward calculation shows that the sets of fixed points of $\boldsymbol{f}$ and $\widetilde{\boldsymbol{f}}$ coincide and that

$$\boldsymbol{f}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta}) = (I - L)\left(\widetilde{\boldsymbol{f}}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta})\right) .$$

Further we have

$$\left\|(I - L)\left(\widetilde{\boldsymbol{f}}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta})\right)\right\|_2 \geq \sigma_1(I - L)\left\|\widetilde{\boldsymbol{f}}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta})\right\|_2$$

where $\sigma_1(I - L)$ denotes the smallest singular value of $I - L$. Note that $\sigma_1(I - L) > 0$ because $I - L$ is invertible. So it suffices to show that there is a $C_{\boldsymbol{f}}$ with

$$\left\|\widetilde{\boldsymbol{f}}(\boldsymbol{\mu} - \boldsymbol{\delta}) - (\boldsymbol{\mu} - \boldsymbol{\delta})\right\| \geq C_{\boldsymbol{f}} \cdot \|\boldsymbol{\delta}\|^2 .$$

If $\boldsymbol{f}(\boldsymbol{X})$ is linear (i.e. $B(\boldsymbol{X},\boldsymbol{X}) \equiv \boldsymbol{0}$), then $\widetilde{\boldsymbol{f}}(\boldsymbol{X})$ is constant and we have $\left\|\widetilde{\boldsymbol{f}}(\boldsymbol{\mu}-\boldsymbol{\delta})-(\boldsymbol{\mu}-\boldsymbol{\delta})\right\| = \|\boldsymbol{\delta}\|$, so we are done in that case. Hence we can assume that some component of $B(\boldsymbol{X},\boldsymbol{X})$ is not the zero polynomial. It remains to argue that $\widetilde{\boldsymbol{f}}$ satisfies the preconditions of Lemma A.1. By definition, $\widetilde{\boldsymbol{f}}$ does not have linear terms. Define

$$S := \{i \mid 1 \le i \le n,\ X_i \text{ is contained in a component of } B(\boldsymbol{X},\boldsymbol{X})\}\ .$$

Notice that S is non-empty. Let $i_0, i_1, \ldots, i_m, i_{m+1}$ ($m \ge 0$) be any sequence such that, in $\boldsymbol{f}$, for all j with $0 \le j < m$ the component i_j depends directly on i_{j+1} via a linear term and i_m depends directly on i_{m+1} via a quadratic term. Then i_0 depends directly on i_{m+1} via a quadratic term in $L^m B(\boldsymbol{X},\boldsymbol{X})$ and hence also in $\widetilde{\boldsymbol{f}}$. So all components are non-constant and depend (directly or indirectly) on every S-component. Furthermore, no component depends on a component that is not in S, because $L^* B(\boldsymbol{X},\boldsymbol{X})$ contains only S-components. Thus, Lemma A.1 can be applied, and the statement follows. □

The following lemma gives a bound on the propagation error for the case that $\boldsymbol{f}$ has a single top SCC.

Lemma A.3. *Let $\boldsymbol{f}$ be a quadratic SPP. Let $S \subseteq \{1,\ldots,n\}$ be the single top SCC of $\boldsymbol{f}$. Let $L := \{1,\ldots,n\} \setminus S$. Then there is a constant $C_{\boldsymbol{f}} \ge 0$ such that*

$$\|\boldsymbol{\mu}_S - \widetilde{\boldsymbol{\mu}}_S\| \le C_{\boldsymbol{f}} \cdot \sqrt{\|\boldsymbol{\mu}_L - \boldsymbol{x}_L\|}$$

for all $\boldsymbol{x}_L$ with $\boldsymbol{0} \le \boldsymbol{x}_L \le \boldsymbol{\mu}_L$ where $\widetilde{\boldsymbol{\mu}}_S := \boldsymbol{\mu}\left(\boldsymbol{f}_S[\boldsymbol{X}_L/\boldsymbol{x}_L]\right)$.

Proof. We write $\boldsymbol{f}_S(\boldsymbol{X}) = \boldsymbol{f}_S(\boldsymbol{X}_S,\boldsymbol{X}_L)$ in the following.

