TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Informatik VII

Solving Polynomial Systems on Semirings: A Generalization of Newton's Method

Michael Luttenberger

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. T. Nipkow, Ph.D.
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. F. J. Esparza Estaun
	2.	Univ.-Prof. Dr. V. Diekert, Universität Stuttgart

Die Dissertation wurde am 25.06.2009 bei der Technischen Universität-München eingereicht und durch die Fakultät für Informatik am 02.02.2010 angenommen.

LaTeXed on Friday 5th March, 2010, 10:37

Zusammenfassung

Polynomielle Systeme über Semiringen treten in verschiedenen Bereichen der Informatik auf, so z.B. in der statischen Analyse prozeduraler Programme oder der Theorie formaler Sprachen. In dieser Dissertation wird ein neues Verfahren zur Berechnung des kleinsten Fixpunkts polynomieller Systeme vorgeschlagen. Hierbei handelt es sich um eine Verallgemeinerung des wohlbekannten Newton-Verfahrens zur Approximation von Nullstellen nichtlinearer Funktionen über den reellen Zahlen durch iterative Linearisierung. Es wird gezeigt, dass die vorgestellte Verallgemeinerung des Newton-Verfahrens mindestens genauso schnell gegen den kleinsten Fixpunkt konvergiert wie die übliche Fixpunktiteration. Weiterhin werden mehrere Klassen von Semiringen identifiziert, für welche das verallgemeinerte Newton-Verfahren, im Gegensatz zu der gewöhnlichen Fixpunktiteration, den kleinsten Fixpunkt bereits nach einer endlichen Anzahl von Iterationen erreicht. Schließlich ergeben sich aus diesen Konvergenzresultaten interessante Querbeziehungen zu anderen Themen aus dem Bereich der formalen Sprachen, wie z.B. Sprachen von endlichem Index oder dem Satz von Parikh.

Ausgehend von den Klassen von Semiringen, über welchen das verallgemeinerte Newton-Verfahren bereits nach endlich vielen Schritten den kleinsten Fixpunkt erreicht, werden in der Dissertation noch drei weitere Typen von Semiringen vorgestellt, welche es gestatten, den kleinsten Fixpunkt durch eine endliche Anzahl von Linearisierungen zu bestimmen. Hierbei erlauben es zwei der drei vorgestellten Semiringklassen den kleinsten Fixpunkt nur mit Hilfe einer Linearisierung bereits zu bestimmen, während im Fall der dritten Klasse auf eine Linearisierung ganz verzichtet werden kann.

Abschließend werden in der Dissertation noch Min-Max-Systeme betrachtet. Disese stellen eine natürliche Erweiterung polynomieller Systeme über totalgeordneten Semiringen dar. Es wird eine Klasse von Semiringen vorgestellt, die es erlaubt, nichtlineare Min-Max-Systeme mit Hilfe des bekannten Ansatzes der Strategieieration, erweitert auf nichtdeterministische Strategien, zu lösen.

Abstract

Systems of polynomials on semirings arise in several branches of computer science, like static analysis of procedural programs or formal language theory. We propose a new technique for calculating the least fixed points of such polynomial systems. This technique is a generalization of Newton's method, the well-known method for approximating a zero of a nonlinear function on the reals. We show that our generalization of Newton's method converges at least as fast as the standard fixed point iteration, and identify classes of semirings on which Newton's method even reaches the least fixed point after a finite number of steps in contrast to the standard fixed point iteration. We further obtain from these convergence results interesting links to other topics of formal language theory, for instance, languages of finite index and the Parikh theorem.

Motivated by our results on the convergence of Newton's method, we then identify three more classes of semirings which allow for an even faster calculation of the least fixed point. Two of these semirings allow for reducing a nonlinear polynomial system to a linear system in such a way that the least fixed point is preserved. In the third case already a finite number of standard fixed point iterations suffice to calculate the least fixed point, although this class of semirings does not satisfy the ascending chain condition.

We then turn to min-max-systems, a natural generalization of polynomial systems on semirings whose natural order is total. We identify a class of semirings which allow to solve nonlinear min-max-systems by the established approach of strategy iteration. In particular, we consider strategy iteration using nondeterministic strategies and show that these strategies allow for choosing an optimal successor.

Acknowledgments

First of all, I want to thank my parents and my sister for supporting and encouraging me through all these years and for giving me the opportunity to focus exclusively on school and university. This thesis would not have been possible without their support .

I would also like to thank Prof. Volker Diekert. He supported me in numerous ways while I was still studying at the University of Stuttgart, supervised my diploma thesis and introduced me to my supervisor Prof. Javier Esparza when I was looking for a position as a Ph.D. student.

In particular, I have to thank Javier for introducing me to an interesting research topic and for guiding and supporting my research. I also would like to thank him as he always takes the time to listen to you, your ideas or problems. I have learned that this is not a matter of course. ¡muchas gracias! Finally, thanks to Javier I have had the pleasure to work with an extraordinary group of people over the last five years:

For example, Stefan Schwoon, the only person I know who can optimize a bicycle tour w.r.t. several target functions, like total length, number of sights along the trip, or total time, and he can do this remarkably faster than google maps! Tobias Heindel, who began his Ph.D. studies at our department back in Stuttgart at the same time as I did, and with whom I have shared the sorrows most beginning Ph.D. students have. I would also like to mention Claus Schröter in whose office I wasted uncounted hours discussing music. Dejvuth Suwimonteerabuth aka Remy courageously endured my nonsense on the way to and back from the university here in Munich for the last three years (Dauerwelle vs. Minipli!). In particular, I am indebted to Stefan Kiefer aka Mr. Pinetree. Many of the results presented in this thesis were obtained in collaboration with him and Javier. Apart from that, Stefan may be the only person on this planet who has physically experienced Moshe Vardi resolving a deadlock. Thanks to these guys, my new colleagues here at the university of Munich (Jörg-einiges-geht, Andreas, Christian, Stefan, and the guys now at the university of Darmstadt) and, of course, Javier, I will always enjoy remembering this time of my life.

Of course, the list of people I feel, or should feel, indebted to is by no means complete. Therefore, a big thank you to my family in Tauberbischofsheim and my friends back in Stuttgart.

Contents

1 Introduction **1**
- 1.1 Interprocedural Dataflow-Analysis . . . 1
  - 1.1.1 From Semilattices to Semirings . . . 5
- 1.2 Solving Systems of Equations . . . 15
- 1.3 Contribution and Related Work . . . 16
- 1.4 Outline . . . 19

2 Preliminaries **21**
- 2.1 Basic Definitions and Notations . . . 21
- 2.2 ω-Continuous Semirings . . . 24

3 Newton's Method on ω-Continuous Semirings **33**
- 3.1 Generalizing Newton's Method . . . 33
  - 3.1.1 The Univariate Case . . . 33
  - 3.1.2 The Multivariate Case . . . 38
  - 3.1.3 Fundamental Properties of the Newton Sequences . . . 40
  - 3.1.4 Uniqueness . . . 44
- 3.2 Derivation Trees and the Newton Approximants . . . 47
  - 3.2.1 Kleene Sequence and Height . . . 49
  - 3.2.2 Newton Sequence and Dimension . . . 50
- 3.3 Idempotent Semirings . . . 53
  - 3.3.1 Language Semirings . . . 55
- 3.4 Commutative Idempotent Semirings . . . 56
  - 3.4.1 Analysis of the Convergence Speed . . . 57
  - 3.4.2 Generalization to Commutative Kleene Algebras . . . 61
  - 3.4.3 Comparison with previous proofs of Parikh's theorem . 64
- 3.5 Non-Distributive Program Analyses . . . 65

4 Derivation Tree Analysis **71**
- 4.1 Introduction . . . 71

4.2 Bamboos and their Yield . . . 73
4.3 Star-Distributive Semirings . . . 74
4.3.1 The (min, +)-Semiring . . . 76
4.3.2 Throughput of Grammars . . . 77
4.4 Lossy Semirings . . . 79
4.5 1-bounded Semirings . . . 81

5 Min-Max-Systems and Strategy Iteration 83
5.1 Introduction . . . 83
5.2 Min-Max-Systems on Totally Ordered cio-Semirings . . . 84
5.3 Strategy Iteration and Semirings . . . 88
5.3.1 Si-Semirings and Min-Max-Systems . . . 89
5.3.2 Nondeterministic and Reasonable Strategies . . . 92
5.3.3 Nondeterministic ⊔-Strategy Iteration . . . 98
5.3.4 Locally Optimal Successor Strategies . . . 101
5.3.5 Clean and Min-Cycle-Free Min-Max-Systems . . . 102
5.4 Linear Min-Max-Systems and Games . . . 105
5.4.1 Interpretation as Games . . . 105
5.4.2 An Improved Bound on the Number of Iterations . . . 110
5.5 Application to Parity Games . . . 114
5.6 Discussion and Related Work . . . 120

6 Geometrical Properties of Newton's Method 123
6.1 Introduction . . . 123
6.2 Preliminaries . . . 125
6.3 The Tangent Method . . . 126
6.4 Existence of a Second Fixed Point . . . 136

7 Conclusions 145
7.1 Contribution . . . 145
7.2 Open Problems . . . 147

A Missing Proofs of Chapter 3 149
A.1 Proofs of Section 3.2 . . . 149
A.2 Proofs of Section 3.3.1 . . . 152
A.3 Redko's Theorem and Commutative Kleene Algebras . . . 154

B Missing Proofs of Chapter 4 157
B.1 Proofs of Section 4.2 . . . 157
B.2 Proofs of Section 4.3 . . . 160
B.3 Proofs of Section 4.4 . . . 161

B.4 Proofs of Section 4.5 166

C Missing Proofs of Chapter 5 **169**
C.1 Proofs of Section 5.2 169
C.2 Proofs of Section 5.3 170

Chapter 1

Introduction

At the heart of this thesis lies the problem of approximating and calculating the least solution of equation systems

$$\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$$

where $\boldsymbol{f}$ is a system of polynomials, or more generally power series, on a specific algebraic structure called ω-continuous semirings. Polynomial equation systems on these structures arise naturally in several branches of computer science, e.g. in formal language theory, in abstract interpretation and static analysis of recursive programs, or in the analysis of probabilistic systems. Our approach for approximating the least solution is based on Newton's method (cf. Figure 1.1), the 300-year-old technique for computing a zero of a differentiable function, i.e., we generalize the idea of linearization to the setting of ω-continuous semirings, and approximate the least solution of $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ by solving a sequence of linear equation systems. We start with an illustration of the use of systems of polynomial equations in computer science, and discuss at hand of this example the results of this thesis in more detail.

1.1 Interprocedural Dataflow-Analysis

Systems of polynomial equations are a natural way of describing the dataflow of a program. One of the first to consider this approach to dataflow analysis have been Sharir and Pnueli in 1981, with many refinements and adaptions of

Let $f : \mathbb{R} \to \mathbb{R}$ be a continuously differentiable function, and let $f'|_x$ denote the derivative of f at some $x \in \mathbb{R}$. We may approximate the function f at x by its linearization

$$l_{f;x}(z) := f(x) + f'|_x(z - x).$$

Let z be a zero of f, and x some point close to z with $f'|_x \neq 0$. We then may approximate z by means of the zero of the linearization $l_{f;x}$, yielding the Newton operator $\mathcal{N}_f(x)$ defined by

$$\mathcal{N}_f(x) := x - f'|_x^{-1} f(x).$$

By repeatedly applying $\mathcal{N}_f$ to the current approximation starting with x we obtain the Newton sequence.

Figure 1.1: Newton's method.

their approach in subsequent years ([JM82, KS92, RHS95, SRH96, NNH99, RSJM05]). In this section, we give a short recap on the approach by Sharir and Pnueli to dataflow analysis. We then slightly generalize their approach which allows us to illustrate the problems considered in this thesis and the results we obtain on them.

Let us start with sketching the approach by Sharir and Pnueli to dataflow analysis ([SP81, JM82, KS92, RHS95, SRH96, NNH99, RSJM05]). As it better suits the results to follow, we state their approach using the duality principle of lattice theory, i.e., we use join-semilattices rather than meet-semilattices, deviating from the classical dataflow analysis literature such as [Kil73, KU77, SP81]. As a consequence, we also replace greatest fixed points by least fixed points, meet-over-all-paths by join-over-all-paths, etc. This change is purely notational (cf. Figure 1.2 for a short recap on the definition of semilattice and complete lattice, duality and fixed points).

Sharir and Pnueli assume that the complete lattice ([1]) $\langle L, \sqsubseteq \rangle$ of *values* is chosen in such a way that values, i.e., the elements of the lattice, capture the information one is interested in. Further, their approach requires a mapping ϕ assigning to every program instruction a value, and a concatenation operator $\cdot$ that, given the values a and b of two program instructions, returns the value $a \cdot b$ corresponding to their sequential execution $a; b$. Finally, they assume that the concatenation operator $\cdot$ distributes over the lattice's join $\sqcup$, i.e.,

[1] More precisely, in [SP81] meet-semilattices L are considered which have both a least and a greatest element, and, further, satisfy the ascending chain condition, i.e., every monotonically increasing sequence becomes stationary eventually. But the authors note themselves: "However, since we will assume that L is finite [...] in any practical application of this approach [...]". This means, Sharir and Pnueli consider finite, and thus complete lattices.

Let L be some set and $\sqsubseteq$ a partial order on L. Then $\langle L, \sqsubseteq\rangle$ is a *meet-semilattice*, resp. *join-semilattice* if for every two elements a, b of L their *greatest lower bound (meet)*, resp. their *least upper bound (join)* denoted by $a \sqcap b$, resp. $a \sqcup b$ exists in L. Every meet-semilattice $\langle L, \sqsubseteq\rangle$ is *dual* to the join-semilattice $\langle L, \sqsupseteq\rangle$ with $\sqsupseteq$ the reverse of $\sqsubseteq$. The semilattice is *complete* if the greatest lower bound $\sqcap A$, resp. least upper bound $\sqcup A$ of arbitrary subsets A of L exists. If a semilattice is complete, then meet, resp. join can be represented by its dual:

$$\sqcap A = \sqcup\{b \in L \mid \forall a \in A : b \sqsubseteq a\}, \text{ resp. } \sqcup A = \sqcap\{b \in L \mid \forall a \in A : a \sqsubseteq b\}.$$

A complete semilattice is therefore called *complete lattice*. In particular, every meet-, resp. join-semilattice $\langle L, \sqsubseteq\rangle$ with L finite is a complete lattice if it possesses a greatest element, resp. least element.
Obvious, but still important examples of (meet-)semilattices are the integers with the canonical order, i.e., $\langle \mathbb{Z}, \leq\rangle$, or the for any given set A the *powerset lattice* on A given by $\langle 2^A, \subseteq\rangle$ with 2^A the powerset of A. Their dual give natural examples of join-semilattices.
For any meet-semilattice $\langle L, \sqsubseteq\rangle$, let $f : L \to L$ be a map on L. We then have that the least fixed point of f exists w.r.t. $\langle L, \sqsubseteq\rangle$ if and only if the greatest fixed point of f exists w.r.t. to its dual $\langle L, \sqsupseteq\rangle$.

Figure 1.2: Lattice – basic definition and examples.

$a \cdot (b \sqcup c) = (a \cdot b) \sqcup (a \cdot c)$ ([2]). This is a restriction, as not all program analyses are distributive [NNH99], but we will also use this assumption for the largest part of this section. With this at hand, Sharir and Pnueli define a system of *abstract data flow equations*, containing one variable for each program point n:

> If n is the initial program point of a procedure then it contributes the equation $v_n = 1$, where v_n denotes n's variable. Otherwise, it contributes the equation
>
> $$v_n = \bigsqcup_{m \in pred(n)} v_m \cdot h(m, n)$$
>
> where $pred(n)$ denotes the set of immediate predecessors of n w.r.t. the flowgraph of the procedure, and $h(m, n)$ is defined as follows: if (m, n) is a call edge calling, e.g., procedure X, then $h(m, n)$ is the variable for the return node of X; otherwise $h(m, n) = \phi(m, n)$. See also the following Example 1.1.1.

They show that for every procedure P of the program and for every program

[2] Actually, in [SP81] the value of a program instruction is the function describing its effect on program variables, and the extension operator is function composition. However, the extension to an arbitrary distributive concatenation operator is unproblematic.

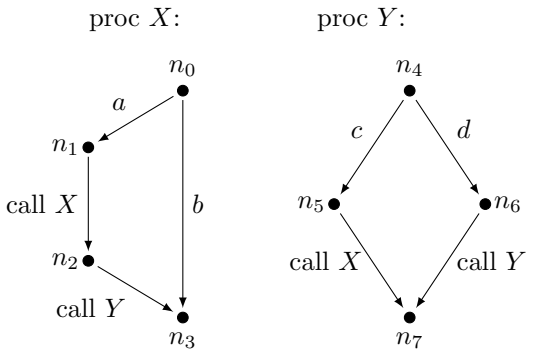


Figure 1.3: Flowgraphs of three procedures

point p of P, the least solution of the system is the join of the values of all valid program paths starting at the initial node of P and leading to p. (Sharir and Pnueli's result was later extended by [KS92] to programs with local variables.) This least solution is usually referred to as *meet on all paths* or short *MOP*. ([3])

Example 1.1.1. Consider a program consisting of two procedures X and Y whose (control) flow graphs are shown in Figure 1.3. Nodes n_i correspond to program points, and edges to program instructions. For instance, procedure X can execute an instruction with value $\phi(n_0, n_3) = b$ and terminate, or execute an instruction with value $\phi(n_0, n_1) = a$, call itself recursively, and, after the recursive call has terminated, call Y. The system of equations for Figure 1.3 can be represented more succinctly if variables for all program points other than return points are eliminated by substitution. Only two equations remain, namely those for the return point n_3 of procedure X, resp. n_7 of procedure Y. If moreover, and abusing language, we reuse X and Y to denote the variables for these points, and a, b, c, d to denote the values $\phi(n_0, n_1), \phi(n_0, n_3), \phi(n_4, n_5), \phi(n_4, n_6)$, respectively, we obtain the system

$$X = a \cdot X \cdot Y \sqcup b \qquad Y = c \cdot X \sqcup d \cdot Y \qquad (1.1.1)$$

Since the right-hand-sides of the equations are monotonic mappings, and $\cdot$ distributes over $\sqcup$, the existence of the least fixed point is guaranteed by Kleene's fixed-point theorem (see Theorem 2.2.12). ◇

We slightly generalize Sharir and Pnueli's setting. Loosely speaking, we allow to replace the join operator $\sqcup$ with any operator satisfying the same algebraic properties but possibly idempotence. In algebraic terms, we extend the framework from the class of lattices considered in [SP81] to an *ω-continuous semiring* [Kui97] (see also Definition 2.2.2), an algebraic structure with two operations, usually called sum and product. The interest of this otherwise

[3] By dualization, meet becomes join in our presentation.

simple extension is that our framework now encompasses equations over the semiring of the nonnegative reals with addition and multiplication. This allows us to compare the efficiency of generic solution methods for dataflow analysis when applied to the reals, with the efficiency of methods applied by numerical mathematics, in particular Newton's method.

It is well-known that Newton's method, when it converges to a solution, usually converges much faster than classical fixed-point iteration (see e.g. [OR70]). Furthermore, Etessami and Yannakakis have recently proved that Newton's method is guaranteed to converge for an analysis concerning the probability of termination of recursive programs [EY05]. These results were the motivation for our research into the question whether Newton's method can be generalized to the more abstract dataflow setting, where values are arbitrary entities, while preserving these good properties.

1.1.1 From Semilattices to Semirings

Let us examine the properties of the join operator $\sqcup$. First of all, since the lattice is complete, it is defined for arbitrary, finite or countably infinite, sets of lattice elements. Furthermore, it is associative, commutative, idempotent, and concatenation distributes over it. If we use the symbols 0 for the bottom element of the lattice (corresponding to an abort operation) and 1 for the element corresponding to a NOP instruction, then we have $0 \sqcup a = a \sqcup 0 = a$ and $1 \cdot a = a \cdot 1 = a$ for every a. It is argued in [SF00] that one can transform every program analysis to an essentially equivalent one that satisfies $0 \cdot a = a \cdot 0 = 0$. So the lattice, together with the two operations $\sqcup$ and $\cdot$ and the elements 0 and 1, constitutes an *idempotent semiring*. In the following we write '$+$' for '$\sqcup$' to conform with the standard semiring notation.

Idempotence of the join operator is not crucial for the existence of the least fixed point; it can be replaced by a weaker property. Consider the relation $\sqsubseteq$ on semiring elements defined as follows: $a \sqsubseteq a + b$ for all elements a, b. A semiring is *naturally ordered* if this relation is a partial order, and a naturally ordered semiring in which infinite sums exist and satisfy standard properties is called *ω-continuous*. Using Kleene's fixed-point theorem it is easy to show that systems of equations over ω-continuous semirings still have a least fixed point with respect to the partial order $\sqsubseteq$ (see Theorem 2.2.12 taken from [Kui97]).

We now study several examples encompassed by this more general setting. This gives us also the opportunity to introduce several of the questions stud-

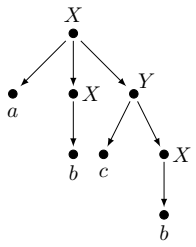
ied in this thesis, resp. related to these. For this recall that different interesting pieces of information correspond to the least solution of the dataflow equations when instantiated over different semirings. In the following we will denote by Σ the set of actions or operations the program consists of, e.g. $\Sigma = \{a, b, c, d\}$ for the program considered in Example 1.1.1, and let σ denote an arbitrary element of Σ.

Language interpretation: We start with an analysis which also shows the connection of our results to formal language theory. Consider the following semiring. The carrier is 2^{Σ^*} (i.e., the set of languages over Σ). The semiring element σ is interpreted as the singleton language $\{\sigma\}$. The sum and product operations are union and concatenation of languages, respectively. We call it the *language semiring* over Σ. Instantiating the abstract dataflow-equations on this semirings, the dataflow-equations merely become a context-free grammar, and their least solution corresponds to the context-free language represented by the grammar, see the following example.