If S is a trivial SCC then $\boldsymbol{\mu}_S = \boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L)$ and $\widetilde{\boldsymbol{\mu}}_S = \boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{x}_L)$. In this case we have with Taylor's theorem (cf. Lemma 1.2)

$$\begin{aligned}
\|\boldsymbol{\mu}_S - \widetilde{\boldsymbol{\mu}}_S\| &= \|\boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L) - \boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{x}_L)\| \\
&\le \|\partial_{\boldsymbol{X}}\boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L)\cdot(\boldsymbol{\mu}_L - \boldsymbol{x}_L)\| \\
&\le \|\partial_{\boldsymbol{X}}\boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L)\| \cdot \|\boldsymbol{\mu}_L - \boldsymbol{x}_L\| \\
&= \|\partial_{\boldsymbol{X}}\boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L)\| \cdot \sqrt{\|\boldsymbol{\mu}_L - \boldsymbol{x}_L\|} \cdot \sqrt{\|\boldsymbol{\mu}_L - \boldsymbol{x}_L\|} \\
&\le \|\partial_{\boldsymbol{X}}\boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L)\| \cdot \sqrt{\|\boldsymbol{\mu}_L\|} \cdot \sqrt{\|\boldsymbol{\mu}_L - \boldsymbol{x}_L\|}
\end{aligned}$$

and the statement follows by setting $C_{\boldsymbol{f}} := \|\partial_{\boldsymbol{X}}\boldsymbol{f}_S(\boldsymbol{0},\boldsymbol{\mu}_L)\| \cdot \sqrt{\|\boldsymbol{\mu}_L\|}$.

Hence, in the following we can assume that S is a non-trivial SCC. Set $\boldsymbol{g}(\boldsymbol{X}_S) := \boldsymbol{f}_S(\boldsymbol{X}_S,\boldsymbol{\mu}_L)$. Notice that $\boldsymbol{g}$ is an scSPP with $\boldsymbol{\mu}\boldsymbol{g} = \boldsymbol{\mu}\boldsymbol{f}_S$. By applying Lemma A.2 to $\boldsymbol{g}$ and setting $c := 1/\sqrt{C_{\boldsymbol{g}}}$ (the $C_{\boldsymbol{g}}$ from Lemma A.2) we get

$$\begin{aligned}
\|\boldsymbol{\mu}_S - \widetilde{\boldsymbol{\mu}}_S\| &\le c \cdot \sqrt{\|\boldsymbol{g}(\boldsymbol{\mu}\boldsymbol{g} - (\boldsymbol{\mu}_S - \widetilde{\boldsymbol{\mu}}_S)) - (\boldsymbol{\mu}\boldsymbol{g} - (\boldsymbol{\mu}_S - \widetilde{\boldsymbol{\mu}}_S))\|} \\
&= c \cdot \sqrt{\|\boldsymbol{f}_S(\widetilde{\boldsymbol{\mu}}_S,\boldsymbol{\mu}_L) - \widetilde{\boldsymbol{\mu}}_S\|} \\
&= c \cdot \sqrt{\|\boldsymbol{f}_S(\widetilde{\boldsymbol{\mu}}_S,\boldsymbol{\mu}_L) - \boldsymbol{f}_S(\widetilde{\boldsymbol{\mu}}_S,\boldsymbol{x}_L)\|}
\end{aligned}$$

and with Taylor's theorem (cf. Lemma 1.2):

$$\begin{aligned}
&\le c \cdot \sqrt{\|\partial_{\boldsymbol{X}_L}\boldsymbol{f}_S(\widetilde{\boldsymbol{\mu}}_S,\boldsymbol{\mu}_L)(\boldsymbol{\mu}_L - \boldsymbol{x}_L)\|} \\
&\le c \cdot \sqrt{\|\partial_{\boldsymbol{X}_L}\boldsymbol{f}_S(\boldsymbol{\mu}_S,\boldsymbol{\mu}_L)(\boldsymbol{\mu}_L - \boldsymbol{x}_L)\|} \\
&\le c \cdot \sqrt{\|\partial_{\boldsymbol{X}_L}\boldsymbol{f}_S(\boldsymbol{\mu}_S,\boldsymbol{\mu}_L)\|} \cdot \sqrt{\|\boldsymbol{\mu}_L - \boldsymbol{x}_L\|}
\end{aligned}$$

So the statement follows by setting $C_{\boldsymbol{f}} := c \cdot \sqrt{\|\partial_{\boldsymbol{X}_L}\boldsymbol{f}_S(\boldsymbol{\mu}_S,\boldsymbol{\mu}_L)\|}$. □

Now we can extend Lemma A.3 to Lemma 1.49, restated here.

Lemma 1.49. *There is a constant* $C_{\boldsymbol{f}} > 0$ *such that*

$$\left\|\boldsymbol{\mu}_{[t]} - \widetilde{\boldsymbol{\mu}}_{[t]}\right\| \leq C_{\boldsymbol{f}} \cdot \sqrt{\left\|\boldsymbol{\mu}_{[>t]} - \boldsymbol{\rho}_{[>t]}\right\|}$$

holds for all $\boldsymbol{\rho}_{[>t]}$ *with* $\mathbf{0} \leq \boldsymbol{\rho}_{[>t]} \leq \boldsymbol{\mu}_{[>t]}$*, where* $\widetilde{\boldsymbol{\mu}}_{[t]} = \boldsymbol{\mu}\big(\boldsymbol{f}_{[t]}[[>t]/\boldsymbol{\rho}_{[>t]}]\big)$.

Proof. Observe that $\boldsymbol{\mu}_{[t]}$, $\widetilde{\boldsymbol{\mu}}_{[t]}$, $\boldsymbol{\mu}_{[>t]}$ and $\boldsymbol{\rho}_{[>t]}$ do not depend on the components of depth $< t$. So we can assume w.l.o.g. that $t = 0$. Let $\mathcal{SCC}(0) = \{S_1, \ldots, S_k\}$.

For any S_i from $\mathcal{SCC}(0)$, let $\boldsymbol{f}^{(i)}$ be obtained from $\boldsymbol{f}$ by removing all top SCCs except for S_i. Lemma A.2 applied to $\boldsymbol{f}^{(i)}$ guarantees a $C^{(i)}$ such that

$$\left\|\boldsymbol{\mu}_{S_i} - \widetilde{\boldsymbol{\mu}}_{S_i}\right\| \leq C^{(i)} \cdot \sqrt{\left\|\boldsymbol{\mu}_{[>0]} - \boldsymbol{\rho}_{[>0]}\right\|}$$

holds for all $\boldsymbol{\rho}_{[>0]}$ with $\mathbf{0} \leq \boldsymbol{\rho}_{[>0]} \leq \boldsymbol{\mu}_{[>0]}$. Using the equivalence of norms let w.l.o.g. the norm $\|\cdot\|$ be the maximum-norm $\|\cdot\|_\infty$. Let $C_{\boldsymbol{f}} := \max_{1 \leq i \leq k} C^{(i)}$. Then we have

$$\left\|\boldsymbol{\mu}_{[0]} - \widetilde{\boldsymbol{\mu}}_{[0]}\right\| = \max_{1 \leq i \leq k} \left\|\boldsymbol{\mu}_{S_i} - \widetilde{\boldsymbol{\mu}}_{S_i}\right\| \leq C_{\boldsymbol{f}} \cdot \sqrt{\left\|\boldsymbol{\mu}_{[>0]} - \boldsymbol{\rho}_{[>0]}\right\|}$$

for all $\boldsymbol{\rho}_{[>0]}$ with $\mathbf{0} \leq \boldsymbol{\rho}_{[>0]} \leq \boldsymbol{\mu}_{[>0]}$. □

Appendix B

Proofs of Chapter 2

B.1 Proof of Lemma 2.28

We need to show some technical lemmata before we can prove Lemma 2.28.

Lemma B.1. *Let $\boldsymbol{f}$ be an SPP. Let $S \subseteq \{1, \ldots, n\}$. Then $\boldsymbol{\mu}\big(\boldsymbol{f}[S/\boldsymbol{0}]\big) \le \boldsymbol{\mu}\big(\boldsymbol{f}^n[S/\boldsymbol{0}]\big)$.*

Proof. W.l.o.g. we assume $S = \{1, \ldots, l\}$ and set $T := \{l+1, \ldots, n\}$. First we show by induction on k that for all $k \ge 0$ and all $\boldsymbol{y} \in \mathbb{R}^{n-l}_{\ge 0}$ we have

$$(\boldsymbol{f}[S/\boldsymbol{0}])^k(\boldsymbol{y}) \le \boldsymbol{f}^k_T(\boldsymbol{0}, \boldsymbol{y}) \; . \tag{B.1}$$

The base case $k = 0$ is trivial. Let $k \ge 0$. Then:

$$\begin{aligned}
(\boldsymbol{f}[S/\boldsymbol{0}])^{k+1}(\boldsymbol{y}) &= \boldsymbol{f}[S/\boldsymbol{0}]\big((\boldsymbol{f}[S/\boldsymbol{0}])^k(\boldsymbol{y})\big) \\
&\le \boldsymbol{f}[S/\boldsymbol{0}]\big(\boldsymbol{f}^k_T(\boldsymbol{0}, \boldsymbol{y})\big) && \text{(induction)} \\
&= \boldsymbol{f}_T\big(\boldsymbol{0}, \boldsymbol{f}^k_T(\boldsymbol{0}, \boldsymbol{y})\big) \\
&\le \boldsymbol{f}_T\big(\boldsymbol{f}^k_S(\boldsymbol{0}, \boldsymbol{y}),\ \boldsymbol{f}^k_T(\boldsymbol{0}, \boldsymbol{y})\big) \\
&= \boldsymbol{f}^{k+1}_T(\boldsymbol{0}, \boldsymbol{y})
\end{aligned}$$