Example 1.1.2. Under the language interpretation, Equations (1.1.1) become the following context-free grammar:

$$X \to aXY \mid b \qquad Y \to cX \mid dY$$

The least solution of (1.1.1) is then the pair $(L(X), L(Y))$, where, for $U \in \{X, Y\}$, $L(U)$ denotes the set of terminating executions of the program with U as main procedure, or, in language-theoretic terms, the language of the associated grammar with U as axiom. In particular, by means of this interpretation of the dataflow-equations as grammar, derivation trees, as known from formal language theory, can be associated with every system of polynomial equations. For example, the terminating run *abcb* corresponds to this derivation tree:



◇

As we will see in Chapter 3, when applying our generalized Newton's method to context-free grammars, the Newton approximants can be characterized by means of subsets of derivations trees associated with the context-free grammar under consideration. This in turn allows us to show a surprising connection between Newton's method, and the notion of *finite-index languages* which have been extensively investigated under different

names by Salomaa, Gruska, Yntema, Ginsburg and Spanier, among others [Sal69, Gru71, Ynt67, GS68].

Counting interpretation: Assume that an action $\sigma \in \Sigma$ corresponds either to an allocation or a deallocation of some resource, and we want to check that the number of allocations and deallocations match. The problem we are facing is therefore to count how many as, bs, etc. we can observe in a (terminating) execution of the program, but we are not interested in the order in which they occur. In the terminology of abstract interpretation, we abstract an execution $w \in \Sigma^*$ by the vector $(n_a, n_b, n_c, n_d) \in \mathbb{N}^{|\Sigma|}$ where $n_a, \ldots, n_d$ are the number of occurrences of $a, \ldots, d$ in w. We call $(n_a, \ldots, n_d)$ the *Parikh image* of w. In order to obtain the Parikh images of the sets of terminating runs, we then instantiate the dataflow-equations on the following semiring: The carrier is $2^{\mathbb{N}^{|\Sigma|}}$, where the j-th action of Σ is interpreted as the singleton set $\{(0, \ldots, 0, 1, 0 \ldots, 0)\}$ with the "1" at the j-th position. The sum operation is set union, and the product operation is given by

$$S \cdot T = \{(s_a + t_a, \ldots, s_i + t_i) \mid (s_a, \ldots, s_i) \in S, (t_a, \ldots, t_i) \in T\} \ .$$

As we have seen, the sets of terminating runs correspond to context-free languages. By virtue of the well-known result by Parikh there exist regular languages whose Parikh image coincides with the ones of the sets of terminating runs. See the following example.

Example 1.1.3. We assume the natural lexicographical order on Σ, i.e., we identify a with the set $\{(1,0,0,0)\}$, and so forth. Our running example then becomes the following equations:

$$X = \{(1,0,0,0)\} \cdot X \cdot Y \cup \{(0,1,0,0)\} \qquad Y = \{(0,0,1,0)\} \cdot X \cup \{(0,0,0,1)\} \cdot Y$$

with multiplication defined as stated above. As we will see, our generalized Newton's method reaches the least solution of this system after already two steps (cf. Theorem 3.4.6). From this result we obtain that the languages represented by the following regular expressions have the same Parikh image as the least solution of the above equation system:

$$X = \left(ad^*cb + a(abd^*c)^*bd^*c\right)^* b \qquad Y = \left(d + c(ad^*cb)^*a(abd^*c)^*b\right)^* cb \qquad \diamond$$

We show in Section 3.4 that Newton's method always reaches the least solution of a polynomial equation system under the counting interpretation. In particular, we show that Newton's method reaches the least solution in at most n steps where n is the number of variables the polynomial system consists of (see Theorem 3.4.6), and that every Newton approximant is given by a regular expression, thus obtaining a new, constructive proof of Parikh's

theorem. Further, we identify a method proposed by Hopkins and Kozen in [HK99] as Newton's method, thereby improving the upper bound of $O(3^n)$ given in [HK99] on the number of steps needed to reach the least solution.

Throughput: Assume that with each of the actions of Σ some measure of the work done by them is associated, for simplicity let this be a function $w : \Sigma \to \mathbb{N}$ such that $w(\sigma)$ is the work done by action σ in some fixed unit of measurement. Further, assume that every action needs exactly one time unit to execute. Given a run $\sigma_1\sigma_2 \dots \sigma_l$ of a program we then may define the throughput of this run to be

$$\frac{\sum_{i=1}^{l} w(\sigma_i)}{l}.$$

Caucal et al. consider in [CCFR07] the problem of determining the throughput of a program, i.e., the greatest lower bound on the throughputs of all runs. In our terms, the algorithms of [CCFR07] obtains this throughput by repeatedly adapting the function w and instantiating the abstract dataflow equations on the $(\min, +)$-semiring with the reals extended by $\pm\infty$ as carrier, minimum as addition, and plus as multiplication. We refer the reader to Subsection 4.3.2 for more details. There we also show how to improve the algorithm of [CCFR07].

Example 1.1.4. We consider the easier, yet underlying problem of calculating the infimum on the work done along the executions of the program, i.e., we want to calculate for $U \in \{X, Y\}$

$$W_U := \inf \left\{ \sum_{i=1}^{l} w(\sigma_i) \mid \sigma_1 \dots \sigma_l \text{ is a terminating run of procedure } U \right\}.$$

Then the vector (W_X, W_Y) is the least solution of the dataflow equations 1.1.1 instantiated on the $(\min, +)$-semiring:

$$X = \min\{w(a) + X + Y, w(b)\} \qquad Y = \min\{w(c) + X, w(d) + Y\}$$

Assuming that all weights $w(a), \dots, w(d)$ are nonnegative, one easily realizes that the least solution of this system is simply

$$X = w(b) \qquad Y = w(b) + w(c)$$

as the final action in every terminating run is b. ◇

Regarding the more general setting where w takes also negative values, we show in Chapter 4 how polynomial systems on so called *star-distributive semirings*, which encompass the $(\min, +)$-semiring, can be reduced to a linear system without changing the least solution. The least solution of such a linear system can then be easily calculated:

Example 1.1.5. In Example 1.1.2 we have already introduced the idea that a polynomial system corresponds to a context-free grammar when addition is idempotent. Consider the following linear system on the $(\min, +)$-semiring:

$$\begin{array}{rcl} X & = & \min\{1, 2 + X, -1 + Y\} \\ Y & = & \min\{1 + X, -1 + Y\}. \end{array}$$

Because semiring multiplication $(+)$ is commutative, this linear system corresponds to the following regular grammar (without explicit axiom):

$$\begin{array}{rcl} X & \to & a \mid bX \mid cY \\ Y & \to & aX \mid cY \end{array}$$

Formally, we use the morphism h from the language semiring generated by $\Sigma = \{a, b, c\}$ into the $(\min, +)$-semiring uniquely determined by $h(a) = 1$, $h(b) = 2$, and $h(c) = -1$.

From this regular grammar we now can easily deduce regular expressions ϕ_X, resp. ϕ_Y representing the language we obtain when taking X, resp. Y as axiom:

$$\phi_X = (b + cc^*a)^*a \text{ resp. } \phi_Y = (ab^*c + c)^*aa.$$

By means of the morphism h one can now obtain from these the least solution of the original equation system on the $(\min, +)$-semiring ([4]). Let us explicitly calculate the image of b^* and c^* under h:

$$\begin{array}{rcccccl} h\big(\bigcup_{k\geq 0}\{b^k\}\big) & = & \inf\{h(b)\cdot k \mid k \geq 0\} & = & \inf\{2\cdot k \mid k \geq 0\} & = & 0 \\ h\big(\bigcup_{k\geq 0}\{c^k\}\big) & = & \inf\{h(c)\cdot k \mid k \geq 0\} & = & \inf\{-k \mid k \geq 0\} & = & -\infty. \end{array}$$

In this way one can calculate that the least solution of the original equation system is given by $X = Y = -\infty$. The reader might want to check this result by applying the standard fixed point iteration to the right-hand side starting from (∞, ∞). ◇

Probabilistic interpretations: Assume that the choices between actions are stochastic. For instance, actions a and b are chosen with probability p and $(1-p)$, respectively. The probability of termination, i.e., the probability that a given procedure eventually terminates, is given by the least solution of Equation (1.1.1) when interpreted over the *real semiring* (see [EKM04, EY05]): The carrier is the set of nonnegative real numbers $\mathbb{R}_{\geq 0}$ extended by ∞; semiring addition and multiplication result from suitably extending the canonical addition and multiplication on the reals to $\mathbb{R}_{\geq 0} \cup \{\infty\}$. The semiring element σ is interpreted as the probability of choosing σ among all enabled actions.

Example 1.1.6. A particular instantiation of the program of Figure 1.3 as an probabilistic program is shown in Figure 1.4. The semiring operations are addition and multiplication

[4] One can show that h preserves the least fixed point.

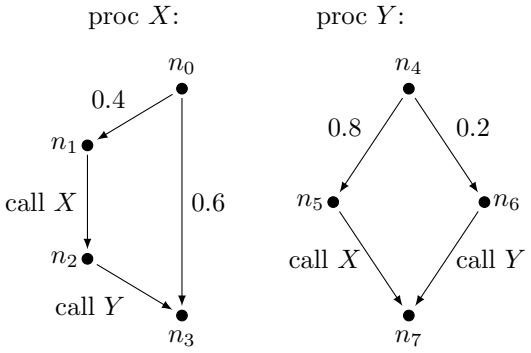


Figure 1.4: Probabilistic flowgraphs

over the nonnegative reals. Notice that addition is not idempotent. The semiring is ω-continuous if an ∞-element with the usual properties is added. As just stated, the least solution of the system

$$X = 0.4 \cdot X \cdot Y + 0.6 \qquad Y = 0.8 \cdot X + 0.2 \cdot Y$$

is the vector of termination probabilities. In our example, we can easily obtain all solutions as the second equation is linear, in particular, we can transform it to $Y = X$. We thus may substitute X for Y in the first equation, leading to the equation $0 = 0.4 \cdot X^2 - X + 0.6$ with solutions $X = 1$ and $X = 1.5$. The least solution is thus $X = 1, Y = 1$, i.e., both procedures terminate with probability 1. ◇

Of course, in general neither calculating termination probabilities can be reduced to solving an univariate problem nor do the termination probabilities need to be rational, let alone representable by radicals. For the latter consider the following example:

Example 1.1.7. . Consider the equation $X = 1/6\,X^6 + 1/2\,X^5 + 1/3$. The solutions of this equation are exactly the roots of the polynomial $p(X) = 1/6\,X^6 + 1/2\,X^5 - X + 1/3$. The polynomial p is reducible, and can be written as $p(X) = 1/6 \cdot (X - 1) \cdot q(X)$ with $q(X) = X^5 + 4\,X^4 + 4\,X^3 + 4\,X^2 + 4\,X - 2$. Obviously, the greatest common divisor of the coefficients of q is one, i.e., q is a primitive polynomial in $\mathbb{Q}[X]$. By Eisenstein's criterion it then immediately follows that q is also irreducible in $\mathbb{Q}[X]$, i.e., it cannot be written as the product of two non-constant polynomials of $\mathbb{Q}[X]$. The roots of its derivative $\frac{d}{dX}q$ can be exactly determined by means of the formula by Ferrari. From this one sees that q has exactly two inflection points, and one can argue that q has exactly three roots in $\mathbb{R}$. Using Galois theory (see e.g. theorem 10 in section 6.1 of [Bos01]) it then follows that the roots of q cannot be represented by means of radicals. ◇

As calculating the probabilities of termination is not possible in general, the question of the convergence speed of Newton's method for these kind of systems naturally arises. This means, how many approximation steps does one need to do such that the current approximant coincides with the exact

solution in at least the k highest bits ([5]). This thesis is only marginally concerned with these important problems; only in Chapter 6 we study some properties of polynomial systems on the nonnegative reals. We refer the reader to [KLE07, EKL08a, EKL09a] for a detailed treatment, and only illustrate one of the main results regarding the convergence speed of Newton's method in the following example:

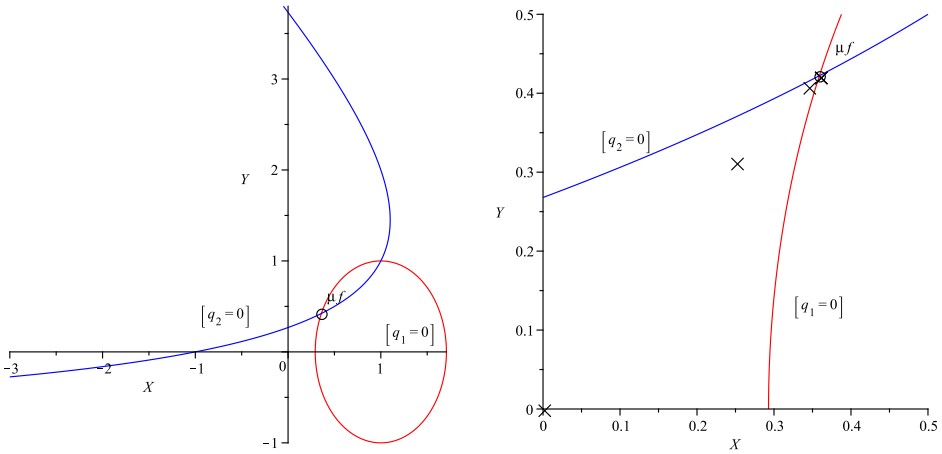
Example 1.1.8. Consider the following system of polynomials on the nonnegative reals:

$$\begin{array}{rcll} X & = & \boldsymbol{f}_1(X,Y) & := & \frac{1}{2}X^2 + \frac{1}{4}Y^2 + \frac{1}{4} \\ Y & = & \boldsymbol{f}_2(X,Y) & := & \frac{1}{4}X + \frac{1}{4}XY + \frac{1}{4}Y^2 + \frac{1}{4}. \end{array}$$

We may also write this as

$$\begin{array}{rcll} \boldsymbol{q}_1(X,Y) & := & \frac{1}{2}X^2 + \frac{1}{4}Y^2 + \frac{1}{4} - X & = & 0 \\ \boldsymbol{q}_2(X,Y) & := & \frac{1}{4}X + \frac{1}{4}XY + \frac{1}{4}Y^2 + \frac{1}{4} & = & 0. \end{array}$$

Let $[\boldsymbol{q}_i = 0]$ denote the set of zeros of the equation $\boldsymbol{q}_i(X,Y) = 0$. Then $[\boldsymbol{q}_i = 0]$ is an implicit surface, in our example a quadric, and the least nonnegative solution $\mu\boldsymbol{f}$ of the above equation system is the least nonnegative point common to both surfaces:



(a) (b)

Figure 1.5: The surfaces of Example 1.1.8.

In our examples, $\boldsymbol{q}_1$ defines a parabola, and $\boldsymbol{q}_2$ corresponds to a ellipse. The reader can guess this by looking at Figure 1.5(a). One can show that the Newton approximants are all located in the region

$$R := \{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \mid \boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x}) \wedge \boldsymbol{x} \leq \mu\boldsymbol{f}\},$$

and that R is exactly the region enclosed by the coordinate axes and the surfaces. See for example Figure 1.5(b), where the Newton approximants of above equation system are depicted by crosses.

[5] Assuming that, for instance, $1/2$ is written as 0.1, and not as $0.0111\ldots = 0.0\bar{1}$.

The convergence speed of Newton's method is then essentially determined by the angle between the tangents at the quadrics in $\mu\boldsymbol{f}$ ([6]): the narrower the angle the slower Newton's method converges. An intuitive understanding of this can be obtained by considering another method of approximating $\mu\boldsymbol{f}$:
Looking at Figure 1.5(b), assume we are given some point $\boldsymbol{x}$ inside the region enclosed by the coordinate axes and the quadric surfaces. We then may move from $\boldsymbol{x}$ to the quadric defined by $\boldsymbol{q}_i$ by following the ray $\boldsymbol{x} + \mathbb{R}_{\geq 0} \cdot \boldsymbol{e}^i$ (with $\boldsymbol{e}^i_j = 0$ for $i \neq j$ and $\boldsymbol{e}^i_i = 1$) yielding the point $\boldsymbol{p}_i$; and take the tangent in $\boldsymbol{p}_i$ at the quadric as an approximation of the quadric of itself. This is shown in Figure 1.6.

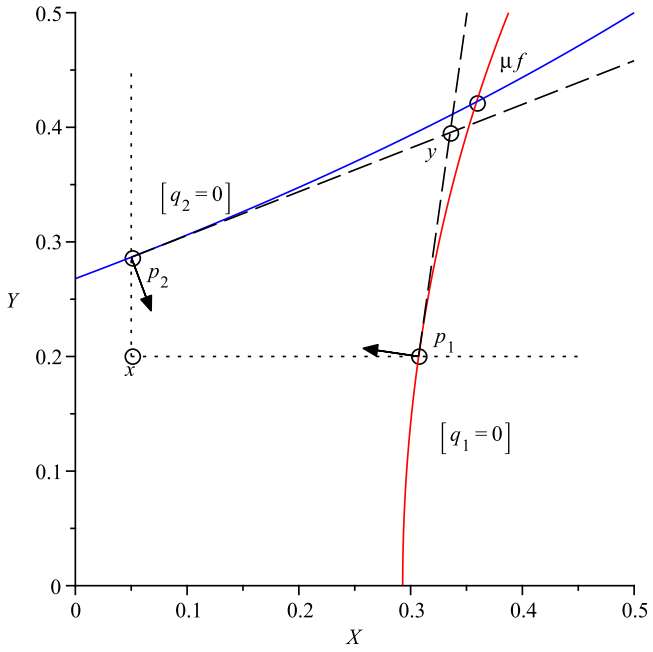
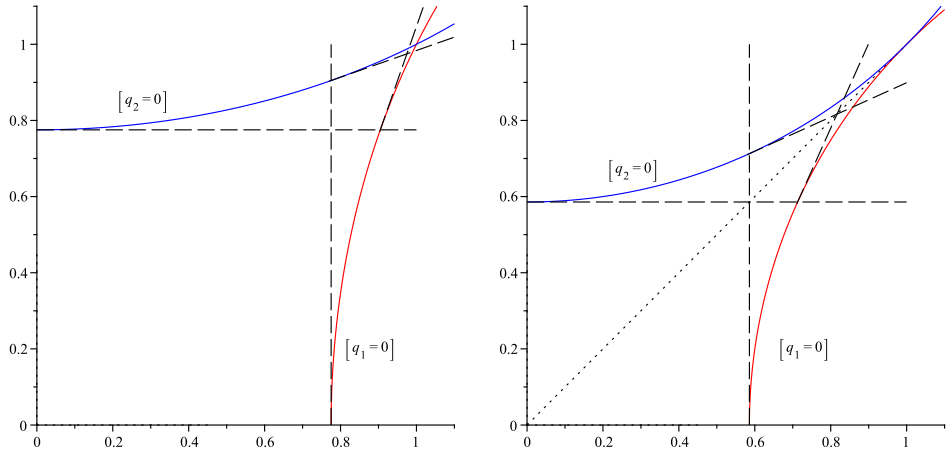


Figure 1.6

Figure 1.6 then suggests to take the intersection $\boldsymbol{y}$ of the tangents as the next approximant. Let us call this the *tangent method*. One can show that the tangent method always converges to $\mu\boldsymbol{f}$, and it does so at least as fast as Newton. We discuss the connection between Newton's method and this tangent method in more detail in Chapter 6. The advantage of the tangent method is that one has an intuitive understanding of its behavior when looking at the surfaces.

Figure 1.7 shows two steps of the tangent method when starting in $\mathbf{0} = (0, 0)$. In Figure 1.7(a), the tangents at $\mu\boldsymbol{f}$ intersect, whereas in Figure 1.7(b) they only touch with

[6] When considering the convergence speed, one can reduce the system $\boldsymbol{f}$ to subsystem where $\mu\boldsymbol{f}$ is positive in every component, and "every variable depends on every other variable", which basically means that if we improve an approximation of $\mu\boldsymbol{f}$ in one component, we can increase it in all components. For this class of systems, the characterization by means of the angle holds true. Our example satisfies these properties.



(a) (b)

Figure 1.7

the common tangent drawn by dots. When comparing the second pair of tangents used in Figure 1.7(a) to the pair used in Figure 1.7(b), one sees that in (b) the surfaces bend away from the tangents stronger than in (a). In other words, the second pair of tangents in (a) approximates the surfaces better than the second pair in (b) does. From this, it is not surprising that the convergence speed depends on the angle of the tangents in $\mu\boldsymbol{f}$; the smaller this angle gets relative to the initial angle enclosed by the tangents in $\mathbf{0}$, the closer we have to get to $\mu\boldsymbol{f}$ in order for the tangents to approximate the surfaces well. We therefore should expect the worst convergence speed of the tangent method, and Newton's method, if the surfaces are only tangent to each other in $\mu\boldsymbol{f}$.

In fact, as long as the surfaces do not touch in $\mu\boldsymbol{f}$, one can show that eventually Newton's method converges exponentially. This means that there is some *threshold*, i.e., a natural number $k_{\boldsymbol{f}}$ determined by the coefficients of $\boldsymbol{f}$ such that the $k_{\boldsymbol{f}}+i$-th Newton approximant coincides with $\mu\boldsymbol{f}$ in at least the 2^i highest bits.

Somewhat surprisingly, one can show that even in the worst case, i.e., both surfaces being tangent to each other in $\mu\boldsymbol{f}$, Newton's method, and the tangent method, eventually converge at least linearly. That is, there is again a constant $k_{\boldsymbol{f}}$ such that the $k_{\boldsymbol{f}} + i$-th approximant coincides with $\mu\boldsymbol{f}$ in the i highest bits. In particular in [KLE07] the following result is shown:
Let $\boldsymbol{z}$ be the first intersection with one of the coordinate axes of the ray starting in $\mu\boldsymbol{f}$ heading along the tangent towards the origin. Then for every point $\boldsymbol{x}$ which is at least as great as $\boldsymbol{z}$ in every component, Newton's method converges linearly when started in $\boldsymbol{x}$. We therefore have $k_{\boldsymbol{f}} = 0$ for the system depicted in Figure 1.7(b), i.e., Newton's method converges linearly right from the start. ◇

Up to now we have considered sequential programs where we assumed that for every branching point of the given program we know, or at least have

obtained an approximation of the probability with which a given branch is taken.