Since $\boldsymbol{f}^k_T(\boldsymbol{0}, \boldsymbol{y}) = \boldsymbol{f}^k[S/\boldsymbol{0}](\boldsymbol{y})$, Equation (B.1) implies $\boldsymbol{\mu}\big((\boldsymbol{f}[S/\boldsymbol{0}])^k\big) \le \boldsymbol{\mu}\big(\boldsymbol{f}^k[S/\boldsymbol{0}]\big)$. By Kleene's theorem we have $\boldsymbol{\mu}(\boldsymbol{f}[S/\boldsymbol{0}]) = \boldsymbol{\mu}\big((\boldsymbol{f}[S/\boldsymbol{0}])^n\big) \le \boldsymbol{\mu}\big(\boldsymbol{f}^n[S/\boldsymbol{0}]\big)$. □

Lemma B.2. *Let $\boldsymbol{f}$ be a feasible max-SPP and $\boldsymbol{x} \le \boldsymbol{f}^\sigma(\boldsymbol{x})$ and $\boldsymbol{x} \le \boldsymbol{\mu}\boldsymbol{f}^\sigma$ for some $\boldsymbol{x}$ and a strategy $\sigma \in \Sigma$. Let σ' be obtained from $\boldsymbol{x}$ and σ by a lazy strategy update. Then $\boldsymbol{x} \le \boldsymbol{\mu}\boldsymbol{f}^{\sigma'}$.*

Proof. In order to prove this lemma, we would like to apply Lemma 2.3 to $\boldsymbol{f}^{\sigma'}$. However, it turns out that it is more convenient to apply it to $\boldsymbol{f}^{(n)}$, where $\boldsymbol{f}^{(k)}$ is, for all $k \ge 1$, defined as the function $\boldsymbol{f}^{(k)} := \big(\boldsymbol{f}^{\sigma'}\big)^k$, the k-fold application of $\boldsymbol{f}^{\sigma'}$. By Kleene's theorem we have $\boldsymbol{\mu}\boldsymbol{f}^{(k)} = \boldsymbol{\mu}\boldsymbol{f}^{\sigma'}$ and, by monotonicity of $\boldsymbol{f}^{\sigma'}$, $\boldsymbol{x} \le \boldsymbol{f}^\sigma(\boldsymbol{x}) \le \boldsymbol{f}^{\sigma'}(\boldsymbol{x}) \le \boldsymbol{f}^{(k)}(\boldsymbol{x})$.

Define for all $k \in \mathbb{N}$ the sets S_k, T_k such that $S_k \dot{\cup} T_k = \{1, \ldots, n\}$ and $\boldsymbol{x}_{S_k} \prec \boldsymbol{f}^{(k)}_{S_k}(\boldsymbol{x})$ and $\boldsymbol{x}_{T_k} = \boldsymbol{f}^{(k)}_{T_k}(\boldsymbol{x})$. Let $S := S_n$ and $T := T_n$. Notice that $S = S_k$ and $T = T_k$ for all $k \ge n$. In particular, $\boldsymbol{x}_T = \boldsymbol{f}^{\sigma'}_T(\boldsymbol{f}^{(n)}(\boldsymbol{x}))$, which implies $\boldsymbol{x}_T = \boldsymbol{f}^{\sigma'}_T(\boldsymbol{x})$. By the lazy strategy update rule, σ and σ' are identical on T.

So we have $\boldsymbol{x}_T = \boldsymbol{f}^{\sigma'}_T(\boldsymbol{f}^{(n)}(\boldsymbol{x})) = \boldsymbol{f}^{\sigma}_T(\boldsymbol{f}^{(n)}(\boldsymbol{x})) = \boldsymbol{f}^{\sigma}[S/\boldsymbol{f}^{(n)}_S(\boldsymbol{x})](\boldsymbol{x}_T)$ and, similarly, $\boldsymbol{x}_T = \boldsymbol{f}^{\sigma}[S/\boldsymbol{x}_S](\boldsymbol{x}_T)$. In other words, $\boldsymbol{x}_T$ is a fixed point of both $\boldsymbol{f}^{\sigma}[S/\boldsymbol{f}^{(n)}_S(\boldsymbol{x})]$ and $\boldsymbol{f}^{\sigma}[S/\boldsymbol{x}_S]$. As $\boldsymbol{x}_S \prec \boldsymbol{f}^{(n)}_S(\boldsymbol{x})$, we have:

$$\boldsymbol{x}_T \text{ is a fixed point of } \boldsymbol{f}^{\sigma}[S/\boldsymbol{z}] \text{ for all vectors } \boldsymbol{z}. \tag{B.2}$$