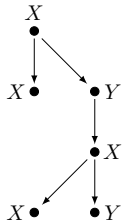
We conclude this paragraph by taking a detour to parallel programs. Here a surprising connection exists between the speed at which Newton's method converges to the least solution, and the space-efficient scheduling of parallel programs (cf. [BEKL09]):

Example 1.1.9. Assume we are given a parallel program which is made up of two types of tasks. Let X and Y denote these two types. For simplicity, every task takes one time unit to complete and spawns at its end a finite number of child tasks. For this example, assume that a task of type X either terminates with probability 0.6 without spawning any children, or it terminates with probability 0.4 spawning two new tasks, one of type X, and one of type Y. Similarly, a task of type Y – after finishing its workload – either spawns a task of type X with probability 0.8, or a task of type Y with probability 0.2. We may succinctly represent the described behavior as a stochastic context-free grammar:

$$\begin{array}{llll} X & \xrightarrow{0.6} & \varepsilon & \qquad\qquad Y \xrightarrow{0.8} X \\ X & \xrightarrow{0.4} & XY & \qquad\qquad Y \xrightarrow{0.2} Y. \end{array}$$

Suppose we are given an initial task of type X. Then again we want to know if this task will eventually terminate almost surely. Let t_X (t_Y) be the probability that a single task of type X (Y) eventually terminates. It is not hard to see that the vector (t_X, t_Y) is then the least solution of the equation system of Example 1.1.6. We therefore know that in our example both task types terminate almost surely.

We turn to the question of scheduling. For this example, we assume that all tasks are executed on a single CPU, i.e., in every time unit only a single task can be executed, and all tasks are assumed to be executable independently of each other, so the CPU is free to choose which task to execute next. Starting from an initial tasks of type, say X, after each time step the CPU adds the newly spawned children to the pool of tasks waiting for execution, and chooses then from this pool by some strategy the next task to be executed. For instance, consider the following run of the described program:



This means the initial task of type X spawns two children, one of type X, one of type Y. The child of type X then terminates, whereas the child of type Y again spawns a child of type X, and so on. Consider now the strategy where the CPU first completes the tasks originating from the first task of type Y. In this case, we will have to store up to three tasks waiting for execution. On the other hand, if the CPU first schedules the tasks of type X, then a at most two tasks have to be remembered along the execution of the parallel program.

Naturally the question arises if there is some optimal strategy for the CPU to schedule the tasks in order to minimize the number of tasks waiting for execution at any given point of time, i.e., in order to minimize the needed space. As we will see in Theorem 3.2.11, the Newton approximants correspond to a particular class of derivations trees (w.r.t. to the interpretation of the dataflow equations as context-free grammar, motivated in Example 1.1.2). This class of trees correspond to the most space-efficient scheduling. From this, one can obtain the surprising result that when using the optimal scheduling strategy, then the probability that at most k units of space are used is equal to the k-th Newton approximant (cf. [BEKL09]). ◇

1.2 Solving Systems of Equations

Current generic algorithms for solving Sharir and Pnueli's equations, like the classical worklist algorithm of dataflow analysis, are based on variants of Kleene's fixed-point theorem [Kui97] (cf. Theorem 2.2.12). The theorem states that the least solution $\mu\boldsymbol{f}$ of a system of equations $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ over an ω-continuous semiring is equal to the supremum of the sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ of *Kleene approximants* given by $\boldsymbol{\kappa}^{(0)} = \vec{0}$ and $\boldsymbol{\kappa}^{(i+1)} = \vec{f}(\boldsymbol{\kappa}^{(i)})$. This yields a procedure (let us call it *Kleene's method*) to compute or at least approximate $\mu\boldsymbol{f}$. If the domain satisfies the well-known *ascending chain condition* [NNH99], then the procedure terminates, because there exists an i such that $\boldsymbol{\kappa}^{(i)} = \boldsymbol{\kappa}^{(i+1)} = \mu\boldsymbol{f}$.

Kleene's method is generic and robust: it always converges when started at the vector $\mathbf{0}$ of 0-elements, for any ω-continuous semiring and for any system of equations. On the other hand, it often fails to terminate, and it can converge very slowly to the solution. We illustrate this point by means of two simple examples. Consider the equation $X = a \cdot X + b$ over the lattice of subsets of the language $\{a, b\}^*$. The least solution is the regular language a^*b, but we have $\boldsymbol{\kappa}^{(i)} = \{b, ab, \ldots, a^{i-1}b\}$, i.e., the solution is not reached in any finite number of steps. For our second example consider a very simple probabilistic procedure that can either terminate or call itself twice, both with probability 1/2. The probability of termination of this program is given by the least solution of the equation $X = 1/2 + 1/2X^2$. It is easy to see that the least solution is equal to 1, but we have $\kappa^{(i)} \leq 1 - \frac{1}{i+1}$ for every $i \geq 0$, i.e., in order to approximate the solution within i bits of precision we have to compute about 2^i Kleene approximants. For instance, we have $\kappa^{(200)} = 0.9990$, i.e., 200 iterations produce only three digits of precision.

After our slight generalization of Sharir and Pnueli's framework, quantitative analyses like the probability of termination fall within the scope of the

approach. So we can look at numerical mathematics for help with the inefficiencies of Kleene's method.

As could be expected, faster approximation techniques for equations over the reals 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 to approximate a zero of a differentiable function, and can be adapted to our problem. Since the least solution of $X = 1/2 + 1/2X^2$ is a zero of $1/2+1/2X^2-X$, the method can be applied, and it yields $\nu^{(i)} = 1-2^{-i}$ for the i-th *Newton approximant*. So the i-th Newton approximant already has i bits of precision, instead of $\log i$ bits for the Kleene approximant.

However, Newton's method also has a number of disadvantages, at least at first sight. Newton's method on the real field is by far not as robust and well behaved as Kleene's method on semirings. The method may converge very slowly, converge only locally (only when started in a small neighborhood of the zero), or even not converge at all [OR70]. So we face the following situation. Kleene's method, a robust and general solution technique for arbitrary ω-continuous semirings, is inefficient in many cases. Newton's method is usually very efficient, but it is only defined for the real field, and it is not robust.

Motivated by their work on Recursive Markov Chains, [EY05] showed that a variant of Newton's method is robust for certain systems of equations over the *real semiring*: the method always converges when started at zero. In other words, moving from the real field to the real semiring (only nonnegative numbers) makes the instability problems disappear.

Naturally the question arises if Newton's method can be lifted from the real semiring to the general class of ω-continuous semirings. This question was the starting point for this thesis and the results obtained therein.

1.3 Contribution and Related Work

Chapter 3 studies the question if and how Newton's method can be extended to arbitrary ω-continuous semirings. Starting from the original definition of Newton's method on the reals, we generalize it step-by-step and prove that its robustness on the real semiring is preserved by the generalization. This means that the generalized Newton's method always converges to the least solution, and that it always does so at least as fast as standard fixed point iteration. We then proceed to further analyze our generalized Newton's method. We

provide a characterization of the Newton approximants, amongst others by means of derivation trees as known from formal language theory, and apply it to idempotent semirings, the structures of classical program analysis. We first study the language semiring, where equation variables are interpreted over languages of finite words, sum is interpreted as union of languages, and product as concatenation. The least solutions of fixed-point equations are the context-free languages, and so our generalized Newton's method can be seen as a tool for approximating context-free languages. We show that the Newton approximants are the context-free languages *of finite index*, a well-known class studied since the 1960s in language theory [Ynt67, GS68, Sal69, Gru71]. We then proceed to study the case of commutative and idempotent semirings. Loosely speaking, these semirings correspond to *counting analysis*, in which one is interested in how often program points are visited, but not in which order. These semirings do not always satisfy the ascending chain condition, and Kleene's method may not terminate. We show that a very elegant iterative solution method for these semirings, obtained by [HK99], is exactly Newton's method, and always terminates in a finite number of steps. As mentioned above, we further use our characterization of Newton approximants to show that the least fixed point is reached after at most n iterations, a tight bound, improving on the $\mathcal{O}(3^n)$ bound of [HK99]. Chapter 3 consists of material previously published in [EKL07b, EKL07a, EKL09b].

Chapter 4 is based on the results published in [EKL08b], and is devoted to the proof principle used for showing the convergence speed of Newton's method in case of commutative and idempotent semirings, and the characterization of the Newton approximants by means of derivation trees. We generalize this principle, and prove its usefulness by identifying several classes of semirings which allow for an efficient calculation of the least fixed point of polynomial systems. We introduce *star-distributive* semirings, a subclass of semirings with idempotent addition and commutative multiplication including, for instance, the semiring on the reals with min as addition, and + as multiplication – the semiring underlying shortest-path problems. We then show that for any non-linear polynomial system on star-distributive semirings, one can efficiently construct a *linear* polynomial system preserving the least solution – the latter a well-known and solvable problem. Our result on star-distributive semirings is then used to improve an algorithm by Caucal et al. [CCFR07]. Further, we study *lossy semirings*, a subclass of semirings with idempotent addition, but not necessarily commutative addition. This subclass of semirings contains e.g. the semiring used in [BEM97] for modeling lossy channel systems, i.e., systems which communicate via channel which are not reliable and therefore can lose parts of the messages. We show that sim-

ilar to star-distributive semirings solving non-linear polynomial system can be reduced to solving linear systems. This allows us to show that the least solutiontion of polynomial systems on lossy semirings can also be computed, and not only approximated.

In Chapter 5 we consider semirings whose natural order is total. On such semirings, both join and meet always are defined, i.e., one can say that such semirings exhibit two additions. It is thus natural to consider the extension of polynomials built up from both kind of additions, and multiplication. A natural example of such semirings are integers where join and meet become max and min, and multiplication is given by addition on the integers. We therefore call systems consisting of polynomials using both meet and join *(polynomial) min-max-systems*. Min-max-systems on the integers arise e.g. in interval analysis introduced by Cousot and Cousot [CC76, CC91] ([7]). As a direct consequence of our result on star-distributive semirings, we obtain in Section 5.2 that on these semirings the least solution of min-max systems can be obtained by solving a linear polynomial system using only the join operator. In Section 5.3 we study *strategy iteration*, a well-known approach for solving min-max-systems. There are many different algorithms based on strategy iteration, for instance, by Gawlitza and Seidl for interval analysis [GS07, GS08], by Jurdzinski and Vöge for parity games [VJ00], or by Björklund, Sandberg, and Vorobyov for mean-payoff and parity games [BSV02, BSV03, BSV04]. We identify a class of semirings, encompassing the special cases just mentioned, for which strategy iteration always allows to obtain the least solution of a polynomial min-max-system. As an application of our results, we show in Section 5.5 how our results can be applied to parity games. This chapter is loosely based on [Lut08].

Finally, in Chapter 6 we study some properties of polynomials systems on the nonnegative reals. We give a characterization of the region the Newton approximants are located in. From this characterization, we obtain a new method, generalizing Newton's method, for approximating the least solution (see Example 1.1.8). We also study the existence of a second nonnegative fixed point, a question motivated by the study of Galton-Watson processes [WG75].

[7] The goal in this analysis is to obtain for each variable and each program point of a given program intervals as tight as possible bounding the range of the variable.

1.4 Outline

This thesis is organized as follows. Section 2.2 introduces ω-continuous semirings, systems of fixed-point equations, and some semirings investigated in the rest of the paper. Section 3.1 recalls Newton's method, and generalizes it to arbitrary ω-continuous semirings. Section 3.2 characterizes the Newton approximants in terms of derivation trees, a generalization of the derivation trees of language theory. Section 3.3 considers the particular case of idempotent semirings and applies the characterization to the language semiring. Section 3.4 applies the characterization to idempotent and commutative semirings. Finally, Section 3.5 shows that Newton's method can also be applied to non-distributive program analyses.

In Chapter 4, we turn on to applying the proof principle underlying the Sections 3.3 and 3.4 to more specialized classes of semirings, thereby obtaining more efficient methods for calculating the least solution of polynomial systems. In particular, we introduce and study star-distributive semirings (Section 4.3, lossy semirings (Section 4.4), and 1-bounded semirings (Section 4.5).

Section 5.2 studies the special case of star-distributive semirings whose natural order is total, and the extension of polynomial equation systems to min-max-systems. From there we move on to study when strategy iteration can be used for solving min-max-systems, leading to the definition of si-semirings and nondeterministic strategy iteration (Section 5.3. In Section 5.5 we then exemplify our results by applying them to parity games.

The tangent method as described in Example 1.1.8 is discussed in Section 6.3. Section 6.4 contains a treatment of the problem when a system of polynomials on the nonnegative reals possesses a second finite fixed point.

Some of the more technical proofs have been moved to the appendix. Instead proof sketches are given in the hope of improving readability.

Chapter 2

Preliminaries

This chapters introduces definitions and notations used throughout the following chapters. We first recall standard definitions from literature and then introduce more specific definitions about semirings.

2.1 Basic Definitions and Notations

Logical Operators: We use $\wedge, \vee, \Rightarrow, \Leftarrow, \Leftrightarrow, \neg$ to denote the standard logical operators.

Sets: We use $\mathbb{N}$ to denote the set of natural numbers, and assume $0 \in \mathbb{N}$. For a natural number $n \geq 1$ we write $[n]$ for $\{1, 2, \ldots, n\}$. The real numbers are denoted by $\mathbb{R}$, and we refer to the nonnegative real numbers by $\mathbb{R}_{\geq 0}$. Operations on sets are denoted as usual by $\cup$ (set union), $\cap$ (set intersection) and $\setminus$ (set difference). Let A be a set. We write $|A|$ for the cardinality of A. Its power set is denoted by 2^A. For B another set, we write A^B for the set of all functions from B to A. The value of a function $f \in A^B$ at some $b \in B$ is denoted either by $f(b)$ or f_b. In particular, for $k \in \mathbb{N}$ we call A^k the set of sequences or words of length k on A where ε denotes the sequence of length 0, i.e., $A^0 = \{\varepsilon\}$. The set of finite sequences on A is denoted by $A^* := \bigcup_{k \in \mathbb{N}} A^k$. Instead of $A^{\mathbb{N}}$ we stick to the more frequently used notation A^ω for denoting the set of countably infinite sequences or words on A.

Regular Expressions: The regular expressions generated by a finite alphabet Σ are denoted by RExp_Σ and defined inductively as usual for $0, 1 \notin \Sigma$:

$$\begin{array}{lcl} \mathsf{RExp}_\Sigma^0 & := & \Sigma \cup \{0,1\} \\ \mathsf{RExp}_\Sigma^{i+1} & := & \mathsf{RExp}_\Sigma^i \\ & \cup & \{\phi\psi, (\phi+\psi), (\phi)^* \mid \phi, \psi \in \mathsf{RExp}_\Sigma^i\} \\ \mathsf{RExp}_\Sigma & := & \bigcup_{i\in\mathbb{N}} \mathsf{RExp}_\Sigma^i. \end{array}$$

We assume that the Kleene star has the highest priority, followed by concatenation, and set union having the lowest priority, and drop parenthesis if no ambiguity arises. The canonical interpretation of a regular expression $\phi \in \mathsf{RExp}_\Sigma$ as language is denoted by $L_\Sigma(\phi)$:

$$\begin{array}{lllll} L_\Sigma : & \mathsf{RExp}_\Sigma & \to & 2^{\Sigma^*} \\ & 0 & \mapsto & \emptyset \\ & 1 & \mapsto & \{\varepsilon\} \\ & \Sigma \ni a & \mapsto & \{a\} \\ & \phi\psi & \mapsto & L_\Sigma(\phi) \cdot L_\Sigma(\psi) \\ & (\phi+\psi) & \mapsto & L_\Sigma(\phi) \cup L_\Sigma(\psi) \\ & (\phi)^* & \mapsto & L_\Sigma(\phi)^*, \end{array}$$

with $L_\Sigma(\phi) \cdot L_\Sigma(\psi)$ the concatenation of languages, i.e.,

$$L_\Sigma(\phi) \cdot L_\Sigma(\psi) := \{uv \mid u \in L_\Sigma(\phi), v \in L_\Sigma(\psi)\}$$

and $L_\Sigma(\phi)^* = \bigcup_{k\in\mathbb{N}} L_\Sigma(\phi)^k$. We drop the subscript, and simply write $L(\phi)$ if Σ is known from the context.

Binary Relations: Let A and B be sets. Then $R \subseteq A \times B$ is *binary relation*. We also write aRb for $(a,b) \in R$. The set $\{b \in B \mid aRb\}$ of *successors* of a w.r.t. R is denoted by aR. Analogously, we write Rb for the set $\{a \in A \mid aRb\}$ of *predecessors of* b. A binary relation $R \subseteq A \times A$ is a partial order if it is reflexive ($\forall a \in A : aRa$), antisymmetrical ($\forall a, b, c \in A : (aRb \wedge bRa) \Rightarrow a = b$) and transitive ($\forall a, b, c \in A : (aRb \wedge bRc) \Rightarrow aRc$). A partial order $R \subseteq A$ is a total order if $\forall a, b \in A : aRb \vee bRa$.

Partial Orders: A partial order or *poset* is a pair $\langle A, \leq\rangle$ with A a set, and $\leq$ a partial order on A. As usual we write $a \geq b$ for $b \leq a$, and $a < b$ for $a \leq b \wedge a \neq b$. An *ω-chain* in $\langle A, \leq\rangle$ is any ascending sequence $(a_i)_{i\in\mathbb{N}} \in A^\omega$, i.e., $a_i \leq a_{i+1}$ for all $i \in \mathbb{N}$. We call $\langle A, \leq\rangle$ *ω-chain complete* if the supremum of any ω-chain $(a_i)_{i\in\mathbb{N}}$ exists in A. Sometimes ω-chains are also defined

to be countable subsets of A on which $\leq$ is total. This latter definition obviously encompasses the former. The following proposition shows that both definitions are equivalent w.r.t. the notion of ω-chain completeness.

Proposition 2.1.1.
Let $\langle A, \leq \rangle$ be a ω-chain complete poset, and $B \subseteq A$ a countable set such that $\leq$ restricted to B is total. Then the supremum of B exists in A. $\diamond$.

Proof. As B is countable, there is a bijection $\beta : \mathbb{N} \to B$. We define the sequence $(i_k)_{k\in\mathbb{N}}$ inductively. We set $i_0 := 0$ for $i = 0$. For $k \geq 0$ we first define $I_k := \{j > i_k \mid \beta(j) > \beta(i_k)\}$. Then we set $i_{k+1} := i_k$ if $I_k = \emptyset$; otherwise $i_{k+1} := \min \mathbb{N} \setminus I_k$.

By construction $(i_k)_{k\in\mathbb{N}}$, resp. $(\beta(i_k))_{k\in\mathbb{N}}$ is an ascending sequence in $\mathbb{N}$, resp. A. We claim that the supremum $\bar{b}$ of $(\beta(i_k))_{k\in\mathbb{N}}$ is the supremum of B. A straightforward induction shows that for any $k \in \mathbb{N}$ we have $\forall j \leq i_k : \beta(j) \leq \beta(i_k)$. So take any $b \in B$. We then find a $k \in \mathbb{N}$ such that $\beta^{-1}(b) \leq i_k$. From this it follows that $b \leq \beta(i_k)$ also holds. Hence, $\bar{b}$ is an upper bound of B. On the other hand there cannot be an upper bound c of B with $c < \bar{b}$ as this would mean that there is a $k \in \mathbb{N}$ with $\beta(i_k) > c$. □

A partial order is an *ω-complete partial order (short: ω-cpo)* if it is ω-chain complete and it has a least element. ([1])

A map $f : A_1 \to A_2$ between two posets $\langle A_1, \leq_1 \rangle$, and $\langle A_2, \leq_2 \rangle$ is *monotone* if

$$\forall x, y \in A_1 : x \leq_1 y \Rightarrow f(x) \leq_2 f(y).$$

For both $\langle A_1, \leq_1 \rangle$, and $\langle A_2, \leq_2 \rangle$ ω-chain-complete, we say that f is *ω-continuous* if for any ω-chain $(a_i)_{i\in\mathbb{N}} \in A_1^\omega$ we have that

$$\sup^{\leq_2}_{i\in\mathbb{N}} f(a_i) = f\big(\sup^{\leq_1}_{i\in\mathbb{N}} a_i\big)$$

where $\sup^{\leq_i}$ denotes the supremum w.r.t. the partial order $\leq_i$. Note that every ω-continuous map is monotone.

Fixed points: Let $\langle A, \leq \rangle$ be a poset , and let f be a map on $\langle A, \leq$. We call $a \in A$ a *fixed point* if $f(a) = a$ holds; a is a *prefixed point* if $f(a) \leq a$, and it is a *postfixed point* if $a \leq f(a)$. By μf we denote the *least fixed point (LFP)* (w.r.t. $\leq$) if it exists. Similarly, the *greatest fixed point (GFP)* is denoted by νf if it exists.

Let $\langle A, \leq \rangle$ be a ω-cpo with least element $\bot$, and $f : A \to A$ an ω-continuous map. Then the *Kleene sequence* $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ of f is inductively defined by $\boldsymbol{\kappa}^{(0)} :=$

[1] The definition of ω-cpo varies with literature. The same structure is also called cpo, dcpo, or ω-dcpo by other authors. Our definition here is in the spirit of [Ési08].

$f(\bot)$, and $\boldsymbol{\kappa}^{(i+1)} := f(\boldsymbol{\kappa}^{(i)})$ for all $i \in \mathbb{N}$ ([2]). Because f is monotone, the Kleene sequence is an ω-chain in $\langle A, \leq \rangle$. Hence, its supremum $\sup^{\leq}_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)}$ exists. Because of ω-continuity this supremum is also a fixed point of f. A straightforward inductions show that $\sup^{\leq}_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)}$ is the least fixed point μf. This well-known result is called *Kleene fixed-point theorem*, or sometimes simply fixed-point theorem:

Theorem 2.1.2.
Let $f : A \to A$ be a ω-continuous function on the ω-cpo $\langle A, \leq \rangle$. Then the least fixed point μf exists and is given by $\sup^{\leq}_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)}$. ◇

2.2 ω-Continuous Semirings

Semirings: We have already introduced (ω-continuous) semirings informally in the introduction by means of several examples. Let us recall two of them:

Example 2.2.1. We described the *language semiring*, or free semiring, generated by some finite alphabet Σ which consisted of the set 2^{Σ^*} with set union as addition, and language concatenation as multiplication.

We also considered the question of determining the probability of termination for a given recursive program. Here, the dataflow equations were instantiated on the so called *real semiring*, i.e., the nonnegative real numbers extended by ∞. Addition, and multiplication on the real semiring were given by the natural extension of the canonical addition, and multiplication on $\mathbb{R}$ to encompass ∞, i.e., $a + \infty = \infty$, $0 \cdot \infty = 0$, and $a \cdot \infty = \infty$ for $a > 0$, otherwise. ◇

One arrives quite naturally at the formal definition of ω-continuous semiring by taking the similarities of these two examples. Note that this definition is more in the spirit of [Ésik08] then the one found in [Kui97]. (See the following remark for more details.)

Definition 2.2.2.
A semiring $\mathcal{S}$ is an algebraic structure $\langle S, +, \cdot, 0, 1 \rangle$ satisfying the following four properties:

(1) $\langle S, +, 0 \rangle$ is a commutative monoid.

(2) $\langle S, \cdot, 1 \rangle$ is a monoid.

(3) $0 \cdot a = a \cdot 0 = 0$ for all $a \in S$.

(4) $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(a + b) \cdot c = a \cdot c + b \cdot c$ for all $a, b, c \in S$.

[2] Defining $\boldsymbol{\kappa}^{(0)} = \mathbf{0}$ would be more straightforward, but less convenient for this thesis.

A semiring $\mathcal{S}$ is *naturally ordered* if it satisfies:

(5) The relation $\sqsubseteq := \{(a, b) \in S \times S \mid \exists d \in S : a + d = b\}$ is a partial order.

We call a semiring *totally ordered* if its natural order is a total order.

For a naturally ordered semiring $\mathcal{S}$ we write $\bigsqcup A$ for the $\sqsubseteq$-supremum of a set $A \subseteq S$ if it exists in S. A naturally ordered semiring is *ω-chain complete* if it satisfies:

(6) For all *ω-chains* $(a_i)_{i\in\mathbb{N}}$ the supremum $\bigsqcup_{i\in\mathbb{N}} a_i$ exists ([3]).

For an ω-chain complete semiring we define the sum of a sequence $(a_i)_{i\in\mathbb{N}}$ on S as follows:

$$\sum_{i\in\mathbb{N}} a_i := \bigsqcup\{a_0 + a_1 + \ldots + a_i \mid i \in \mathbb{N}\}.$$

We refer to $\sum$ as countable summation or ω-summation.

An ω-chain complete semiring is *ω-continuous* if:

(7) For any sequence $(a_i)_{i\in\mathbb{N}}$, any $c \in S$, and every partition $(I_j)_{j\in J}$ of $\mathbb{N}$:

$$c \cdot \left(\textstyle\sum_{i\in\mathbb{N}} a_i\right) = \textstyle\sum_{i\in\mathbb{N}}(c \cdot a_i)\ , \quad \left(\textstyle\sum_{i\in\mathbb{N}} a_i\right) \cdot c = \textstyle\sum_{i\in\mathbb{N}}(a_i \cdot c)\ ,$$

$$\textstyle\sum_{j\in J} \left(\sum_{i\in I_j} a_j\right) = \sum_{i\in\mathbb{N}} a_i\ .$$

In an ω-continuous semiring we define the Kleene star $^* : S \to S$ by

$$a^* := \sum_{k\in\mathbb{N}} a^k = \bigsqcup\{1 + a + a \cdot a + \ldots + a^k | k \in \mathbb{N}\} \text{ for } a \in S.$$

A semiring is *idempotent* if $a + a = a$ holds for all $a \in S$. It is *commutative* if $a \cdot b = b \cdot a$ for all $a, b \in S$. We use *io-semiring*, resp. *cio-semiring* as shorthands for idempotent ω-continuous semiring, resp. commutative idempotent ω-continuous semiring. ◇

In the following we often write ab instead of $a \cdot b$.

Remark 2.2.3.
(a) Note that in [Kui97] an ω-continuous semiring is defined to have an infinitary sum operator $\sum$ which is defined for any index set I and any family $(a_i)_{i\in I}$ of semiring elements. It is then additionally required that for $I = \mathbb{N}$ we have $\sum_{i\in\mathbb{N}} a_i = \bigsqcup\{a_0 + a_1 + \ldots + a_i \mid i \in \mathbb{N}\}$. From this it then

[3] This means that $\langle S, \sqsubseteq\rangle$ is a ω-cpo with least element 0.

immediately follows that the supremum of any ω-chain also has to exists (see (b)). As we only need $\sum$ to be defined for countable index sets I in the following, we chose to require (6) instead, and define $\sum$ directly.

(b) In any ω-chain complete semiring we can write the supremum $\bigsqcup_{i\in\mathbb{N}} a_i$ of an ω-chain $(a_i)_{i\in\mathbb{N}}$ as the sum $\sum_{i\in\mathbb{N}} d_i$ of its "differences" d_i with $d_0 := a_0$, and d_{i+1} defined by $a_i + d_{i+1} = a_{i+1}$. As $(a_i)_{i\in\mathbb{N}}$ is an ω-chain, we are guaranteed to find such d_i. Note that d_i does not need to be uniquely determined.

(c) The sum $\sum_{i\in\mathbb{N}} a_i$ of a sequence $(a_i)_{i\in\mathbb{N}}$ is independent of the enumeration of the elements appearing in $(a_i)_{i\in\mathbb{N}}$. (The reader may compare this to the notion of absolute convergence known from calculus.) That is for any bijection $\beta : \mathbb{N} \to \mathbb{N}$ we have $\sum_{i\in\mathbb{N}} a_i = \sum_{i\in\mathbb{N}} a_{\beta(i)}$. In particular, by Definition 2.2.2(7) we have

$$\left(\sum_{i\in\mathbb{N}} a_i\right) \cdot \left(\sum_{j\in\mathbb{N}} b_j\right) = \sum_{j\in\mathbb{N}} \sum_{i\in\mathbb{N}} (a_i \cdot b_j) = \sum_{k\in\mathbb{N}} (a_{I(k)} \cdot b_{J(k)})$$

for any bijection $(I, J) : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ using the partition $(I_j)_{j\in\mathbb{N}}$ with $I_j = \{k \in \mathbb{N} \mid J(k) = j\}$. We therefore simply write $\sum_{i,j\in\mathbb{N}} a_i \cdot b_j$ for the right-hand side in the following. Similarly, for any countable (multi)set A of elements of S we define $\sum A$ to be the sum of any sequence enumerating the elements of A.

(d) Every naturally ordered semiring $\mathcal{S}$ with idempotent addition is a *bounded join-semilattice*, i.e., for any two elements a, b of S the least upper bound $\bigsqcup\{a, b\}$ exists, as $\bigsqcup\{a, b\} = a + b$, and there is a least element, namely 0. In particular, if $\sqsubseteq$ is total, then addition and supremum coincide. If additionally S is also countable, then S itself is an ω-chain and, thus, a complete lattice. $\diamond$

The next proposition shows that multiplication and addition are themselves ω-continuous on an ω-continuous semiring.

Proposition 2.2.4.
In any ω-continuous semiring we have

$$c \cdot (\bigsqcup_{i\in\mathbb{N}} a_i) = \bigsqcup_{i\in\mathbb{N}} (c \cdot a_i) \ , \quad (\bigsqcup_{i\in\mathbb{N}} a_i) \cdot c = \bigsqcup_{i\in\mathbb{N}} (a_i \cdot c) \ ,$$

$$c + (\bigsqcup_{i\in\mathbb{N}} a_i) = \bigsqcup_{i\in\mathbb{N}} (c + a_i) \ .$$

for any ω-chain $(a_i)_{i\in\mathbb{N}}$ and any $c \in S$. $\diamond$

Proof. We use part (b) of Remark 2.2.3. Let $(d_i)_{i\in\mathbb{N}}$ be a sequence of differences of the

ω-chain $(a_i)_{i\in\mathbb{N}}$. We then have:

$$\begin{aligned}
c\cdot(\bigsqcup_{i\in\mathbb{N}} a_i) &= c\cdot(\sum_{i\in\mathbb{N}} d_i) && \text{(Rem. 2.2.3(b))}\\
&= \sum_{i\in\mathbb{N}}(c\cdot d_i) && \text{(Def. 2.2.2(7))}\\
&= \bigsqcup_{i\in\mathbb{N}}(c\cdot d_0+\ldots+c\cdot d_i) && \text{(Def. of } \textstyle\sum\text{)}\\
&= \bigsqcup_{i\in\mathbb{N}}\big(c\cdot(d_0+\ldots+d_i)\big) && \text{(Def. 2.2.2(4))}\\
&= \bigsqcup_{i\in\mathbb{N}}(c\cdot a_i) && \text{(Rem. 2.2.3(b)).}
\end{aligned}$$

Analogously, $(\bigsqcup_{i\in\mathbb{N}} a_i)\cdot c = \bigsqcup_{i\in\mathbb{N}}(a_i\cdot c)$ follows.

For the last equation we first define the sequence $(b_i)_{i\in\mathbb{N}}$ with $b_0 := c$, $b_1 := a_0$, and b_i is a difference of a_{i-1} and a_{i-2}, i.e., $a_{i-2}+b_i = a_{i-1}$ for all $i>1$. Further, set $J := \{0,1\}$, $I_0 := \{0\}$, and $I_1 := \mathbb{N}\setminus\{0\}$. We then have

$$\begin{aligned}
c+(\bigsqcup_{i\in\mathbb{N}} a_i) &= b_0+\sum_{i\in I_1} b_i && \text{(Rem. 2.2.3(b))}\\
&= \sum_{j\in J}\sum_{i\in I_j} b_i && (a+b=\textstyle\bigsqcup\{a,a+b\})\\
&= \sum_{i\in\mathbb{N}} b_i && \text{(Def. 2.2.2(7))}\\
&= \bigsqcup_{i\in\mathbb{N}}(b_0+b_1+\ldots b_i) && \text{(Rem. 2.2.3(b))}\\
&= \bigsqcup_{i\in\mathbb{N}}(c+a_i).
\end{aligned}$$

□

Remark 2.2.5.
In [Ési08] an ω-continuous semiring is defined as a semiring which is an ω-cpo w.r.t. the natural order ([4]) and whose addition and multiplication are ω-continuous. The sum of a (countable) sequence $(a_i)_{i\in\mathbb{N}}$ is then defined by taking the supremum of the ω-chain consisting of all sums of finite subsequences, i.e.,

$$\sum_{i\in I} a_i := \bigsqcup\left\{\sum_{i\in F} a_i \mid F\subseteq I, |F|\leq|\mathbb{N}|\right\}. \tag{2.1}$$

Using ω-continuity of multiplication and addition one then can show that the equations of Definition 2.2.2 (7) are satisfied, see for example [Kar92].

[4] In fact, in [Ési08] an ω-continuous semiring are not restricted to only the natural order, but a broader class of partial orders called positive orders.

So every naturally ordered ω-continuous semiring w.r.t. [Ési08] is also an ω-continuous semiring w.r.t. Definition 2.2.2. On the other hand, from Proposition 2.2.4 if follows that every semiring which is ω-continuous in our sense, is also ω-continuous w.r.t. [Ési08]. We therefore may freely use the results stated in [Ési08]. ◇

Definition 2.2.6.
Given two semirings $\mathcal{S}$ and $\mathcal{S}'$ with carrier S, resp. S', a map $h : S \to S'$ is a *semiring (homo)morphism* if it respects addition and multiplication, and preserves the neutral elements, i.e.,

$$h(a + b) = h(a + b), \quad h(a \cdot b) = h(a) \cdot h(b), \quad h(0) = 0, \quad h(1) = 1. \qquad \diamond$$

Proposition 2.2.7.
Given two ω-continuous semirings $\mathcal{S}$ and $\mathcal{S}'$ with carrier S, resp. S', a semiring morphism $h : S \to S'$ is ω-continuous iff it respects countable summation, i.e., $h(\sum_{i\in\mathbb{N}} a_i) = \sum_{i\in\mathbb{N}} h(a_i)$ holds. ◇

Proof. If h is ω-continuous, we may write

$$h(\sum_{i\in\mathbb{N}} a_i) = h(\bigsqcup\{a_1 + \ldots + a_i \mid i \in \mathbb{N}\}) = \bigsqcup\{h(a_1) + \ldots + h(a_i) \mid i \in \mathbb{N}\} = \sum_{i\in\mathbb{N}} h(a_i).$$

If h respects countable summation, then given some ω-chain $(a_i)_{i\in\mathbb{N}}$ we turn it into a sequence $(d_i)_{i\in\mathbb{N}}$ with $\sum_{i\in\mathbb{N}} d_i = \bigsqcup_{i\in\mathbb{N}} a_i$ as described in Remark 2.2.3 (b). We then have

$$h(\bigsqcup_{i\in\mathbb{N}} a_i) = h(\sum_{i\in\mathbb{N}} d_i) = \sum_{i\in\mathbb{N}} h(d_i) = \bigsqcup\{h(d_0) + \ldots + h(d_i) \mid i \in \mathbb{N}\} = \bigsqcup_{i\in\mathbb{N}} h(a_i). \qquad \square$$

Definition 2.2.8.
A semiring morphism h between two ω-continuous semirings is a *morphism of ω-continuous semirings* if it is ω-continuous or, equivalently, respects countable summation. ◇

Obviously, every morphism of ω-continuous semirings preserves the Kleene star, i.e., $h(a^*) = h(a)^*$.

We illustrate the definitions by means of the semirings encountered in the introduction:

Example 2.2.9. The *language semiring* $\mathcal{S}_\Sigma$ generated by a set Σ is given by $\langle 2^{\Sigma^*}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$ where $\cdot$ denotes the concatenation of languages. Its natural order is given by the subset relation. Countable summation is set union. We embed RExp_Σ into 2^{Σ^*} by means of L_Σ. It is well-known that the language semiring is (isomorphic to) the free io-continuous semiring generated by Σ (see [Ési08]), i.e., given some other io-semiring $\mathcal{S}$ with carrier S and some map $h : \Sigma \to S$, there is a unique extension of h to an homomorphism of ω-continuous semirings from the language semiring to $\mathcal{S}$.

The *counting semiring* $\mathcal{C}_k$ has the signature $\langle 2^{\mathbb{N}^k}, \cup, \cdot, \emptyset, \{\mathbf{0}\}\rangle$ with $k \in \mathbb{N}$, and $\mathbf{0} = (0, \ldots, 0) \in \mathbb{N}^k$. As already mentioned, we set

$$A \cdot B = \{a + b \mid a \in A, b \in B\} \text{ for } A, B \subseteq \mathbb{N}^k.$$

As in the case of the language semiring, natural order, resp. countable summation coincides with the subset relation, resp. set union. Similar to the language semiring, the counting semiring $\mathcal{C}_k$ is (isomorphic to) the free cio-semiring generated by k.

The free ω-continuous semiring generated by some set Σ is (isomorphic to) $\langle (\mathbb{N} \cup \{\infty\})^{\Sigma^*}, +, \cdot, \mathbf{0}, \mathbf{1}\rangle$ where $\mathbf{0} : \Sigma^* \to \{0\}$, resp. $\mathbf{1} : \Sigma^* \to \{0, 1\}$ with $\mathbf{1}(\varepsilon) = 1$, and otherwise $\mathbf{1}(w) = 0$ for all $w \in \Sigma^+$. Addition, natural order and countable summation are extended pointwise to $(\mathbb{N} \cup \{\infty\})^{\Sigma^*}$. Multiplication is defined by means of the cauchy product [Ési08]., i.e.,

$$(a \cdot b)(w) := \sum_{\Sigma^* \ni u, v \,:\, uv = w} a(u) \cdot b(v) \text{ for all } w \in \Sigma^*.$$

The $(\min, +)$-*semiring* is defined by $\langle \mathbb{R} \cup \{\pm\infty\}, \min, +, \infty, 0\rangle$. Unlike to the preceding examples, the natural order is the reverse of the canonical order on $\mathbb{R}$ unlike to the preceding examples. Countable summation is the infimum. In particular, the Kleene star is given by $a^* = \min\{k \cdot a \mid k \in \mathbb{N}\}$ with $a^* = 0$ for $a \geq 0$, and $*^* = -\infty$ otherwise.

The *real semiring* is defined by $\langle \mathbb{R}_{\geq 0} \cup \{\infty\}, +, \cdot, 0, 1\rangle$. Addition and multiplication in this semiring are obtained from the corresponding operations on $\mathbb{R}$ by adding the axioms $a + \infty = \infty$, $0 \cdot \infty = 0$, and $b \cdot \infty = \infty$ for all $a \geq 0$, and $b > 0$. The natural order is again given by the canonical order on the carrier. Countable summation is given by $\lim_{n\to\infty} \sum_{i=0}^{n} a_i$. All series converge because of the inclusion of ∞. The Kleene star is again easy to evaluate. We have $a^* = \frac{1}{1-a}$ for $a < 1$, and $a^* = \infty$ otherwise. $\diamond$

Vectors: Let $\mathcal{S} = \langle S, +, \cdot, 0, 1\rangle$ be an ω-continuous semiring. We use $\mathcal{X}$ for denoting a finite set of variables. The elements of $S^{\mathcal{X}}$ are then called *vectors* and are printed in boldface. We simply write V for $S^{\mathcal{X}}$ if S and $\mathcal{X}$ are clear from the context. A vector $\boldsymbol{v} \in S^{\mathcal{X}}$ assigns to a variable $X \in \mathcal{X}$ the value $\boldsymbol{v}_X$, or $(\boldsymbol{v})_X$ if necessary for avoiding ambiguities. For example, given a sequence of vectors, like $(\boldsymbol{v}_i)_{i\in\mathbb{N}}$, we write $\boldsymbol{v}_i$ for denoting the ith vector in the series, and $(\boldsymbol{v}_i)_X$ for the value $\boldsymbol{v}_i$ maps X to. If there is some canonical total order given on $\mathcal{X}$ like e.g. the lexicographic order in the case $\mathcal{X} = \{X, Y, Z\}$, or the total order on the indices in the case $\mathcal{X} = \{X_1, X_2, X_3\}$ we will also write a vector $\boldsymbol{v}$ as a (traditional) column vector of dimension $|\mathcal{X}|$ enumerating the values starting with the variable of lowest rank. Addition on $\mathcal{S}$ is lifted componentwise to V, i.e., $(\boldsymbol{u}+\boldsymbol{v})_X := \boldsymbol{u}_X + \boldsymbol{v}_X$, and $(\sum_{i\in\mathbb{N}} \boldsymbol{v}_i)_X := \sum_{i\in\mathbb{N}} (\boldsymbol{v}_i)_X$. Similarly, we lift the natural order on S to V by defining $\boldsymbol{u} \sqsubseteq \boldsymbol{v} :\Leftrightarrow \forall X \in \mathcal{X} : \boldsymbol{u}_X \sqsubseteq \boldsymbol{v}_X$. Then V is also ω-chain complete, as for any ω-chain $(\boldsymbol{v}_i)_{i\in\mathbb{N}}$ the supremum $\bigsqcup_{i\in\mathbb{N}} \boldsymbol{v}_i$ is given by $(\bigsqcup_{i\in\mathbb{N}} \boldsymbol{v}_i)_X = \bigsqcup_{i\in\mathbb{N}} (\boldsymbol{v}_i)_X$ for all $X \in \mathcal{X}$.

Monomials, Polynomials, and Power-Series: Fix some finite set $\mathcal{X}$ of variables. A *monomial* in $\mathcal{X}$ is a finite product $a_1X_1a_2X_2\cdots a_kX_ka_{k+1}$, where $k \geq 0$, $a_1, \ldots, a_{k+1} \in S$ and $X_1, \ldots, X_k \in \mathcal{X}$. We assume that for $k \geq 1$ all coefficients $a_1, \ldots, a_{k+1}$ are different from 0.

Note that this more general definition of monomial is necessary as we do not require that multiplication is commutative. A *polynomial* in $\mathcal{X}$ is an expression of the form $m_1 + \ldots + m_k$ where $k \geq 0$ and $m_1, \ldots, m_k$ are monomials in $\mathcal{X}$. A *power series* in $\mathcal{X}$ is an expression of the form $\sum_{i\in I} m_i$, where I is a countable set and m_i is a monomial in $\mathcal{X}$ for every $i \in I$.

Given a monomial $f = a_1X_1a_2X_2 \ldots a_kX_ka_{k+1}$ and a vector $\vec{v}$, we define $f(\vec{v})$, the *value of* f *at* $\vec{v}$, as $a_1\boldsymbol{v}_{X_1}a_2\boldsymbol{v}_{X_2}\cdots a_k\boldsymbol{v}_{X_k}a_{k+1}$. We extend this to any power series $f = \sum_{i\in I} f_i$ by $f(\vec{v}) = \sum_{i\in I} f_i(\vec{v})$.

A *vector of power series* is a mapping $\boldsymbol{f}$ that assigns to each variable $X \in \mathcal{X}$ a power series $\boldsymbol{f}(X)$. Again we write $\boldsymbol{f}_X$ for $\boldsymbol{f}(X)$. Given a vector $\vec{v}$, we define $\boldsymbol{f}(\vec{v})$ as the vector satisfying $(\boldsymbol{f}(\vec{v}))_X = \boldsymbol{f}_X(\vec{v})$ for every $X \in \mathcal{X}$, i.e., $\boldsymbol{f}(\vec{v})$ is the vector that assigns to X the result of evaluating the power series $\boldsymbol{f}_X$ at $\vec{v}$. So, $\boldsymbol{f}$ naturally induces a mapping $\boldsymbol{f}\colon V \to V$.

As addition and multiplication are ω-continuous on an ω-continuous semiring, one can show that (vectors of) power-series are ω-continuous maps.