This holds in particular for $\boldsymbol{z} = (\boldsymbol{\mu f}^{\sigma})_S$. Since by assumption $\boldsymbol{x}_T \le (\boldsymbol{\mu f}^{\sigma})_T = \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/(\boldsymbol{\mu f}^{\sigma})_S]\big)$, it follows $\boldsymbol{x}_T = \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/(\boldsymbol{\mu f}^{\sigma})_S]\big)$. Choose some $\boldsymbol{y}$ with $(\boldsymbol{\mu f}^{\sigma})_S \prec \boldsymbol{y}$. Then $\boldsymbol{x}_T \le \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/\boldsymbol{y}]\big)$. With (B.2) we have $\boldsymbol{x}_T = \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/\boldsymbol{y}]\big)$. By Lemma 2.1 we know that $\boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/\boldsymbol{z}]\big)$ is a positive power series. But it takes the same value $\boldsymbol{x}_T$ regardless if evaluated at $\boldsymbol{z} = (\boldsymbol{\mu f}^{\sigma})_S$ or at $\boldsymbol{z} = \boldsymbol{y}$. So it must, in fact, be constant, and (B.2) can be strengthened to

$$\boldsymbol{x}_T = \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/\boldsymbol{z}]\big) \text{ holds for all vectors } \boldsymbol{z}. \tag{B.3}$$

Now we have:

$$\begin{aligned}
\boldsymbol{x}_T &= \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma}[S/\boldsymbol{0}]\big) && \text{(Equation (B.3))}\\
&= \boldsymbol{\mu}\big(\boldsymbol{f}^{\sigma'}[S/\boldsymbol{0}]\big) && (\sigma, \sigma' \text{ identical on } T)\\
&\le \boldsymbol{\mu}\big(\boldsymbol{f}^{(n)}[S/\boldsymbol{0}]\big) && \text{(Lemma B.1)}\\
&\le \boldsymbol{\mu}\big(\boldsymbol{f}^{(n)}[S/\boldsymbol{x}_S]\big)
\end{aligned}$$

Recall that, by definition of S, we have $\boldsymbol{x}_S \prec \boldsymbol{f}^{(n)}_S(\boldsymbol{x})$. As $\boldsymbol{\mu} \ge \boldsymbol{f}^{\sigma'}(\boldsymbol{\mu})$, we also have $\boldsymbol{\mu} \ge \boldsymbol{f}^{(n)}(\boldsymbol{\mu})$ by monotonicity of $\boldsymbol{f}^{\sigma'}$. Hence, the preconditions of Lemma 2.3 are satisfied, and we conclude $\boldsymbol{x} \le \boldsymbol{\mu f}^{(n)} = \boldsymbol{\mu f}^{\sigma'}$. □

Lemma B.3. *Let $\boldsymbol{f}$ be a min-max-SPP and let $\boldsymbol{x} \le \boldsymbol{f}^{\sigma}(\boldsymbol{x})$ for some $\boldsymbol{x}$ and a strategy $\sigma \in \Sigma$. Let $\boldsymbol{x} \le \boldsymbol{\mu f}^{\sigma\pi}$ hold for a strategy $\pi \in \Pi^*$. Let σ' be obtained from $\boldsymbol{x}$ and σ by a lazy strategy update. Then $\boldsymbol{x} \le \boldsymbol{\mu f}^{\sigma'\pi}$.*

Proof. We have $\boldsymbol{x} \le \boldsymbol{f}^{\sigma}(\boldsymbol{x}) \le \boldsymbol{f}^{\sigma\pi}(\boldsymbol{x})$ and $\boldsymbol{x} \le \boldsymbol{\mu f}^{\sigma\pi}$. So Lemma B.2 can be applied to the feasible max-SPP $\boldsymbol{f}^{\pi}$ and we conclude $\boldsymbol{x} \le \boldsymbol{\mu f}^{\sigma'\pi}$. □

Now we are ready to prove Lemma 2.28 which is restated here.