Proposition 2.2.10 ([Kui97])**.**
Let $\mathcal{S}$ be an ω-continuous semiring, and $\boldsymbol{f}$ a system of power-series in $\mathcal{X}$. Then $\boldsymbol{f}$ is an ω-continuous and, thus, monotone map from V to V. ◇

We refer the reader to [Kui97] for a formal proof. As every ω-continuous semiring is an ω-cpo, we may instantiate Kleene's fixed-point Theorem 2.1.2 for our setting:

Definition 2.2.11.
Let $\boldsymbol{f}$ be a vector of power series. The Kleene sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ (w.r.t. $\boldsymbol{f}$) is defined by $\boldsymbol{\kappa}^{(0)} := \boldsymbol{f}(\mathbf{0})$, and $\boldsymbol{\kappa}^{(i+1)} := \boldsymbol{f}(\boldsymbol{\kappa}^{(i)})$ for all $i \in \mathbb{N}$. ◇

Theorem 2.2.12.
Let $\boldsymbol{f}$ be a vector of power series. Then its Kleene sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ is an ω-chain. The supremum $\bigsqcup_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)}$ is the least fixed point $\mu\boldsymbol{f}$ of $\boldsymbol{f}$. ◇

We introduce two properties of vectors of power series:

Definition 2.2.13.
Let $\boldsymbol{f}$ be a vector of power series. We say that $\boldsymbol{f}$ is *clean* if $\mu\boldsymbol{f}$ is greater than zero in every component, i.e., $0 \sqsubset (\mu\boldsymbol{f})_X$ for all $X \in \mathcal{X}$ holds. ◇

Definition 2.2.14.
Let $\boldsymbol{f}$ be a vector of power series given in the variables $\mathcal{X}$. The *dependency graph* of $\boldsymbol{f}$ is the directed graph which has $\mathcal{X}$ as its nodes, and there is an edge from X to Y if there is a monomial m appearing in $\boldsymbol{f}_X$ such that Y is a variable of m. We say that X *depends* on Y if there is a finite path from X to Y in the dependency graph. $\boldsymbol{f}$ is *strongly-connected* if its dependency graph is strongly-connected. $\diamond$

We can easily decide whether $\boldsymbol{f}$ is strongly-connected. The following lemma shows that we can also determine whether $\boldsymbol{f}$ is clean:

Lemma 2.2.15.
Let $\boldsymbol{f}$ be a vector of power series given in the variables $\mathcal{X}$ with $n = |\mathcal{X}|$ on a ω-continuous semiring without zero divisors, i.e., $a \cdot b = 0 \Rightarrow a = 0 \vee b = 0$ for all $a, b \in S$. Then $\boldsymbol{f}$ is clean iff $0 \sqsubset \boldsymbol{\kappa}_X^{(n-1)}$ for all $X \in \mathcal{X}$. $\diamond$

Proof. By Theorem 2.2.12 we have only to show the implication from left to right. So assume that $\boldsymbol{f}$ is clean. Again by Theorem 2.2.12 we obtain that for every variable X there has to exist a least $k_X \in \mathbb{N}$ such that $0 \sqsubset \boldsymbol{\kappa}_X^{(k_X)}$ Obviously, we have either $k_X = 0$ or there is some subset A of $\mathcal{X}$ such that $k_X = 1 + \max\{k_Y \mid Y \in A\}$. As $k_X = 0$ has to hold for at least one $X \in \mathcal{X}$, we have $k_X < n$ for all $X \in \mathcal{X}$. $\square$

Viewing vectors of power series as maps we define the product $\boldsymbol{f} \circ \boldsymbol{g}$ of two vectors $\boldsymbol{f}, \boldsymbol{g}$ of power series to be the composition of the two maps, i.e.,

$$(\boldsymbol{f} \circ \boldsymbol{g})(\boldsymbol{v})_X := \boldsymbol{f}_X(\boldsymbol{g}(\boldsymbol{v})).$$

Note that $\boldsymbol{f} \circ \boldsymbol{g}$ is also a vector of power series, as the underlying semiring is required to be ω-continuous. Defining the addition of two vectors $\boldsymbol{f}, \boldsymbol{g}$ of power series by

$$(\boldsymbol{f} + \boldsymbol{g})_X := \boldsymbol{f}_X + \boldsymbol{g}_X$$

we therefore obtain the *semiring of vectors of power series over $\mathcal{S}$ w.r.t. $\mathcal{X}$*. The neutral elements are given by the vector assigning zero to every variable and the vector assigning each variable X the monomial X. Further, as $\mathcal{S}$ is ω-continuous, the semiring of vectors of power series is ω-continuous, too.

Example 2.2.16. Consider the following polynomial system $\boldsymbol{f}$ w.r.t. $\mathcal{X} = \{X, Y\}$:

$$\boldsymbol{f}_X := aX + bY \quad \boldsymbol{f}_Y := cX + dY,$$

or equivalently

$$\boldsymbol{f} = \begin{pmatrix} aX + bY \\ cX + dY \end{pmatrix}.$$

We can write $\boldsymbol{f}$ as

$$\boldsymbol{f} = \underbrace{\begin{pmatrix} a & b \\ c & d \end{pmatrix}}_{=:M} \underbrace{\begin{pmatrix} X \\ Y \end{pmatrix}}_{=:\boldsymbol{X}} = M\boldsymbol{X}$$

using standard matrix-vector-multiplication assuming here that the elements of $\boldsymbol{X}$ are multiplied from the right to the elements of M. The Kleene-star $\boldsymbol{f}^*$ of $\boldsymbol{f}$, formally defined by

$$\boldsymbol{f}^* : V \to V : \boldsymbol{v} \mapsto \sum_{k\in\mathbb{N}} \boldsymbol{f}^k(\boldsymbol{v}) = \sum_{k\in\mathbb{N}} \underbrace{\boldsymbol{f}(\boldsymbol{f}(\ldots \boldsymbol{f}(\boldsymbol{v})))}_{k \text{ applications of } \boldsymbol{f}},$$

can then be written as

$$\boldsymbol{f}^* : V \to V : \boldsymbol{v} \mapsto \sum_{k\in\mathbb{N}} \begin{pmatrix} a & b \\ c & d \end{pmatrix}^k \boldsymbol{v} = M^*\boldsymbol{v}.$$

So, $\boldsymbol{f}^*$ can be identified with the Kleene-star of the matrix M. Specifically, we can think of M as the adjacency matrix of finite graph G_M with nodes X and Y where every edge is weighted by a semiring element. Then the entries of M^k correspond to the weight of all paths of length exactly k between two states, i.e., M^* is the reflexive-transitive closure.

For example, let $\mathcal{S}$ be the language semiring generated by $\Sigma = \{a, b, c, d\}$. Then we can calculate M^* in the same way in which we obtain regular expressions describing all finite paths between two states from a finite automata, e.g. by means of the Floyd-Warshall algorithm. For this example we obtain

$$M^* = \begin{pmatrix} (a + bc^*d)^* & b(c + da^*b)^* \\ d(a + bc^*d)^* & (c + da^*b)^* \end{pmatrix}.$$

Assume we are given some particular matrix $A = (a_{i,j})_{i,j=1,2}$ on some io-semiring. Then the map $h(a) = a_{1,1}, h(b) = a_{1,2}, h(c) = a_{2,1}, h(d) = a_{2,2}$ uniquely determines a morphism of io-semirings as the language semiring is freely generated by Σ. We simply call this homomorphism also h. As h respects the Kleene star, we therefore obtain A^* by applying h to the entries of M^*. See also Example 1.1.5. This observation yields the well-known fact that for all io-semirings $\mathcal{S}$ we can obtain regular expressions of the entries of M^* efficiently by calculating M^* on the language semiring generated by the matrix coefficients.

In the case of the real semiring there is another interpretation of M^* coming from analysis (or functional analysis), as M^* corresponds to the geometric series ($|\mathcal{X}| = 1$), or the Neumann series ($|\mathcal{X}| > 1$). Here, we have that M^* equals $(\mathrm{Id} - M)^{-1}$ if the spectral radius, i.e., the largest absolute value of an eigenvalue of M, is less than 1 (assuming the M exists in $\mathbb{R}^{2\times 2}$). ◇

Chapter 3

Newton's Method on ω-Continuous Semirings

3.1 Generalizing Newton's Method

We introduce our generalization of Newton's method for ω-continuous semirings. We first consider only equations in a single variable, i.e., the univariate case. This allows us to introduce the underlying ideas while reducing the amount of additional notation. We first recall Newton's method for approximating a zero of a differentiable univariate function. We then take a close look at the analytical definition, and identify those parts of the definition which do not carry over directly to ω-continuous semirings. Finally, we motivate our "translations" of these parts to ω-continuous semirings. This is done in Section 3.1.1. In Section 3.1.2 we then lift our generalization of Newton's method to the multivariate case, while Sections 3.1.3 and 3.1.4 finally present the proofs that our generalization of Newton's method is well-defined and converges to the least fixed point.

3.1.1 The Univariate Case

Given a differentiable function $g\colon \mathbb{R} \to \mathbb{R}$, Newton's method computes a zero of g, i.e., a solution of the equation $g(X) = 0$. The method starts at some value $\nu^{(0)}$ "close enough" to the zero, and proceeds iteratively: given $\nu^{(i)}$, it computes a value $\nu^{(i+1)}$ closer to the zero than $\nu^{(i)}$. For that, the method *linearizes* g at $\nu^{(i)}$, i.e., computes the tangent to g passing through the point

$(\nu^{(i)}, g(\nu^{(i)}))$, and takes $\nu^{(i+1)}$ as the zero of the tangent (i.e., the x-coordinate of the point at which the tangent cuts the x-axis). One therefore may say that Newton's method reduces the problem of finding the zero of a non-linear function to the problem of finding the zeros of a sequence of linear functions.

It is convenient for our purposes to formulate Newton's method in terms of the *differential (form)* of g at a given point $v \in \mathbb{R}$. Recall that the differential of g is basically the mapping $Dg|_v : \mathbb{R} \to \mathbb{R}$ that assigns to each $v \in \mathbb{R}$ the linear function describing to the tangent of g at $(v, g(v))$, represented in the coordinate system having $(v, g(v))$ as origin ([1]). If we denote the differential of g at v by $Dg|_v$, then we have $Dg|_v(X) = g'(v) \cdot X$ (for example, if $g(X) = X^2 + 3X + 1$, then $Dg|_3(X) = 9X$). In terms of differentials, Newton's method is formulated as follows. Starting at some $\nu^{(0)}$, compute iteratively $\nu^{(i+1)} = \nu^{(i)} + \Delta^{(i)}$, where $\Delta^{(i)}$ is the solution of the linear equation $Dg|_{\nu^{(i)}}(X) + g(\nu^{(i)}) = 0$ (assume for simplicity that the solution of the linear system is unique).

Computing the solution of a fixed-point equation, $f(X) = X$ amounts to computing a zero of $g(X) = f(X) - X$, and so we can apply Newton's method. Since for every real number v we have $Dg|_v(X) = Df|_v(X) - X$, the method looks as follows:

Starting at some $\nu^{(0)}$, compute iteratively

$$\nu^{(i+1)} = \nu^{(i)} + \Delta^{(i)} \tag{3.1}$$

where $\Delta^{(i)}$ is the solution of the linear equation

$$Df|_{\nu^{(i)}}(X) + f(\nu^{(i)}) - \nu^{(i)} = X \; . \tag{3.2}$$

So Newton's method "breaks down" the problem of finding a solution to a non-linear system $f(X) = X$ into finding solutions to the sequence (3.2) of linear systems.

Generalization

Generalizing Newton's method to arbitrary ω-continuous semirings requires to overcome two obstacles. First, the notion of differential seems to require

[1] More precisely, in differential geometry the differential (form) is defined as an one-form. For instance, let $f : \mathbb{R} \to \mathbb{R} : x \mapsto f(x)$ be a differentiable map. Then its differential df is given by $df = f' \cdot dx$ with f' the derivative of f w.r.t. x, and dx the coordinate differential one-form. Evaluating the differential at some point $p \in \mathbb{R}$ yields then the one-form $df|_p = f'(p) \cdot dx$, which describes the tangent at f in p w.r.t. the origin $(p, f(p))$.

a richer algebraic structure than a semiring: differentials are usually defined in terms of derivatives, which are the limit of a quotient of differences, which requires both the sum and product operations to have inverses. Second, equation (3.2) contains the term $f(\nu^{(i)}) - \nu^{(i)}$, which again seems to be defined only if summation has an inverse.

The first obstacle Differentiable functions satisfy well-known algebraic rules with respect to sums and products of functions. We take these rules as the definition of the differential of a power series f over an ω-continuous semiring $\mathcal{S}$. We remark that this definition of differential generalizes the usual algebraic definition of derivatives.

Definition 3.1.1.
Let f be a power series in one variable X over an ω-continuous semiring $\mathcal{S}$. The *differential of* f at the point v is the mapping $Df|_v : S \to S$ inductively defined as follows for every $b \in S$:

$$Df|_v(b) = \begin{cases} 0 & \text{if } f \in S \\ b & \text{if } f = X \\ Dg|_v(b) \cdot h(v) + g(v) \cdot Dh|_v(b) & \text{if } f = g \cdot h \\ \sum_{i \in I} Df_i|_v(b) & \text{if } f = \sum_{i \in I} f_i(b)\ . \end{cases}$$

◇

Example 3.1.2. Consider first a polynomial f over some *commutative* ω-continuous semiring. Because of commutative multiplication, we may write any monomial as $a \cdot X^k$ for some $k \in \mathbb{N}$ and $a \in S$, and so $f = \sum_{k=0}^{n} a_k \cdot X^k$ for suitable $n \in \mathbb{N}$ and $a_k \in S$. Let f' denote the usual algebraic derivative of f w.r.t. X, i.e., $f' = \sum_{k=1}^{n} k \cdot a_k \cdot X^{k-1}$ where $k \cdot a_k$ is an abbreviation of $a_k \cdot \sum_{i=1}^{k} 1$. We then have

$$\begin{aligned} Df|_v(b) &= \sum_{k=0}^{n} D(a_k \cdot X^k)|_v(b) \\ &= \sum_{k=0}^{n} (Da_k|_v(b) \cdot (X^k)(v) + \sum_{j=0}^{k-1} a_k \cdot (X^j)(v) \cdot DX|_v(b) \cdot (X^{k-1-j})(v)) \\ &= \sum_{k=0}^{n} \sum_{j=0}^{k-1} a_k \cdot v^j \cdot DX|_v(b) \cdot v^{k-1-j} \\ &= (\sum_{k=1}^{n} k \cdot a_k \cdot v^{k-1}) \cdot b \\ &= f'(v) \cdot b. \end{aligned}$$

So, on commutative semirings, we have $Df|_v(b) = f'(v) \cdot b$ for all $v, b \in S$.

Now, assume that multiplication is not commutative, and consider the simple case of a quadratic monomial $m = a_0 X a_1 X a_2$. We then have

$$\begin{aligned} Dm|_v(b) &= a_0 \cdot DX|_v(b) \cdot a_1 \cdot v \cdot a_2 + a_0 \cdot v \cdot a_1 \cdot DX|_v(b) \cdot a_2 \\ &= a_0 \cdot b \cdot a_1 \cdot v \cdot a_2 + a_0 \cdot v \cdot a_1 \cdot b \cdot a_2. \end{aligned}$$

The important point here is that the differential "remembers" the position of the variables, and therefore not simply "appends" the value b. ◇

The second obstacle Profiting from the fact that 0 is the unique minimal element of $\mathcal{S}$ with respect to $\sqsubseteq$, we fix $\nu^{(0)} = f(0)$, which guarantees that $\nu^{(0)}$ satisfies $\nu^{(0)} \sqsubseteq f(\nu^{(0)})$. We *guess* that with this choice $\nu^{(i)} \sqsubseteq f(\nu^{(i)})$ will hold not only for $i = 0$, but for every $i \geq 0$ (the correctness of this guess is proved in Theorem 3.1.10). If the guess is correct, then, by the definition of $\sqsubseteq$, the semiring contains an element $\delta^{(i)}$ such that $f(\nu^{(i)}) = \nu^{(i)} + \delta^{(i)}$. We replace $f(\nu^{(i)}) - \nu^{(i)}$ by any such $\delta^{(i)}$. This leads to the following definition:

Definition 3.1.3.
Let f be a power series in one variable. A *Newton sequence* $(\nu^{(i)})_{i\in\mathbb{N}}$ is given by:

$$\nu^{(0)} = f(0) \quad \text{and} \quad \nu^{(i+1)} = \nu^{(i)} + \Delta^{(i)} \tag{3.3}$$

where $\Delta^{(i)}$ is the least solution of

$$Df|_{\nu^{(i)}}(X) + \delta^{(i)} = X \tag{3.4}$$

and $\delta^{(i)}$ is any element satisfying $f(\nu^{(i)}) = \nu^{(i)} + \delta^{(i)}$. ◇

In Section 3.1.3 we show that Newton sequences always exist (i.e., there is always at least one possible choice for $\delta^{(i)}$), and that they all converge at least as fast as the Kleene sequence. More precisely, we show that for every $i \geq 0$

$$\kappa^{(i)} \sqsubseteq \nu^{(i)} \sqsubseteq \nu^{(i+1)} \sqsubseteq \mu f \ .$$

Since we have $\mu f = \bigsqcup_{i\in\mathbb{N}} \kappa^{(i)}$ by Kleene's theorem, Newton sequences converge to μf.

In general, there might be more than one choice for $\delta^{(i)}$. In Section 3.1.4 we show, however, that the Newton sequence $(\nu^{(i)})_{i\geq 0}$ itself is uniquely determined by $\boldsymbol{f}$ (and $\mathcal{S}$). In other words, the choice of $\delta^{(i)}$ does not influence the Newton approximants $\nu^{(i)}$.

Before proving these results, let us consider some examples.

Examples

We compute the Newton sequence for a program that can execute a and terminate, or execute b and then call itself twice, recursively (the abstract scheme of a divide-and-conquer procedure). The abstract dataflow equation of the program is

$$X = a + b \cdot X \cdot X. \tag{3.5}$$

Example 3.1.4 (**The real semiring**). Consider the case $a = b = 1/2$ (we can interpret a and b as probabilities). We have $Df|_v(X) = v \cdot X$, and one single possible choice for $\delta^{(i)}$, namely $\delta^{(i)} = f(\nu^{(i)}) - \nu^{(i)} = 1/2 + 1/2\,(\nu^{(i)})^2 - \nu^{(i)}$. Equation (3.4) becomes

$$\nu^{(i)}\, X + 1/2 + 1/2\,(\nu^{(i)})^2 - \nu^{(i)} = X$$

with $\Delta^{(i)} = (1 - \nu^{(i)})/2$ as unique solution. We get

$$\nu^{(0)} = 1/2 \qquad \nu^{(i+1)} = (1 + \nu^{(i)})/2$$

and therefore $\nu^{(i)} = 1 - 2^{(i+1)}$. So the Newton sequence converges to 1 and gains one bit of accuracy per iteration. ◇

Example 3.1.5 (**The language semiring**). Consider the language semiring with $\Sigma = \{a, b\}$. The product operation is concatenation of languages, and hence non-commutative. So we have $Df|_v(X) = bvX + bXv$. We show in Proposition 3.3.1 that when sum is idempotent (as in this case, where it is union of languages) the definition of the Newton sequence can be simplified to

$$\nu^{(0)} = f(0) \quad \text{and} \quad \nu^{(i+1)} = \Delta^{(i)}, \tag{3.6}$$

where $\Delta^{(i)}$ is the least solution of

$$Df|_{\nu^{(i)}}(X) + f(\nu^{(i)}) = X\ . \tag{3.7}$$

With $f = a + b \cdot X \cdot X$ from Equation (3.5), Equation (3.7) becomes

$$\underbrace{b\nu^{(i)}X + bX\nu^{(i)}}_{Df|_{\nu^{(i)}}(X)} + \underbrace{a + b\nu^{(i)}\nu^{(i)}}_{f(\nu^{(i)})} = X\ . \tag{3.8}$$

Its least solution (which by (3.6) is equal to $(i+1)$-th Newton approximant) is a context-free language. Let $G^{(i)}$ be a grammar with axiom $S^{(i)}$ such that $\nu^{(i)} = L(G^{(i)})$. Since $\nu^{(0)} = f(0)$, the grammar $G^{(0)}$ contains one single production, namely $S^{(0)} \to a$. Equation (3.8) allows us to define $G^{(i+1)}$ in terms of $G^{(i)}$, and we get:

$$\begin{aligned} G^{(0)} &= \{S^{(0)} \to a\} \\ G^{(i+1)} &= G^{(i)} \cup \{S^{(i+1)} \to a \mid bS^{(i+1)}S^{(i)} \mid bS^{(i)}S^{(i+1)} \mid bS^{(i)}S^{(i)}\} \end{aligned}$$

Let $G = \{S \to a \mid bSS\}$ be the grammar derived from Equation (3.5). We have $L(G) = \bigcup_{i=1}^{n} L(G^{(i)})$. It is easy to see that $L(G^{(i)})$ contains the words of $L(G)$ *of index* $i + 1$.

Loosely speaking, the index of a word $w \in L(G)$ is the least number i such that some derivation of w contains no intermediate word having more than i occurrences of variables [Sal69]. Formally, the index of $w \in L(G)$ is the least number i for which a derivation $X = \alpha_0 \Rightarrow \cdots \Rightarrow \alpha_r = w$ exists such that for every $i \in \{0, \ldots, r\}$ the projection of α_i onto $\{X_1, \ldots, X_n\}$ has at most length i. In Section 3.3.1 we show that this characterization of the Newton approximants holds in general, i.e., the i-th Newton approximant of the language generated by a grammar G contains the words of $L(G)$ of index at most $i+1$. $\diamond$

Example 3.1.6 (**The counting semiring**). Consider the counting semiring with $r_a = \{(1,0)\}$ and $r_b = \{(0,1)\}$. Since the sum operation is union of sets of vectors, it is idempotent and Equations (3.6) and (3.7) hold. Since the product operation is now commutative, we obtain for our example

$$b \cdot \nu^{(i)} \cdot X + a + b \cdot \nu^{(i)} \cdot \nu^{(i)} = X \tag{3.9}$$

Using Kleene's fixed-point theorem (Theorem 2.2.12), it is easy to see that the least solution of a linear equation $X = u \cdot X + v$ over a commutative ω-continuous semiring is $u^* \cdot v$, where $u^* = \sum_{i \in \mathbb{N}} u^i$. The least solution $\Delta^{(i)}$ of Equation (3.9) is then given by

$$\Delta^{(i)} = (r_b \cdot \nu^{(i)})^* \cdot (r_a + r_b \cdot \nu^{(i)} \cdot \nu^{(i)})$$

and we obtain:

$$\begin{aligned}
\nu^{(0)} &= r_a = \{(1,0)\} \\
\nu^{(1)} &= (r_b \cdot r_a)^* \cdot (r_a + r_b \cdot r_a \cdot r_a) = \{(n,n) \mid n \geq 0\} \cdot \{(1,0),(2,1)\} \\
&= \{(n+1,n) \mid n \geq 0\} \\
\nu^{(2)} &= (\{(n,n) \mid n \geq 1\})^* \cdot (\{(1,0)\} \cup \{(2n+2,2n+1) \mid n \geq 0\}) \\
&= \{(n+1,n) \mid n \geq 0\}
\end{aligned}$$

So the Newton sequence reaches a fixed point after one iteration. In Section 3.4 we show that the Newton sequence of a system of n equations over *any commutative* and *idempotent* semiring converges after at most n iterations. Further note that the counting semiring does not satisfy the ascending-chain property, i.e. there are monotonically increasing sequences in the counting semiring which do not become stationary. Therefore, the Kleene sequence (and possible variations) does not reach $\mu \boldsymbol{f}$ after a finite number of steps in general. $\diamond$

3.1.2 The Multivariate Case

Newton's method can be easily generalized to the multivariate case. Given differentiable functions $g_1, \ldots, g_n \colon \mathbb{R}^n \to \mathbb{R}$, the method computes a solution of $\boldsymbol{g}(\boldsymbol{X}) = \boldsymbol{0}$, where $\boldsymbol{g} = (g_1, \ldots, g_n)$. Starting at some $\boldsymbol{\nu}^{(0)}$, the method computes $\boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)}$, where $\boldsymbol{\Delta}^{(i)}$ is the solution of the *system* of linear equations

$$\begin{aligned}
Dg_1|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{X}) + g_1(\boldsymbol{\nu}^{(i)}) &= 0 \\
&\vdots \\
Dg_n|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{X}) + g_n(\boldsymbol{\nu}^{(i)}) &= 0
\end{aligned}$$

and $Dg_j|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{X})$ is the differential of g_j at $\boldsymbol{\nu}^{(i)}$, i.e., the function corresponding to the tangent hyperplane of g_j at the point $(\boldsymbol{\nu}^{(i)}, g_j(\boldsymbol{\nu}^{(i)}))$.

Given a function $g\colon \mathbb{R}^n \to \mathbb{R}$ differentiable at a point $\boldsymbol{v}$, there exists a function $D_X g|_v$ for each variable $X \in \mathcal{X}$ such that $Dg|_{\boldsymbol{v}} = \sum_{X\in\mathcal{X}} D_X g|_{\boldsymbol{v}}$. These functions are closely related to the partial derivatives, more precisely we have $D_X g|_{\vec{v}}(\boldsymbol{X}) = \partial_X g|_{\vec{v}} \cdot X$.

We denote the system above by $D\boldsymbol{g}|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{X}) + \boldsymbol{g}(\boldsymbol{\nu}^{(i)}) = \mathbf{0}$. For the problem of computing a solution of a system of fixed-point equations, the method looks as follows:

> Starting at some $\boldsymbol{\nu}^{(0)}$, compute iteratively
>
> $$\boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)} \tag{3.10}$$
>
> where $\boldsymbol{\Delta}^{(i)}$ is the least solution of the linear system of fixed-point equations
>
> $$D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{X}) + \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) - \boldsymbol{\nu}^{(i)} = \boldsymbol{X}\ . \tag{3.11}$$

Generalization

Again, we use the algebraic definition of differential:

Definition 3.1.7.
Let f be a power series over an ω-continuous semiring $\mathcal{S}$ and let $X \in \mathcal{X}$ be a variable. The *differential of f w.r.t. X* at the point $\boldsymbol{v}$ is the mapping $D_X f|_{\boldsymbol{v}} : V \to S$ inductively defined as follows:

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

Further, we define the *differential of f* at $\boldsymbol{v}$ as the function

$$Df|_{\boldsymbol{v}} := \sum_{X\in\mathcal{X}} D_X f|_v.$$

Finally, the differential of a vector of power series $\boldsymbol{f}$ at $\boldsymbol{v}$ is defined as the function $D\boldsymbol{f}|_{\boldsymbol{v}} : V \to V$ with

$$(D\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}))_X := D\boldsymbol{f}_X|_{\boldsymbol{v}}(\boldsymbol{b})\ . \qquad \diamond$$

Remark 3.1.8.
The differential is *additive*, i.e. for any $\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{b}' \in V$, and $\boldsymbol{f}$ we have

$$D\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b} + \boldsymbol{b}') = D\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}) + D\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}') \ .$$

If multiplication is commutative, then the differential is even a *linear* operator for any fixed $\boldsymbol{v}$. Further, for commutative multiplication we can again represent the differential of a power series f w.r.t. X by means of derivatives: we have $D_X f|_{\boldsymbol{v}}(\boldsymbol{b}) = \partial_X f|_{\boldsymbol{v}} \cdot \boldsymbol{b}_X$ with $\partial_X f|_{\boldsymbol{v}}$ the (algebraic) partial derivative of the power series f w.r.t. X. Similarly, the differential can be represented by means of the gradient of a power series f, or more generally, by the Jacobian of a vector $\boldsymbol{f}$ of power series. ◇

As in the univariate case we guess that $\boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$ will hold for every $i \geq 0$. If the guess is correct, then the semiring contains an element $\boldsymbol{\delta}^{(i)}$ such that $\boldsymbol{f}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$, and Equation (3.11) becomes

$$D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{X}) + \boldsymbol{\delta}^{(i)} = \boldsymbol{X} \ . \tag{3.12}$$

This leads to the following definition:

Definition 3.1.9.
Let $\boldsymbol{f} \colon V \to V$ be a vector of power series. For $i \in \mathbb{N}$, an i-th *Newton approximant* $\boldsymbol{\nu}^{(i)}$ is inductively defined by

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\vec{0}) \quad \text{and} \quad \boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)} \ ,$$

where $\boldsymbol{\Delta}^{(i)}$ is the least solution of Equation (3.12) and $\boldsymbol{\delta}^{(i)}$ is any vector satisfying $\boldsymbol{f}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$.

A sequence $(\boldsymbol{\nu}^{(i)})_{i \in \mathbb{N}}$ of Newton approximants is called *Newton sequence*. ◇

3.1.3 Fundamental Properties of the Newton Sequences

In the rest of the section we prove the following theorem, showing that there exists exactly one Newton sequence, that it converges to the least fixed point, and does so at least as fast as the Kleene sequence.

Theorem 3.1.10.
Let $\boldsymbol{f} \colon V \to V$ be a vector of power series. Then, the Newton sequence $(\boldsymbol{\nu}^{(i)})_{i \in \mathbb{N}}$ is uniquely determined. Further, the Newton sequence increases

monotonically, converges to the least fixed point, and does so at least as fast as the Kleene sequence. More precisely, it satisfies

$$\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) \sqsubseteq \boldsymbol{\nu}^{(i+1)} \sqsubseteq \mu\boldsymbol{f} = \bigsqcup_{j\in\mathbb{N}} \boldsymbol{\kappa}^{(j)} \text{ for all } i \in \mathbb{N}. \qquad \diamond$$

We split the proof Theorem 3.1.10 in two propositions. Proposition 3.1.16 in Section 3.1.4 states that there is only one Newton sequence. The following proposition covers the rest of Theorem 3.1.10:

Proposition 3.1.11.
Let $\boldsymbol{f}\colon V \to V$ be a vector of power series. For every Newton approximant $\boldsymbol{\nu}^{(i)}$ there exists a vector $\boldsymbol{\delta}^{(i)}$ such that $\boldsymbol{f}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$. So there is at least one Newton sequence. Moreover, every Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ satisfies

$$\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) \sqsubseteq \boldsymbol{\nu}^{(i+1)} \sqsubseteq \mu\boldsymbol{f} = \bigsqcup_{j\in\mathbb{N}} \boldsymbol{\kappa}^{(j)} \text{ for all } i \in \mathbb{N}. \qquad \diamond$$

The proof of Proposition 3.1.11 is based on two lemmata. The first one, an easy consequence of Kleene's theorem, provides a closed form for the least solution of a linear system of fixed-point equations in terms of the Kleene star operator, defined as follows:

Definition 3.1.12.
Let $\boldsymbol{g}\colon V \to V$ be a monotone map. The map $\boldsymbol{g}^*\colon V \to V$ is defined as $\boldsymbol{g}^*(\boldsymbol{v}) := \sum_{i\in\mathbb{N}} \boldsymbol{g}^i(\boldsymbol{v})$, where $\boldsymbol{g}^0(\boldsymbol{v}) := \boldsymbol{v}$, and $\boldsymbol{g}^{i+1}(\boldsymbol{v}) := \boldsymbol{g}(\boldsymbol{g}^i(\boldsymbol{v}))$ for every $i \geq 0$. Similarly, we set for all $j \in \mathbb{N}$: $\boldsymbol{g}^{\leq j}(\boldsymbol{v}) := \sum_{0\leq i\leq j} \boldsymbol{g}^i(\boldsymbol{v})$. $\diamond$

The existence of $\boldsymbol{g}^*(\boldsymbol{v})$ is guaranteed by the properties of ω-continuous semirings. Observe that $\boldsymbol{v} \sqsubseteq \boldsymbol{g}^*(\boldsymbol{v})$ and $\boldsymbol{g}^*(\boldsymbol{v}) = \boldsymbol{v} + \boldsymbol{g}(\boldsymbol{g}^*(\boldsymbol{v}))$ hold.

Lemma 3.1.13.
Let $\boldsymbol{f}\colon V \to V$ be a vector of power series, and $\boldsymbol{u}, \boldsymbol{v} \in V$. Then the least solution of $D\boldsymbol{f}|_{\boldsymbol{u}}(\boldsymbol{X}) + \boldsymbol{v} = \boldsymbol{X}$ is $D\boldsymbol{f}|^*_{\boldsymbol{u}}(\boldsymbol{v})$. In particular, a Newton sequence from Definition 3.1.9 can be equivalently defined by setting $\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\boldsymbol{0})$ and $\boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)})$. $\diamond$

Proof. Set $\boldsymbol{g}(\boldsymbol{X}) := D\boldsymbol{f}|_{\boldsymbol{u}}(\boldsymbol{X}) + \boldsymbol{v}$. The vector $\boldsymbol{g}$ is a power series in every component and thus a monotone map from V to V. By Kleene's fixed-point theorem, the least solution of $\boldsymbol{g}(\boldsymbol{X}) = \boldsymbol{X}$ is given by $\bigsqcup\{\boldsymbol{g}^i(\boldsymbol{0}) \mid i \in \mathbb{N}\} = \bigsqcup\{D\boldsymbol{f}|^{\leq i}_{\boldsymbol{u}}(\boldsymbol{v}) \mid i \in \mathbb{N}\} = D\boldsymbol{f}|^*_{\boldsymbol{u}}(\boldsymbol{v})$. $\square$

The second lemma, which is interesting by itself, is a generalization of Taylor's theorem to arbitrary ω-continuous semirings.

Lemma 3.1.14.
Let $\boldsymbol{f}\colon V \to V$ be a vector of power series and let $\boldsymbol{u}, \boldsymbol{v}$ be two vectors. We have

$$\boldsymbol{f}(\boldsymbol{u}) + D\boldsymbol{f}|_{\boldsymbol{u}}(\boldsymbol{v}) \sqsubseteq \boldsymbol{f}(\boldsymbol{u}+\boldsymbol{v}) \sqsubseteq \boldsymbol{f}(\boldsymbol{u}) + D\boldsymbol{f}|_{\boldsymbol{u}+\boldsymbol{v}}(\boldsymbol{v}) \,. \qquad \diamond$$

Proof. It suffices to show those inequations for each component separately, so let w.l.o.g. $\boldsymbol{f} = f\colon V \to S$ be a power series. We proceed by induction on the construction of f. The base case (where f is a constant) and the case where f is a sum of polynomials are easy, and so it suffices to consider the case in which f is a monomial. So let

$$f = g \cdot X \cdot a$$

for a monomial g, a variable $X \in \mathcal{X}$ and a constant a. We have

$$f(\boldsymbol{u}) = g(\boldsymbol{u}) \cdot \boldsymbol{u}_X \cdot a \quad \text{and} \quad Df|_{\boldsymbol{u}}(\boldsymbol{v}) = g(\boldsymbol{u}) \cdot \boldsymbol{v}_X \cdot a + Dg|_{\boldsymbol{u}}(\boldsymbol{v}) \cdot \boldsymbol{u}_X \cdot a \,.$$

By induction we obtain:

$$\begin{aligned} f(\boldsymbol{u}+\boldsymbol{v}) &= g(\boldsymbol{u}+\boldsymbol{v}) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &\sqsupseteq \big(g(\boldsymbol{u}) + Dg|_{\boldsymbol{u}}(\boldsymbol{v})\big) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &= g(\boldsymbol{u}) \cdot \boldsymbol{u}_X \cdot a + g(\boldsymbol{u}) \cdot \boldsymbol{v}_X \cdot a + Dg|_{\boldsymbol{u}}(\boldsymbol{v}) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &\sqsupseteq f(\boldsymbol{u}) + g(\boldsymbol{u}) \cdot \boldsymbol{v}_X \cdot a + Dg|_{\boldsymbol{u}}(\boldsymbol{v}) \cdot \boldsymbol{u}_X \cdot a \\ &= f(\boldsymbol{u}) + Df|_{\boldsymbol{u}}(\boldsymbol{v}) \end{aligned}$$

and

$$\begin{aligned} f(\boldsymbol{u}+\boldsymbol{v}) &= g(\boldsymbol{u}+\boldsymbol{v}) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &\sqsubseteq \big(g(\boldsymbol{u}) + Dg|_{\boldsymbol{u}+\boldsymbol{v}}(\boldsymbol{v})\big) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &= g(\boldsymbol{u}) \cdot \boldsymbol{u}_X \cdot a + g(\boldsymbol{u}) \cdot \boldsymbol{v}_X \cdot a + Dg|_{\boldsymbol{u}+\boldsymbol{v}}(\boldsymbol{v}) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &\sqsubseteq f(\boldsymbol{u}) + g(\boldsymbol{u}+\boldsymbol{v}) \cdot \boldsymbol{v}_X \cdot a + Dg|_{\boldsymbol{u}+\boldsymbol{v}}(\boldsymbol{v}) \cdot (\boldsymbol{u}_X + \boldsymbol{v}_X) \cdot a \\ &= f(\boldsymbol{u}) + Df|_{\boldsymbol{u}+\boldsymbol{v}}(\boldsymbol{v}) \end{aligned}$$

□

We can now proceed to prove Proposition 3.1.11.

Proof of Proposition 3.1.11. First we prove for all $i \in \mathbb{N}$ that a suitable $\boldsymbol{\delta}^{(i)}$ exists and, at the same time, that the inequality $\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$ holds. We proceed by induction on i. The base case $i = 0$ is easy. For the induction step, let $i \geq 0$.

$$\begin{aligned} \boldsymbol{\kappa}^{(i+1)} &= \boldsymbol{f}(\boldsymbol{\kappa}^{(i)}) && \text{(definition of } \boldsymbol{\kappa}^{(i)}) \\ &\sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) && \text{(induction: } \boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)}) \\ &= \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)} \text{ for some } \boldsymbol{\delta}^{(i)} && \text{(induction)} \\ &\sqsubseteq \boldsymbol{\nu}^{(i)} + D\boldsymbol{f}|^{*}_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}) && (\boldsymbol{v} \sqsubseteq \boldsymbol{g}^{*}(\boldsymbol{v})) \\ &= \boldsymbol{\nu}^{(i+1)} && \text{(Lemma 3.1.13)} \\ &= \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)} + D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|^{*}_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)})) && (\boldsymbol{g}^{*}(\boldsymbol{v}) = \boldsymbol{v} + \boldsymbol{g}(\boldsymbol{g}^{*}(\boldsymbol{v}))\,) \end{aligned}$$

$$
\begin{aligned}
&= \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) + D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)})) && \text{(definition of } \boldsymbol{\delta}^{(i)}) \\
&\sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)} + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)})) && \text{(Lemma 3.1.14)} \\
&= \boldsymbol{f}(\boldsymbol{\nu}^{(i+1)}) && \text{(Lemma 3.1.13)}
\end{aligned}
$$

Since $\boldsymbol{\nu}^{(i+1)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i+1)})$, there exists a $\boldsymbol{\delta}^{(i+1)}$ such that $\boldsymbol{\nu}^{(i+1)} + \boldsymbol{\delta}^{(i+1)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i+1)})$. Next we prove $\boldsymbol{f}(\boldsymbol{\nu}^{(i)}) \sqsubseteq \boldsymbol{\nu}^{(i+1)}$:

$$
\begin{aligned}
\boldsymbol{f}(\boldsymbol{\nu}^{(i)}) &= \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)} && \text{(as shown above)} \\
&\sqsubseteq \boldsymbol{\nu}^{(i)} + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}) && (\boldsymbol{v} \sqsubseteq \boldsymbol{g}^*(\boldsymbol{v})) \\
&= \boldsymbol{\nu}^{(i+1)} && \text{(Lemma 3.1.13)}
\end{aligned}
$$

It remains to prove $\bigsqcup_{j\in\mathbb{N}} \boldsymbol{\kappa}^{(j)} = \mu\boldsymbol{f}$ and $\boldsymbol{\nu}^{(i)} \sqsubseteq \mu\boldsymbol{f}$ for all i. The equation $\bigsqcup_{j\in\mathbb{N}} \boldsymbol{\kappa}^{(j)} = \mu\boldsymbol{f}$ holds by Kleene's theorem (Theorem 2.2.12). To prove $\boldsymbol{\nu}^{(i)} \sqsubseteq \mu\boldsymbol{f}$ for all i we need a technical lemma.

Lemma 3.1.15.
Let $\boldsymbol{f}(\boldsymbol{x}) \sqsupseteq \boldsymbol{x}$. For all $i \geq 0$ there exists a vector $\boldsymbol{r}^{(i)}(\boldsymbol{x})$ such that

$$
\begin{aligned}
\boldsymbol{f}^d(\boldsymbol{x}) + \boldsymbol{r}^{(i)}(\boldsymbol{x}) &= \boldsymbol{f}^{i+1}(\boldsymbol{x}) \quad \text{and} \\
\boldsymbol{r}^{(i)}(\boldsymbol{x}) &\sqsupseteq D\boldsymbol{f}|_{\boldsymbol{f}^{i-1}(\boldsymbol{x})}(D\boldsymbol{f}|_{\boldsymbol{f}^{i-2}(\boldsymbol{x})}(\ldots D\boldsymbol{f}|_{\boldsymbol{x}}(\boldsymbol{r}^{(0)}(\boldsymbol{x}))\ldots)) \\
&\sqsupseteq D\boldsymbol{f}|^i_{\boldsymbol{x}}(\boldsymbol{r}^{(0)}(\boldsymbol{x}))\ . \qquad \diamond
\end{aligned}
$$

Proof of the lemma. By induction on i. For $i = 0$ there is an appropriate $\boldsymbol{r}^{(0)}(\boldsymbol{x})$ by assumption. Let $d \geq 0$.

$$
\begin{aligned}
\boldsymbol{f}^{i+2}(\boldsymbol{x}) &= \boldsymbol{f}(\boldsymbol{f}^i(\boldsymbol{x}) + \boldsymbol{r}^{(i)}(\boldsymbol{x})) && \text{(induction)} \\
&\sqsupseteq \boldsymbol{f}^{i+1}(\boldsymbol{x}) + D\boldsymbol{f}|_{\boldsymbol{f}^i(\boldsymbol{x})}(\boldsymbol{r}^{(i)}(\boldsymbol{x})) && \text{(Lemma 3.1.14)} \\
&\sqsupseteq \boldsymbol{f}^{i+1}(\boldsymbol{x}) + D\boldsymbol{f}|_{\boldsymbol{f}^i(\boldsymbol{x})}(\ldots D\boldsymbol{f}|_{\boldsymbol{x}}(\boldsymbol{r}^{(0)}(\boldsymbol{x}))\ldots) && \text{(induction)}
\end{aligned}
$$

Therefore, there exists an $\boldsymbol{r}^{(i+1)}(\boldsymbol{x}) \sqsupseteq D\boldsymbol{f}|_{\boldsymbol{f}^i(\boldsymbol{x})}(\ldots D\boldsymbol{f}|_{\boldsymbol{x}}(\boldsymbol{r}^{(0)}(\boldsymbol{x}))\ldots)$. Since $D\boldsymbol{f}|_{\boldsymbol{y}}$ is monotone in $\boldsymbol{y}$ and $\boldsymbol{x} \sqsubseteq \boldsymbol{f}(\boldsymbol{x}) \sqsubseteq \boldsymbol{f}^2(\boldsymbol{x}) \sqsubseteq \ldots$, the second inequality also holds. □

Notice that Lemma 3.1.15 holds for $\boldsymbol{x} = \boldsymbol{\nu}^{(i)}$ and $\boldsymbol{r}^{(0)}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\delta}^{(i)}$, because we have already shown $\boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$. Now we can prove $\boldsymbol{\nu}^{(i)} \sqsubseteq \mu\boldsymbol{f}$ by induction on i. The case $i = 0$ is trivial. Let $i \geq 0$. We have:

$$
\begin{aligned}
\boldsymbol{\nu}^{(i+1)} &= \boldsymbol{\nu}^{(i)} + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}) && \text{(Lemma 3.1.13)} \\
&= \boldsymbol{\nu}^{(i)} + \sum_{d\in\mathbb{N}} D\boldsymbol{f}|^d_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}) && \text{(definition of } D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}) \\
&\sqsubseteq \boldsymbol{\nu}^{(i)} + \sum_{d\in\mathbb{N}} \boldsymbol{r}^{(d)}(\boldsymbol{\nu}^{(i)}) && \text{(Lemma 3.1.15)} \\
&= \bigsqcup_{d\in\mathbb{N}} \boldsymbol{f}^d(\boldsymbol{\nu}^{(i)}) && (\omega\text{-continuity}) \\
&\sqsubseteq \mu\boldsymbol{f} && \text{(induction: } \boldsymbol{f}^d(\boldsymbol{\nu}^{(i)}) \sqsubseteq \mu\boldsymbol{f})
\end{aligned}
$$

This completes the proof of Proposition 3.1.11. □

3.1.4 Uniqueness

In the previous subsection we have seen that, given a vector $\boldsymbol{f}$ of power series, there indeed exists a Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ in the sense of Definition 3.1.9. because for each $\boldsymbol{\nu}^{(i)}$ there exists a $\boldsymbol{\delta}^{(i)}$ such that $\boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)} = \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$. However, recall that the "difference" $\boldsymbol{\delta}^{(i)}$ in Definition 3.1.9 might not be uniquely determined. ([2]) Thus, the Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ could, in principle, depend on the choice of $\boldsymbol{\delta}^{(i)}$. We show now that this is *not* the case, i.e., there is only one Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$, independent of the choice of $\boldsymbol{\delta}^{(i)}$:

Proposition 3.1.16.
Let $\boldsymbol{f} : V \to V$ be a vector of power series. There is exactly one Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$. ◇

Theorem 3.1.10 follows directly by combining Proposition 3.1.11 and Proposition 3.1.16. So for Theorem 3.1.10 it remains to prove Proposition 3.1.16, which we do in the rest of this section.