Lemma 2.28. *Let $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ be a lazy $\boldsymbol{\nu}$-sequence. Then $\boldsymbol{\nu}^{(k)} \le \boldsymbol{\mu f}^{\sigma^{(k)}\pi}$ holds for all $k \in \mathbb{N}$ and for all $\pi \in \Pi^*$ (i.e., for all π such that $\boldsymbol{f}^{\pi}$ is feasible).*

Proof. Let $\pi \in \Pi^*$. We proceed by induction on k. The base case $k = 0$ is trivial. Let $k \ge 0$. We have:

$$\begin{aligned}
\boldsymbol{\nu}^{(k+1)} &= \mathcal{N}_{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)}) && (\text{definition of } \boldsymbol{\nu}^{(k+1)})\\
&\le \mathcal{N}_{\sigma^{(k)}\pi}(\boldsymbol{\nu}^{(k)}) && \text{(Lemma 2.16.6)}\\
&\le \mathcal{N}_{\sigma^{(k)}\pi}(\boldsymbol{\mu f}^{\sigma^{(k)}\pi}) && \text{(induction, Lemma 2.16.3)}\\
&= \boldsymbol{\mu f}^{\sigma^{(k)}\pi} && \text{(Lemma 2.16.2)}
\end{aligned}$$

Furthermore we know $\boldsymbol{\nu}^{(k)} \le \boldsymbol{f}(\boldsymbol{\nu}^{(k)})$ from Proposition 2.26 and $\boldsymbol{f}(\boldsymbol{\nu}^{(k)}) = \boldsymbol{f}^{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)})$ by definition of $\sigma^{(k)}$. So we have $\boldsymbol{\nu}^{(k)} \le \boldsymbol{f}^{\sigma^{(k)}}(\boldsymbol{\nu}^{(k)})$, and, by Lemma 2.16.4, $\boldsymbol{\nu}^{(k+1)} \le \boldsymbol{f}^{\sigma^{(k)}}(\boldsymbol{\nu}^{(k+1)})$. As $\sigma^{(k+1)}$ is obtained from $\boldsymbol{\nu}^{(k+1)}$ and $\sigma^{(k)}$ by a lazy strategy update, Lemma B.3 allows to conclude $\boldsymbol{\nu}^{(k+1)} \le \boldsymbol{\mu f}^{\sigma^{(k+1)}\pi}$. □

B.2 Proof for the Claims in Example 2.41

Let $k \in \mathbb{N} \setminus \{0\}$, and $\varepsilon := 2^{-2(k+1)}$. Let

$$g_1(X_1, X_2) = X_1^2 + 0.25, \quad g_2(X_1, X_2) = X_1 + \varepsilon$$

and consider the min-max-SPP

$$\boldsymbol{f}(X_1, X_2) = \begin{pmatrix} X_2 \wedge 2 \\ g_1(X_1, X_2) \vee g_2(X_1, X_2) \end{pmatrix} .$$

Its least fixed point is

$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 4.25 \end{pmatrix} .$$

We claim:

(1) The $\boldsymbol{\tau}$-method needs 2 iterations.

(2) The $\boldsymbol{\nu}$-method needs more than k iterations and $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\mu} - \begin{pmatrix} 1.5 \\ 3.75 \end{pmatrix}$.

We prove the claims in turn.

(1) For the linearization at $\mathbf{0}$ we have

$$\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \mathbf{0}\right)(\boldsymbol{X}) \geq \begin{pmatrix} X_2 \wedge 2 \\ \mathcal{L}\left(g_2, 0\right)(X_1) \end{pmatrix} = \begin{pmatrix} X_2 \wedge 2 \\ X_1 + \varepsilon \end{pmatrix} .$$

Thus, $\boldsymbol{\tau}^{(1)} = \mathcal{N}_{\boldsymbol{f}}(\mathbf{0}) = \boldsymbol{\mu}(\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}, \mathbf{0}\right)) \geq (2, 2+\varepsilon)^\top$. By Lemma 2.16.1 we have

$$\boldsymbol{\tau}^{(2)} \geq \boldsymbol{f}(\boldsymbol{\tau}^{(1)}) \geq \boldsymbol{f}\begin{pmatrix} 2 \\ 2+\varepsilon \end{pmatrix} = \begin{pmatrix} 2 \\ 4.25 \end{pmatrix} ,$$

so we conclude that $\boldsymbol{\tau}^{(2)} = \begin{pmatrix} 2 \\ 4.25 \end{pmatrix} = \boldsymbol{\mu}$.

(2) There are two possible $\vee$-strategies $\sigma_1, \sigma_2 \in \Sigma$, namely $\sigma_1(2) = g_1$ and $\sigma_2(2) = g_2$. We first show:

$$\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}} = \begin{pmatrix} 2^{-1} - 2^{-i-1} \\ 2^{-1} - 2^{-i-1} \end{pmatrix} \tag{B.4}$$

We proceed by induction on $i \in \mathbb{N}$. For the base case we have

$$\boldsymbol{\nu}^{(0)}_{\boldsymbol{f}^{\sigma_1}} = \mathbf{0} = \begin{pmatrix} 2^{-1} - 2^{0-1} \\ 2^{-1} - 2^{0-1} \end{pmatrix} .$$