It is convenient for the proof to introduce *substitutionals*, a notion related to differentials, see the following Proposition 3.1.19.

Definition 3.1.17.
Let f be a power series over an ω-continuous semiring $\mathcal{S}$ and let $s \in \mathbb{N}_+$. The *substitutional of f w.r.t. s* at the point $\boldsymbol{v}$ is the mapping $\$_s f|_{\boldsymbol{v}} : V \to S$ defined as follows:
If f is a monomial, i.e., of the form $f = a_1 X_1 \cdots a_k X_k a_{k+1}$, then

$$\$_s f|_{\boldsymbol{v}}(\boldsymbol{b}) = \begin{cases} a_1 \boldsymbol{v}_{X_1} \cdots a_{s-1} \boldsymbol{v}_{X_{s-1}} a_s \boldsymbol{b}_{X_s} a_{s+1} \boldsymbol{v}_{X_{s+1}} \cdots a_k \boldsymbol{v}_{X_k} a_{k+1} & \text{if } 1 \le s \le k \\ 0 & \text{otherwise.} \end{cases}$$

If f is a power series, i.e., of the form $f = \sum_{i\in I} f_i$, then

$$\$_s f|_{\boldsymbol{v}}(\boldsymbol{b}) = \sum_{i\in I} \$_s f_i|_{\boldsymbol{v}}(\boldsymbol{b}).$$

In words: if f is a monomial with at least s variables then $\$_s f|_{\boldsymbol{v}}(\boldsymbol{b})$ is obtained from f by replacing the s-th variable X_s by $\boldsymbol{b}_{X_s}$ and all other variables by the corresponding component of $\boldsymbol{v}$. If f is a monomial with less than s variables then $\$_s f|_{\boldsymbol{v}}(\boldsymbol{b}) = 0$. If f is a power series then the substitutional of f is the sum of the substitutionals of f's monomials.

[2] For example, consider the language semiring generated by some finite alphabet Σ. For any two languages $A, B \subseteq \Sigma^*$ with $A \subseteq B$ we might take any language D satisfying $B \setminus A \subseteq D \subseteq B$ as a "difference".

Analogously to differentials, we extend the definition of substitutionals to vectors of power series by applying the substitution componentwise. Formally, we define the substitutional of a vector of power series $\boldsymbol{f}$ at $\boldsymbol{v}$ as the function $\$_s\boldsymbol{f}|_{\boldsymbol{v}} : V \to V$ with

$$(\$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}))_X := \$_s\boldsymbol{f}_X|_{\boldsymbol{v}}(\boldsymbol{b}) \; . \qquad \diamond$$

Observe that, like the differential (see Remark 3.1.8), the substitutional is "additive", i.e., $\$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}+\boldsymbol{b}') = \$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}) + \$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}')$.

Definition 3.1.18.
For any $j \in \mathbb{N}$ and any sequence $s = (s_1, \ldots s_j) \in \mathbb{N}_+^j$ we write $\$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b})$ for $\$_{s_1}\boldsymbol{f}|_{\boldsymbol{v}}(\$_{s_2}\boldsymbol{f}|_{\boldsymbol{v}}(\cdots \$_{s_j}\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b})\cdots))$, and $\$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}) = \boldsymbol{b}$ if $j = 0$. $\diamond$

The following proposition states the connection between the differential and the substitutional.

Proposition 3.1.19.
Let f be a monomial. Then

$$D_X f|_{\boldsymbol{v}}(\boldsymbol{b}) = \sum \{\$_s f|_{\boldsymbol{v}}(\boldsymbol{b}) \mid X \text{ is the } s\text{-th variable in } f\} \, .$$

Let $\boldsymbol{f}$ be a vector of power series. Then:

(1) $D\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b}) = \sum_{s\in\mathbb{N}_+} \$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b})$.

(2) $D\boldsymbol{f}|_{\boldsymbol{v}}^j(\boldsymbol{b}) = \sum_{s\in\mathbb{N}_+^j} \$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{b})$.

(3) For all $s \in \mathbb{N}_+$ we have $\boldsymbol{f}(\boldsymbol{v}) \sqsupseteq \$_s\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{v})$. $\diamond$

The proposition roughly says that the differential $D\boldsymbol{f}|_{\boldsymbol{v}}$ can be obtained by replacing in all possible ways all occurrences of variables, except one by the values supplied by $\boldsymbol{v}$:

Example 3.1.20. Consider the polynomial $f = aXYX + cY$. Then

$$\begin{array}{lcl}
\$_1 f|_{\boldsymbol{v}}(\boldsymbol{b}) & = & a\boldsymbol{b}_X\boldsymbol{v}_Y\boldsymbol{v}_X + c\boldsymbol{b}_Y \\
\$_2 f|_{\boldsymbol{v}}(\boldsymbol{b}) & = & a\boldsymbol{v}_X\boldsymbol{b}_Y\boldsymbol{v}_X \\
\$_3 f|_{\boldsymbol{v}}(\boldsymbol{b}) & = & a\boldsymbol{v}_X\boldsymbol{v}_Y\boldsymbol{b}_X \\
D_X f|_{\boldsymbol{v}}(\boldsymbol{b}) & = & a\boldsymbol{b}_X\boldsymbol{v}_Y\boldsymbol{v}_X + a\boldsymbol{v}_X\boldsymbol{v}_Y\boldsymbol{b}_X \\
D_Y f|_{\boldsymbol{v}}(\boldsymbol{b}) & = & a\boldsymbol{v}_X\boldsymbol{b}_Y\boldsymbol{v}_X + c\boldsymbol{b}_Y \; .
\end{array}$$

Observe that $Df|_{\boldsymbol{v}}(\boldsymbol{b}) = D_X f|_{\boldsymbol{v}}(\boldsymbol{b}) + D_Y f|_{\boldsymbol{v}}(\boldsymbol{b}) = \$_1 f|_{\boldsymbol{v}}(\boldsymbol{b}) + \$_2 f|_{\boldsymbol{v}}(\boldsymbol{b}) + \$_3 f|_{\boldsymbol{v}}(\boldsymbol{b})$ and that $f(\boldsymbol{v}) = a\boldsymbol{v}_X\boldsymbol{v}_Y\boldsymbol{v}_X + c\boldsymbol{v}_Y \sqsupseteq \$_s f|_{\boldsymbol{v}}(\boldsymbol{v})$ holds for all $s \in \mathbb{N}_+$. $\diamond$

For the proof of Proposition 3.1.16 we need the following two lemmata.

Lemma 3.1.21.
Let $\boldsymbol{f}$ be a vector of power series. Let $\boldsymbol{\nu} + \boldsymbol{\delta} = \boldsymbol{f}(\boldsymbol{\nu})$. Let $j \in \mathbb{N}$ and $(s_1, \ldots, s_{j+1}) \in \mathbb{N}_+^{j+1}$. Then $\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}) \sqsupseteq \$_{(s_1,\ldots,s_{j+1})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu})$. ◇

Proof. By induction on j. For $j = 0$ we have $\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq 0}(\boldsymbol{\delta}) = \boldsymbol{\nu} + \boldsymbol{\delta} = \boldsymbol{f}(\boldsymbol{\nu}) \sqsupseteq \$_{s_1}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu})$ by Proposition 3.1.19.3. Let $j \geq 0$. We have:

$$\begin{aligned}
\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j+1}(\boldsymbol{\delta}) &= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}) + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{j+1}(\boldsymbol{\delta}) \\
&\sqsupseteq \$_{(s_1,\ldots,s_{j+1})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu}) + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{j+1}(\boldsymbol{\delta}) && \text{(induction)} \\
&\sqsupseteq \$_{(s_1,\ldots,s_{j+1})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu}) + \$_{(s_1,\ldots,s_{j+1})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}) && \text{(Prop. 3.1.19.2.)} \\
&= \$_{(s_1,\ldots,s_{j+1})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{f}(\boldsymbol{\nu})) && (\boldsymbol{\nu} + \boldsymbol{\delta} = \boldsymbol{f}(\boldsymbol{\nu})) \\
&\sqsupseteq \$_{(s_1,\ldots,s_{j+1})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\$_{s_{j+2}}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu})) && \text{(Prop. 3.1.19.3.)} \\
&= \$_{(s_1,\ldots,s_{j+1},s_{j+2})}\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu}) && \square
\end{aligned}$$

Lemma 3.1.22.
Let $\boldsymbol{f}$ be a vector of power series. Let $\boldsymbol{\nu} + \boldsymbol{\delta} = \boldsymbol{\nu} + \boldsymbol{\delta}' = \boldsymbol{f}(\boldsymbol{\nu})$. Then $\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{*}(\boldsymbol{\delta}) = \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{*}(\boldsymbol{\delta}')$. ◇

Proof. We show $\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}) = \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}')$ for all $j \in \mathbb{N}$. Then the lemma follows by ω-continuity. We proceed by induction on j. The induction base ($j = 0$) is clear. Let $j \geq 0$. We immediately obtain:

$$\begin{aligned}
\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j+1}(\boldsymbol{\delta}) &= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}) + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{j+1}(\boldsymbol{\delta}) \\
&= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}') + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{j+1}(\boldsymbol{\delta}) && \text{(induction)} \\
&= \underbrace{\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}')}_{=:\boldsymbol{u}} + \sum_{s \in \mathbb{N}_+^{j+1}} \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}) && \text{(Prop. 3.1.19.2.)}
\end{aligned}$$

With $\boldsymbol{u} := \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}')$, it follows from Lemma 3.1.21 that $\boldsymbol{u} \sqsupseteq \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu})$ holds for all $s \in \mathbb{N}_+^{j+1}$. In other words, for all $s \in \mathbb{N}_+^{j+1}$ there is a $\boldsymbol{u}'$ such that $\boldsymbol{u} = \boldsymbol{u}' + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu})$. Hence, for all $s \in \mathbb{N}_+^{j+1}$, we have

$$\begin{aligned}
\boldsymbol{u} + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}) &= \boldsymbol{u}' + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu}) + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}) \\
&= \boldsymbol{u}' + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{f}(\boldsymbol{\nu})) \\
&= \boldsymbol{u}' + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\nu}) + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}') \\
&= \boldsymbol{u} + \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}').
\end{aligned}$$

One might say that $\boldsymbol{u}$ acts like a "catalyst" in the above equation, as we can replace $\boldsymbol{\delta}$ by $\boldsymbol{\delta}'$ due to the "presence" of $\boldsymbol{u}$. With this at hand, we may continue with the induction step:

$$\begin{aligned}
\boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j+1}(\boldsymbol{\delta}) &= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}') + \sum_{s \in \mathbb{N}_+^{j+1}} \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}) \\
&= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}') + \sum_{s \in \mathbb{N}_+^{j+1}} \$_s\boldsymbol{f}|_{\boldsymbol{\nu}}(\boldsymbol{\delta}') && \text{(as argued above)} \\
&= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j}(\boldsymbol{\delta}') + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{j+1}(\boldsymbol{\delta}') && \text{(Prop. 3.1.19.2.)} \\
&= \boldsymbol{\nu} + D\boldsymbol{f}|_{\boldsymbol{\nu}}^{\leq j+1}(\boldsymbol{\delta}') && \square
\end{aligned}$$

Now Proposition 3.1.16 follows immediately from Lemma 3.1.22 by a straightforward inductive proof.

3.2 Derivation Trees and the Newton Approximants

In the previous section we used the relation between substitutions and differentiation to show the uniqueness of the Newton sequence. In this section we study the relation between the Newton sequence, substitutions and differentiation in more detail. For this we will use the concept of derivation tree for representing sequences of substitutions as known from formal language theory. For instance, in the case of a context-free grammar a derivation tree describes how a word is obtained via a finite number of substitutions starting from the axiom of the grammar. In a similar way as context-free grammars and derivation trees are related to each other, we associate with every system of power series sets of derivation trees. We have already sketched this idea in Example 1.1.2.

Our main goal is then the characterization of the Kleene and Newton approximants by means of derivation trees. In the case of the Kleene approximants $\boldsymbol{\kappa}^{(i)}$ the corresponding set of derivation trees can easily be characterized by means of the height of a tree. Regarding the Newton approximants, we introduce the notion of *dimension* of a tree, see Definition 3.2.9 below. We show that the purely graph theoretical notion of tree dimension exactly describes the Newton approximants, i.e., the k-th Newton approximant coincides with the derivation trees of dimension at most k. This result is used in subsequent sections. In Section 3.3.1 the characterization of the Newton approximants by means of the tree dimension allows us to show that our generalization of Newton's method and the notion of languages of *finite index* [Ynt67] are deeply connected; in Section 3.4 we show that the Newton sequence reaches $\mu\boldsymbol{f}$ already after n steps (with n the number of variables) which in turn allows us to improve a result by Hopkins and Kozen [HK99].

For the rest of the section we fix a vector $\boldsymbol{f}$ of power series over a fixed but arbitrary ω-continuous semiring. Without loss of generality, we assume that $\boldsymbol{f}_X = \sum_{j\in J} m_{X,j}$ holds for every variable $X \in \mathcal{X}$, i.e., we assume that for all variables the sum is over the same countable set J of indices.

Consider the set of ordered trees whose nodes are labeled by pairs (X, j), where $X \in \mathcal{X}$ and $j \in J$. For convenience, we often identify a tree and its

root. In particular, we say that a tree t is labeled by (X, j) if its root is labeled by (X, j). The mappings λ, λ_v and λ_m are defined by $\lambda(t) := (X, j)$, $\lambda_v(t) := X$, and $\lambda_m(t) := j$. Given a set T of trees, we denote by T_X the set of trees $t \in T$ such that $\lambda_v(t) = X$.

We define the set of derivation trees of $\boldsymbol{f}$, and show how to assign to each tree a semiring element called the yield of the tree. For technical reasons, our definition differs slightly from the straightforward generalization of derivation trees for grammars.

Definition 3.2.1 (derivation tree, yield)**.**
The *derivation trees* of $\boldsymbol{f}$ and their *yields* are inductively defined as follows:

- For every monomial $m_{X,j}$ of $\boldsymbol{f}_X$, if no variable occurs in $m_{X,j}$, then the tree t consisting of one single node labeled by (X, j) is a derivation tree of $\boldsymbol{f}$. Its yield $\mathsf{Y}(t)$ is equal to $m_{X,j}$.
- Let $m_{X,j} = a_1 X_1 a_2 X_2 \dots a_k X_k a_{k+1}$ for some $k \geq 1$, and let $t_1, \dots, t_k$ be derivation trees of $\boldsymbol{f}$ such that $\lambda_v(t_i) = X_i$ for $1 \leq i \leq k$. Then the tree t labeled by (X, j) and having $t_1, \dots, t_k$ as (ordered) children is also a derivation tree of $\boldsymbol{f}$, and its yield $\mathsf{Y}(t)$ is equal to $a_1 \mathsf{Y}(t_1) \dots a_k \mathsf{Y}(t_k) a_{k+1}$.

We call an derivation tree t of $\boldsymbol{f}$ an *X-tree* if $\lambda_v(t) = X$ ($X \in \mathcal{X}$). The set of all X-trees is denoted by $\mathcal{T}_X^{\boldsymbol{f}}$. We simply write $\mathcal{T}^{\boldsymbol{f}}$ for all derivation trees associated with $\boldsymbol{f}$. If $\boldsymbol{f}$ is given by the context, then we drop the superscript.

The *yield* $\mathsf{Y}(T)$ of a countable set T of derivation trees is defined by

$$\mathsf{Y}(T) = \sum_{t \in T} \mathsf{Y}(t).$$

In the following, we mean *derivation tree* whenever we say *tree*. ◇

Example 3.2.2. Figure 3.1(a) shows a system of equations and Figure 3.1(b) a derivation tree associated with it. Consider the node labeled by $(Y, 1)$ (the right child of the root). Since the first monomial of the equation for Y is cYZ, the node has two children, say n_1, n_2 with $\lambda_v(n_1) = Y$ and $\lambda_v(n_2) = Z$. As $\lambda_m(n_2) = 2$, the children of n_2 are determined by the second monomial of the equation for Z. Since this monomial is constant, n_2 has no children. The Figure 3.1(c) shows the result of labeling each node of the tree with the yield of the subtree rooted at it. ◇

Remark 3.2.3.
As we assume that $\boldsymbol{f}_X = \sum_{j \in J} m_{X,j}$ for some fixed index set J and all variables X, we formally have to pad polynomials by adding monomials $m_{X,j} = 0$. Hence, derivation trees might have yield 0. This allows for a more convenient notation in the proofs to follow. Aside from this purely technical aspect, these trees can be neglected for determining $\mu\boldsymbol{f}$. ◇

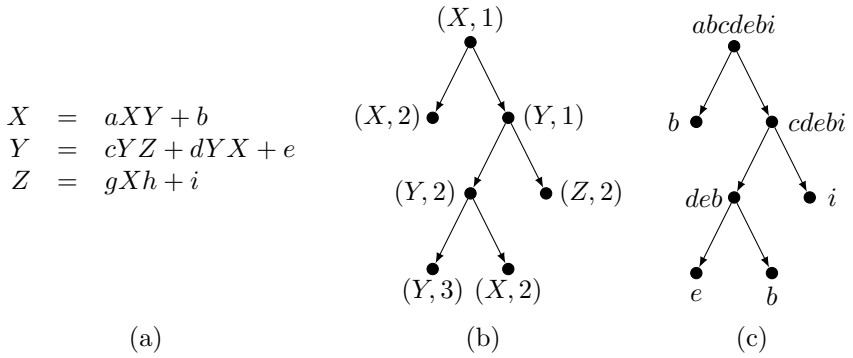


Figure 3.1: (a) A system of equations, (b) a derivation tree associated with it, (c) and (the recursive calculation of) its yield.

3.2.1 Kleene Sequence and Height

As a warm-up for the Newton case, we characterize the Kleene sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ in terms of the derivation trees of a certain height.

Definition 3.2.4 (height)**.**
Let t be a derivation tree. The *height* of t, denoted by $h(t)$, is the length (number of edges) of a longest path from the root to some leaf. We denote by $\mathcal{H}^i$ the set of derivation trees of height at most i. ◇

Proposition 3.2.5.
$\left(\boldsymbol{\kappa}^{(i)}\right)_X = \mathsf{Y}(\mathcal{H}^i_X)$, i.e., the X-component of the i-th Kleene approximant $\boldsymbol{\kappa}^{(i)}$ is equal to the yield of $\mathcal{H}^i_X$. ◇

Remark 3.2.6.
Notice that Proposition 3.2.5 no longer holds if nodes are only labeled with a variable, and not with a pair. Consider for instance the equation $X = a + a$, for which $\kappa^{(0)} = a + a$. There are two derivation trees t_1, t_2 of height 0, both consisting of one single node: t_1 is labeled by $(X, 1)$, and t_2 by $(X, 2)$. We get $\mathsf{Y}(t_1) + \mathsf{Y}(t_2) = a + a = \kappa^{(0)}$. If we labeled nodes only with variables, then there would be one single derivation tree t, and we would get $\mathsf{Y}(t) = a$, which in general is different from $a + a$. ◇

Example 3.2.7. Consider again the equation $X = 1/2 \cdot X^2 + 1/2$ over the real semiring. We have $\kappa^{(2)} = 89/128$. Figure 3.2 shows the five derivation trees of height at most 2. It is easy to see that their yields are $1/2, 1/8, 1/32, 1/32, 1/128$, which add up to $89/128$. ◇

By Kleene's theorem we obtain that the least solution of the equation system is equal to the yield of the set of all trees.

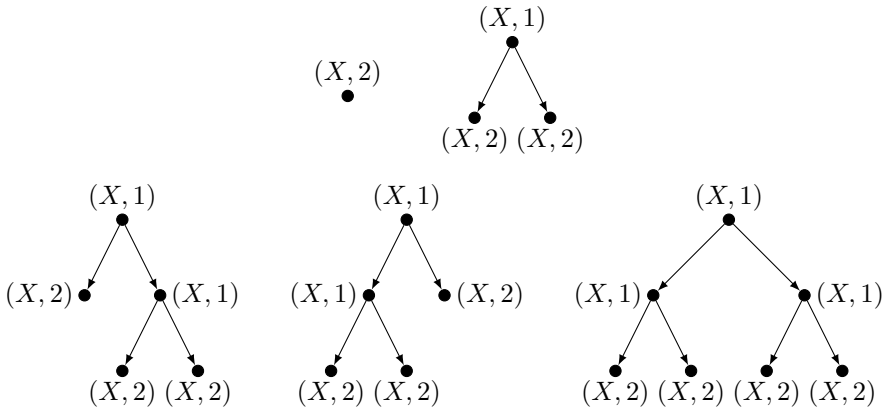


Figure 3.2: Trees of height at most 2 for the equation $X = 1/2 \cdot X^2 + 1/2$.

Corollary 3.2.8.
For all $X \in \mathcal{X}$: $(\mu \boldsymbol{f})_X = \mathsf{Y}(\mathcal{T}_X)$. ◇

Proof. By Kleene's Theorem (Theorem 2.2.12) we have $(\mu \boldsymbol{f})_X = \bigsqcup_{i \in \mathbb{N}} (\boldsymbol{\kappa}^{(i)})_X$. The result then follows from Proposition 3.2.5. □

3.2.2 Newton Sequence and Dimension

We introduce a second parameter of a tree, namely its *dimension*. Like the height, it depends only on the tree structure, and not on the labels of its nodes. Loosely speaking, a tree has dimension 0 if it consists of just one node; a tree has dimension i if there is a path from its root to some node which has at least 2 children with dimension $i - 1$ and all subtrees of the path that are not themselves on the path have dimension at most $i - 1$. The path is called the *backbone* of the tree. Figure 3.3 illustrates this idea.

Formally, we use an inductive definition of dimension that is more convenient for proofs.