For the induction step, let $i \geq 0$. We have:

$$\begin{aligned} &\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}^{\sigma_1}, \boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}\right)(\boldsymbol{x}) \\ &= \begin{pmatrix} x_2 \wedge 2 \\ (2^{-1} - 2^{-i-1})^2 + 2^{-2} + 2 \cdot (2^{-1} - 2^{-i-1}) \cdot (x_1 - (2^{-1} - 2^{-i-1})) \end{pmatrix} \\ &= \begin{pmatrix} x_2 \wedge 2 \\ 2^{-2} - (2^{-1} - 2^{-i-1})^2 + (1 - 2^{-i}) \cdot x_1 \end{pmatrix} \end{aligned}$$

Then

$$\boldsymbol{\nu}^{(i+1)}_{\boldsymbol{f}^{\sigma_1}} = \boldsymbol{\mu}\left(\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}^{\sigma_1}, \boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}\right) \vee \boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}\right) = \boldsymbol{\mu}\left(\boldsymbol{\mathcal{L}}\left(\boldsymbol{f}^{\sigma_1}, \boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}\right)\right) = \begin{pmatrix} y \\ y \end{pmatrix}$$

where y satisfies

$$y = 2^{-2} - (2^{-1} - 2^{-i-1})^2 + (1 - 2^{-i}) \cdot y \; .$$

Hence,

$$y = \frac{1}{1 - (1 - 2^{-i})} \cdot \left(2^{-2} - \left(2^{-1} - 2^{-i-1}\right)^2\right) = 2^{-1} - 2^{-i-2} \; .$$

This shows (B.4).

As a next subgoal, we show

$$g_1(\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}) \geq g_2(\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}), \qquad i = 1, \ldots, k. \tag{B.5}$$

Let $i \in \{1, \ldots, k\}$. We have $2^{-2(i+1)} \geq 2^{-2(k+1)} = \varepsilon$. Thus $2^{-1} - 2^{-i-1} + 2^{-2(i+1)} \geq 2^{-1} - 2^{-i-1} + \varepsilon$. Using (B.4) we get

$$\begin{aligned} g_1(\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}) &= (\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}})_1^2 + \frac{1}{4} = (2^{-1} - 2^{-i-1})^2 + \frac{1}{4} = 2^{-1} - 2^{-i-1} + 2^{-2(i+1)} \\ &\geq 2^{-1} - 2^{-i-1} + \varepsilon = (\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}})_1 + \varepsilon = g_2(\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}}) \; , \end{aligned}$$

so (B.5) is proved.

Consider the sequences $(\sigma^{(i)})$ and $(\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}})$ of occurring strategies and approximants of a lazy $\boldsymbol{\nu}$-sequence. Since $g_1(\mathbf{0}) = 0.25 > \frac{1}{16} \geq \varepsilon \geq g_2(\mathbf{0})$, we have $\sigma^{(0)} = \sigma_1$. From (B.5) we get $\sigma^{(i)} = \sigma_1$ for $i = 1, \ldots, k$ and thus in particular

$$\boldsymbol{\nu}^{(i)}_{\boldsymbol{f}} = \boldsymbol{\nu}^{(i)}_{\boldsymbol{f}^{\sigma_1}} = \begin{pmatrix} 2^{-1} - 2^{-i-1} \\ 2^{-1} - 2^{-i-1} \end{pmatrix} , \qquad i = 1, \ldots, k.$$

Hence, $\boldsymbol{\nu}^{(k)} = \begin{pmatrix} 2^{-1} - 2^{-k-1} \\ 2^{-1} - 2^{-k-1} \end{pmatrix} \leq \begin{pmatrix} 2^{-1} \\ 2^{-1} \end{pmatrix} = \boldsymbol{\mu} - \begin{pmatrix} 1.5 \\ 3.75 \end{pmatrix}$. □

Bibliography

[ABKPM09] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009.

[AN72] K.B. Athreya and P.E. Ney. *Branching Processes*. Springer-Verlag, 1972.

[BKS05] T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *Proceedings of STACS*, LNCS 3404, pages 145–157. Springer, 2005.

[BP79] A. Berman and R.J. Plemmons. *Nonnegative matrices in the mathematical sciences*. Academic Press, 1979.

[Can88] J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of STOC*, pages 460–467, 1988.

[Con92] A. Condon. The complexity of stochastic games. *Inf. and Comp.*, 96(2):203–224, 1992.

[DE04] R.D. Dowell and S.R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5(71), 2004.

[DEKM98] R. Durbin, S.R. Eddy, A. Krogh, and G.J. Michison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[DK80] D.W. Decker and C.T. Kelley. Newton's method at singular points I. *SIAM Journal on Numerical Analysis*, 17(1):66–70, 1980.

[EKL07a] J. Esparza, S. Kiefer, and M. Luttenberger. An extension of Newton's method to ω-continuous semirings. In *Proceedings of DLT*, LNCS 4588, pages 157–168, 2007.

[EKL07b] J. Esparza, S. Kiefer, and M. Luttenberger. On fixed point equations over commutative semirings. In *Proceedings of STACS*, LNCS 4397, pages 296–307, 2007.

[EKL08] J. Esparza, S. Kiefer, and M. Luttenberger. Newton's method for ω-continuous semirings. In *Proceedings of ICALP, part II*, LNCS 5126, pages 14–26, 2008. Invited paper.

[EKL09] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian program analysis. 2009. Submitted for publication. Available at `http://www.model.in.tum.de/um/bibdb/info/esparza.EKL09:newtProgAn.shtml`.

[EKM04] J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pages 12–21, 2004.

[EKM05] J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proceedings of LICS 2005*, pages 117–126. IEEE Computer Society Press, 2005.

[EY05a] K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic systems. In *Proceedings of TACAS*, LNCS 3440, pages 253–270, 2005.

[EY05b] K. Etessami and M. Yannakakis. Checking LTL properties of recursive Markov chains. In *Proceedings of 2nd Int. Conf. on Quantitative Evaluation of Systems (QEST'05)*, 2005.

[EY05c] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proceedings of ICALP*, LNCS 3580, pages 891–903, 2005.

[EY06] K. Etessami and M. Yannakakis. Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games. In *STACS*, pages 634–645, 2006.

[EY09] K. Etessami and M. Yannakakis. Recursive markov chains, stochastic grammars, and monotone systems of nonlinear equations. *Journal of the ACM*, 56(1):1–66, 2009. Earlier version appeared in STACS'05, pp. 340–352.

[FKK$^+$00] R. Fagin, A.R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons" (extended abstract). In *STOC*, pages 484–493, 2000.

[FKK$^+$01] R. Fagin, A.R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons". *Annals of Applied Probability*, 11(3):810–862, 2001.

[Fri09] O. Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *Proceedings of LICS*, 2009. To appear.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision processes.* Springer, 1997.

[GJ02] S. Geman and M. Johnson. Probabilistic grammars and their applications, 2002.

[GO81] A. Griewank and M.R. Osborne. Newton's method for singular problems when the dimension of the null space is > 1. *SIAM Journal on Numerical Analysis*, 18(1):145–149, 1981.

[GS07a] T. Gawlitza and H. Seidl. Precise fixpoint computation through strategy iteration. In *European Symposium on Programming (ESOP)*, LNCS 4421, pages 300–315, 2007.

[GS07b] T. Gawlitza and H. Seidl. Precise relational invariants through strategy iteration. In *Computer Science Logic (CSL)*, LNCS 4646, pages 23–40, 2007.

[Har63] T.E. Harris. *The Theory of Branching Processes.* Springer, 1963.

[HJ85] R.A. Horn and C.A. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[Kel95] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* SIAM, 1995.

[KH03] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31(13):3423–3428, 2003.

[Kui97] W. Kuich. *Handbook of Formal Languages*, volume 1, chapter 9: Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata, pages 609 – 677. Springer, 1997.

[LT85] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. Academic Press, second edition, 1985.

[Lut09] M. Luttenberger. *Solving Polynomial Systems on Semirings: A Generalization of Newton's Method*. PhD thesis, TU München, 2009.

[MS99] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[NS03] A. Neyman and S. Sorin. *Stochastic Games and Applications*. Kluwer Academic Press, 2003.

[OR70] J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.

[Ort72] J.M. Ortega. *Numerical Analysis: A Second Course*. Academic Press, New York, 1972.

[PP80] F.A. Potra and V. Ptak. Sharp error bounds for Newton's process. *Numerische Mathematik*, 34(1):63–72, 1980.

[Red78] G.W. Reddien. On Newton's method for singular problems. *SIAM Journal on Numerical Analysis*, 15:993–996, 1978.

[SBH$^+$94] Y. Sakabikara, M. Brown, R. Hughey, I.S. Mian, K. Sjolander, R.C. Underwood, and D. Haussler. Stochastic context-free grammars for tRNA. *Nucleic Acids Research*, 22:5112–5120, 1994.

[Ste00] I. Stewart. *Galois Theory*. Chapman and Hall, 3rd edition, 2000.

[WE07] D. Wojtczak and K. Etessami. PReMo: An analyzer for probabilistic recursive models. In *Proceedings of TACAS*, LNCS 4424, pages 66–71, 2007.

[WG74] H.W. Watson and F. Galton. On the probability of the extinction of families. *J. Anthropol. Inst. Great Britain and Ireland*, 4:138–144, 1874.