Definition 3.2.9 ((dimension))**.**
The *dimension* $d(t)$ of a tree t is inductively defined as follows: If t has no children, we set $d(t) := 0$. Otherwise let d be the maximal dimension of a child of t, and let k be the number of children of t which have exactly dimension d. We set $d(t) := d + 1$ if $k > 1$, and $d(t) := d$ otherwise.

We denote by $\mathcal{D}^i$ the set of derivation trees of dimension at most i. ◇

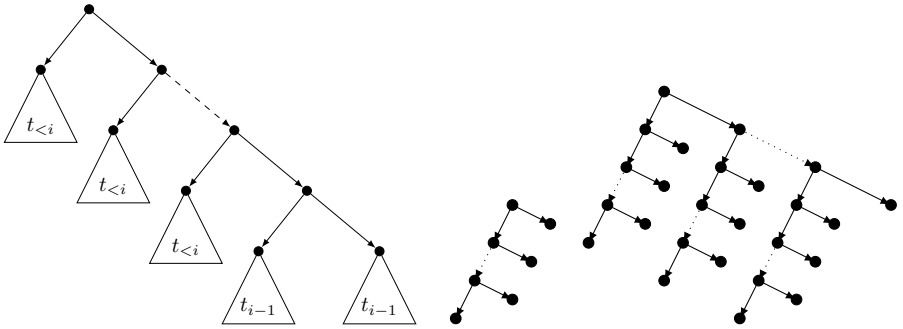


Figure 3.3: (a) shows the general structure of a tree of dimension i, where $t_{<i}$ (resp. t_{i-1}) represents any tree of dimension $< i$ (resp. $= i-1$). (b) and (c) give some idea of the topology of one-, resp. two-dimensional trees.

Remark 3.2.10.
It is easy to prove by induction that $h(t) \geq d(t)$ holds for every derivation tree t. In particular, the trees of height 0 and the trees of dimension 0 coincide. ⋄

In the rest of the section we show that the i-th Newton approximant $\boldsymbol{\nu}^{(i)}$ is equal to the yield of the derivation trees of dimension at most i:

Theorem 3.2.11 (Tree Characterization of the Newton Sequence)**.**
Let $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ be the Newton sequence of $\boldsymbol{f}$. For every $X \in \mathcal{X}$ and every $i \geq 0$ we have $\left(\boldsymbol{\nu}^{(i)}\right)_X = \mathsf{Y}(\mathcal{D}^i_X)$, i.e., the X-component of the i-th Newton approximant is equal to the yield of $\mathcal{D}^i_X$. ⋄

The proof is as follows. We define, in terms of trees, a sequence $(\boldsymbol{\tau}^{(i)})_{i\in\mathbb{N}}$ satisfying $\boldsymbol{\tau}^{(i)}_X = \mathsf{Y}(\mathcal{D}^i_X)$ (Lemma 3.2.14), and we prove that it is a Newton sequence (Lemma 3.2.15). As the Newton sequence is unique by Proposition 3.1.16, we have $\boldsymbol{\tau}^{(i)} = \boldsymbol{\nu}^{(i)}$ and Theorem 3.2.11 follows. For this, we need the following definition.

Definition 3.2.12.
A tree t is *proper* if $d(t) > d(t')$ for every child t' of t. For every $i \geq 0$, let P^i be the set of proper trees of dimension i, and set $\boldsymbol{\delta}^{(i)}_X := \mathsf{Y}(P^{i+1}_X)$ for all $X \in \mathcal{X}$. Then the sequence $(\boldsymbol{\tau}^{(i)})_{i\in\mathbb{N}}$ is defined by $\boldsymbol{\tau}^{(0)} := \boldsymbol{f}(\mathbf{0})$ and

$$\boldsymbol{\tau}^{(i+1)} := \boldsymbol{\tau}^{(i)} + D\boldsymbol{f}|^*_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)}) \text{ for all } i \geq 0. \qquad \diamond$$

Remark 3.2.13.
Note that $P^0_X = \mathcal{D}^0_X = \mathcal{H}^0_X$. ⋄

Lemma 3.2.14.
For every variable $X \in \mathcal{X}$ and every $i \geq 0$: $\boldsymbol{\tau}_X^{(i)} = \mathsf{Y}(\mathcal{D}_X^i)$. ◇

Lemma 3.2.15.
The sequence $(\boldsymbol{\tau}^{(i)})_{i\in\mathbb{N}}$ is a Newton sequence as defined in Definition 3.1.9, i.e., the $\boldsymbol{\delta}^{(i)}$ of Definition 3.2.12 satisfy $\boldsymbol{f}(\boldsymbol{\tau}^{(i)}) = \boldsymbol{\tau}^{(i)} + \boldsymbol{\delta}^{(i)}$. ◇

The proofs of Lemma 3.2.14 and Lemma 3.2.15 are technically involved, and can be found in Appendix A.1. Here we briefly sketch the main ideas of these proofs by means of an example:

Example 3.2.16. Recall the abstract dataflow equations describing the recursive program depicted in Figure 1.3:

$$\begin{aligned} X &= a \cdot X \cdot Y + b \\ Y &= c \cdot X + d \cdot Y. \end{aligned}$$

As usual, we write $\boldsymbol{f}(X, Y)$ for the right-hand side. We then have for every $\boldsymbol{v} \in V$:

$$D\boldsymbol{f}|_{\boldsymbol{v}}(\boldsymbol{X}) = \begin{pmatrix} a \cdot \boldsymbol{v}_X \cdot Y + a \cdot X \cdot \boldsymbol{v}_Y \\ c \cdot X + d \cdot Y \end{pmatrix}.$$

We instantiate Definition 3.1.9 for this $\boldsymbol{f}$. As $\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\mathbf{0})$, we have $\boldsymbol{\nu}_X^{(0)} = b$ and $\boldsymbol{\nu}_Y^{(0)} = 0$. By Proposition 3.2.5, $\boldsymbol{\nu}_X^{(0)}$, resp. $\boldsymbol{\nu}_Y^{(0)}$ is exactly the yield of the X-, resp. Y-trees of height 0. The reader can easily check that this is the case, as there is exactly one X-tree of height 0 associated with the system; and, no Y-tree of height 0 exists (neglecting trees of yield 0, see Remark 3.2.3).

We now try to get an idea which trees correspond to $\boldsymbol{\nu}^{(1)}$. By definition we have

$$\boldsymbol{\nu}^{(1)} = \boldsymbol{\nu}^{(0)} + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(0)}}(\boldsymbol{\delta}^{(0)}) \text{ with } \boldsymbol{\nu}^{(0)} + \boldsymbol{\delta}^{(0)} = \boldsymbol{f}(\boldsymbol{\nu}^{(0)}).$$

We already have characterized $\boldsymbol{\nu}^{(0)}$ by means of derivation trees associated with $\boldsymbol{f}$. So, it remains to consider $D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(0)}}(\boldsymbol{\delta}^{(0)})$.

Recall that $\boldsymbol{\delta}^{(0)}$ is only required to satisfy $\boldsymbol{\nu}^{(0)} + \boldsymbol{\delta}^{(0)} = \boldsymbol{f}(\boldsymbol{\nu}^{(0)})$. Now, as $\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\mathbf{0})$, we have that $\boldsymbol{f}(\boldsymbol{\nu}^{(0)}) = \boldsymbol{\kappa}^{(0)}$, which by Proposition 3.2.5 corresponds to the trees $\mathcal{H}^1$ of height at most 1. A natural choice for $\boldsymbol{\delta}_X^{(0)}$ is thus the yield of the trees $\mathcal{H}_X^1 \setminus \mathcal{H}_X^0$. The reader can check that this corresponds to the set P_X^1 in Definition 3.2.12. Similarly, the proof of Lemma 3.2.15 relies on characterizing the trees yielding $\boldsymbol{f}(\boldsymbol{\nu}^{(i)})$, and removing those yielding already $\boldsymbol{\nu}^{(i)}$.

We turn to $D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(0)}}(\boldsymbol{\delta}^{(0)})$, which is by definition the least solution of the linear system

$$\begin{aligned} X &= a \cdot \boldsymbol{\nu}_X^{(0)} \cdot Y + a \cdot X \cdot \boldsymbol{\nu}_Y^{(0)} + \boldsymbol{\delta}_X^{(0)} \\ Y &= c \cdot X + d \cdot Y + \boldsymbol{\delta}_Y^{(0)}. \end{aligned} \tag{3.13}$$

Assume for a moment that besides the coefficients $\{a, b, c, d\}$ of $\boldsymbol{f}$, also $\{\boldsymbol{\nu}_X^{(0)}, \boldsymbol{\nu}_Y^{(0)}, \boldsymbol{\delta}_X^{(0)}, \boldsymbol{\delta}_Y^{(0)}\}$ are distinct semiring elements, i.e., we move for a short moment to the free semiring generated by all these eight symbols. Here, every derivation tree associated with Equation 3.13 is obviously a chain as every inner node has exactly one child.

The crucial point is that from every such X-, resp. Y-tree associated with Equation 3.13 we obtain an X-, resp. Y-tree by replacing every occurrence of (a monomial corresponding to) a symbol of $\{\boldsymbol{\nu}_X^{(0)}, \boldsymbol{\nu}_Y^{(0)}, \boldsymbol{\delta}_X^{(0)}, \boldsymbol{\delta}_Y^{(0)}\}$ by some tree associated with the respective symbol. This is shown in the following figure where we label nodes directly by the monomial itself as every monomial occurs at most once in $\boldsymbol{f}$, resp. Equation 3.13:

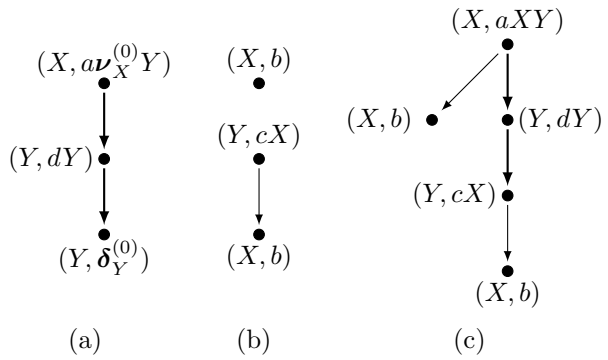


Figure 3.4

Part (a) of Figure 3.4 shows an X-tree of Equation 3.13. In (b) the trees (w.r.t. $\boldsymbol{f}$) yielding $\boldsymbol{\nu}_X^{(0)}$, resp. $\boldsymbol{\delta}_Y^{(0)}$ are shown. We obtain the tree shown in (c) as follows: First recall that the monomial $a\boldsymbol{\nu}_X^{(0)}Y$ of Equation 3.13 originates from the monomial aXY of $\boldsymbol{f}$ via the differential; we relabel the root of (a) accordingly; we then may add the upper tree of (b) with yield $\boldsymbol{\nu}_X^{(0)}$ as the (unique) X-child; similarly, we replace the leaf labeled by $(Y, \boldsymbol{\delta}_Y^{(0)})$ directly with the tree lower tree of (b). The reader can check that the tree of (c) is indeed a derivation tree w.r.t. $\boldsymbol{f}$. In particular, the tree of (c) has dimension 1 with its backbone resulting from the tree of (a). In a similar manner, all trees (w.r.t. $\boldsymbol{f}$) yielding $\boldsymbol{\nu}^{(1)}$ can be obtained, and all of them have dimension 1.

The proof of Lemma 3.2.14 builds up on this idea of obtaining the trees yielding $\boldsymbol{\nu}^{(i+1)}$ by means of substituting the trees yielding $\boldsymbol{\nu}^{(i)}$, resp. $\boldsymbol{\delta}^{(i)}$ into the trees w.r.t. the linear system determined by the differential. In particular it is shown that this yields exactly the trees of dimension $i+1$. ◇

3.3 Idempotent Semirings

In this and the next section we focus on io-semirings, i.e., ω-continuous semiring whose summation operator is idempotent. Here, the natural order can be characterized as follows: $a \sqsubseteq b$ holds if and only if $a + b = b$ ([3]). This extends analogously to vectors.

[3]By definition, we have $a \sqsubseteq b$ if there is a d such that $a + D = b$. This implies $a + b = a + a + d = a + d = b$.

The following proposition shows that the definition of the Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ can be simplified in the idempotent case.

Proposition 3.3.1.
Let $\boldsymbol{f}$ be a vector of power series over an io-semiring. The Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ of $\boldsymbol{f}$ satisfies the following equations for all $i \in \mathbb{N}$:

(a) $\boldsymbol{\nu}^{(i+1)} = D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)}))$

(b) $\boldsymbol{\nu}^{(i+1)} = D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)})$

(c) $\boldsymbol{\nu}^{(i+1)} = D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0}))$ ◇

Proof. We first show (a). By Theorem 3.1.10 we have $\boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$, hence with idempotence $\boldsymbol{\nu}^{(i)} + \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$. So we can choose $\boldsymbol{\delta}^{(i)} = \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$ and have $\boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)})) = D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)}))$, because $\boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) \sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)}))$. So (a) is shown.

Again by Theorem 3.1.10 we have $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{\nu}^{(0)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$. So we have $D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0})) \sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) \sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)}))$. Hence, for (b) and (c), it remains to show $D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)})) \sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)})$ and $D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) \sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0}))$, respectively. For (b) we have:

$$
\begin{aligned}
&D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{\nu}^{(i)})) \\
&\sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)})) && \text{(Lemma 3.1.14)} \\
&= D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0})) + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)})) \\
&\sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)})) && (\boldsymbol{f}(\boldsymbol{0}) \sqsubseteq \boldsymbol{\nu}^{(i)}) \\
&\sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) + D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) && \text{(Lemma 3.1.13)} \\
&= D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) && \text{(idempotence)}
\end{aligned}
$$

So (b) is shown.

For (c) it remains to show $D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) \sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0}))$. We proceed by induction on i. The base case $i = 0$ is easy because $\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\boldsymbol{0})$. Let $i \geq 1$. We have:

$$
\begin{aligned}
&D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)}) \\
&= D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i-1)}}(\boldsymbol{\nu}^{(i-1)})) && \text{(by (b))} \\
&\sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i-1)}}(\boldsymbol{f}(\boldsymbol{0}))) && \text{(by induction)} \\
&\sqsubseteq D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0}))) && \text{(Theorem 3.1.10: } \boldsymbol{\nu}^{(i-1)} \sqsubseteq \boldsymbol{\nu}^{(i)}) \\
&= D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{f}(\boldsymbol{0})) && \text{(see explanation below)}
\end{aligned}
$$

For the last step we used that in the idempotent case we have $\boldsymbol{g}^*(\boldsymbol{g}^*(\boldsymbol{x})) = \boldsymbol{g}^*(\boldsymbol{x})$ for any linear map $\boldsymbol{g} : V \to V$. Recall that Remark 3.1.8 states that $D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}$ is additive.

$$
\boldsymbol{g}^*(\boldsymbol{g}^*(\boldsymbol{x})) = \sum_{j\in\mathbb{N}} \boldsymbol{g}^j \left(\sum_{k\in\mathbb{N}} \boldsymbol{g}^k(\boldsymbol{x}) \right) \qquad \text{(Definition 3.1.12)}
$$

$$
\begin{aligned}
&= \sum_{j\in\mathbb{N}} \sum_{k\in\mathbb{N}} \boldsymbol{g}^j(\boldsymbol{g}^k(\boldsymbol{x})) && \text{(linearity)}\\
&= \sum_{l\in\mathbb{N}} \boldsymbol{g}^l(\boldsymbol{x}) && \text{(idempotence)}\\
&= \boldsymbol{g}^*(\boldsymbol{x}) && \text{(Definition 3.1.12)}
\end{aligned}
$$

This concludes the proof. □

3.3.1 Language Semirings

Now we consider language semirings, the typical example of idempotent semirings. Let $\mathcal{S}_\Sigma$ be the language semiring over a finite alphabet Σ. Let $\boldsymbol{f}$ be a vector of polynomials over $\mathcal{X}$ whose coefficients are elements of Σ. Then, for each $X_0 \in \mathcal{X}$, there is a naturally associated context-free grammar $G_{\boldsymbol{f},X_0} = (\mathcal{X}, \Sigma, P, X_0)$, where the set of productions is

$$P = \{X \to m \mid m \text{ is a monomial of } \boldsymbol{f}_X\}.$$

We write $L(G_{\boldsymbol{f},X_0})$ for the language represented by this grammar. In particular, we have $L(G_{\boldsymbol{f},X_0}) = \big(\mu\boldsymbol{f}\big)_{X_0}$. This follows directly from Proposition 3.2.5 as the derivation X_0-trees w.r.t. $\boldsymbol{f}$ are in one-to-one correspondence with the derivation trees associated with $G_{\boldsymbol{f},X_0}$. Analogously, each grammar is naturally associated with a vector of polynomials. In the following we use grammars and vectors of polynomials interchangeably.

We show in this section that the Newton approximants $\boldsymbol{\nu}^{(i)}$ are strongly linked with the *finite-index* approximations of $L(G)$. Finite-index languages have been extensively investigated under different names by Salomaa, Gruska, Yntema, Ginsburg and Spanier, among others [Sal69, Gru71, Ynt67, GS68] (see [FH97] for historical background).

Definition 3.3.2.
Let G be a grammar, and let D be a derivation $X_0 = \alpha_0 \Rightarrow \cdots \Rightarrow \alpha_r = w$ of $w \in L(G)$. For every $i \in \{0, \ldots, r\}$ let β_i be the projection of α_i onto the variables of G. The *index* of D is the maximum of $\{|\beta_0|, \ldots, |\beta_r|\}$. The *index-$i$ approximation* of $L(G)$, denoted by $L_i(G)$, contains the words derivable by some derivation of G of index at most i. ◇

Example 3.3.3. Consider the following context-free grammar G:

$$X \;\to\; aXY \mid b \qquad Y \;\to\; cX \mid dY.$$

Assume that X is the axiom. Two possible derivations of the word *abcabcb* are:

$$X \Rightarrow aXY \Rightarrow \left\{ \begin{array}{l} abY \Rightarrow abcX \\ aXcX \Rightarrow aXcaXY \end{array} \right\} \Rightarrow abcaXY \Rightarrow abcabY \Rightarrow abcabcX \Rightarrow abcabcb.$$

The upper derivation has index 2, while the lower has index 3. We have already described in Example 3.1.5 how one can construct from G a grammar representing $L_i(G)$. $\diamond$

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

Theorem 3.3.4.
Let $G = (\mathcal{X}, \Sigma, P, X_0)$ be a context-free grammar in CNF and let $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ be the Newton sequence associated with G. Then $\big(\boldsymbol{\nu}^{(i)}\big)_{X_0} = L_{i+1}(G)$ for every $i \geq 0$. $\diamond$

Proof sketch. The proof builds on the tree-dimension characterization of the Newton approximants (Theorem 3.2.11). Given a tree of dimension i we can recursively obtain a derivation of index at most $i+1$ by processing a subtree of minimal dimension always first. For the other direction it can be similarly shown that a derivation of index $i+1$ corresponds to a derivation tree of dimension at most i. $\square$

In particular, it follows from Theorem 3.3.4 that the (X_0-component of the) Newton sequence for a context-free grammar G converges in finitely many steps if and only if $L(G) = L_i(G)$ for some $i \in \mathbb{N}$.

3.4 Commutative Idempotent Semirings

In this section we study Newton's method on cio-semirings, i.e., ω-continuous semirings which do not only have an idempotent addition (as in the previous section), but also a *commutative* multiplication. The counting semirings $\mathcal{C}_k$ are a prominent example of cio-semirings.

In the previous section we have seen that, even though the Newton sequence accelerates the Kleene sequence, it does not generally converge in finitely many steps not even on io-semirings: The language semirings are examples of idempotent ω-continuous semirings, but the Newton sequence of a context-free grammar G with start symbol X_0 does not reach $(\mu\boldsymbol{f})_{X_0} = L(G)$ after finitely many steps, unless $L(G)$ has finite index.

An instance of the Newton sequence in a cio-semiring has already been presented in the counting semiring example on page 38. We show another one here.

Example 3.4.1. Let $\langle 2^{\{a\}^*}, +, \cdot, 0, 1\rangle$ denote the cio-semiring $\langle 2^{\{a\}^*}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$. The multiplication $\cdot$ is meant to be commutative. For simplicity, we write a^i instead of $\{a^i\}$.

Consider $\boldsymbol{f}(X_1, X_2) = (X_2^2 + a,\ X_1^2)$. We have:

$$D\boldsymbol{f}|_{(v_1,v_2)}(X_1, X_2) = \big(v_2 X_2,\ v_1 X_1\big)$$

and

$$D\boldsymbol{f}|^*_{(v_1,v_2)}(X_1, X_2) = (v_1 v_2)^*\big(X_1 + v_2 X_2,\ v_1 X_1 + X_2\big)\ .$$

The first three elements of the Newton sequence are:

$$\boldsymbol{\nu}^{(0)} = (a, 0), \quad \boldsymbol{\nu}^{(1)} = (a, a^2), \quad \boldsymbol{\nu}^{(2)} = (a^3)^*(a, a^2)\ .$$

It is easy to check that $\boldsymbol{\nu}^{(2)}$ is a fixed point of $\boldsymbol{f}$. Hence we have $\boldsymbol{\nu}^{(2)} = \mu\boldsymbol{f}$, as $\boldsymbol{\nu}^{(2)} \sqsubseteq \mu\boldsymbol{f}$ by Theorem 3.1.10. ◇

As shown in Proposition 3.3.1 the Newton sequence has several equivalent descriptions. In particular, the description given in Proposition 3.3.1 exactly corresponds with the sequence proposed in [HK99] ([4]). In this article, the authors discussed the problem of calculating the least fixed point of a system of regular expressions on a commutative Kleene algebra. For now it suffices to know that every cio-semiring is also a commutative Kleene algebra. One of their main results can then be stated for cio-semirings as follows:

Theorem 3.4.2 ([HK99])**.**
Let $\boldsymbol{f}$ be a vector of power series over a cio-semiring induced by regular expressions over the variables $\mathcal{X}$ with $n := |\mathcal{X}|$. There is a function $P : \mathbb{N} \to \mathbb{N}$ with $P(n) \in \mathcal{O}(3^n)$ such that $\boldsymbol{\nu}^{(P(n))} = \mu\boldsymbol{f}$. ◇

This means that the Newton sequence always reaches $\mu\boldsymbol{f}$ after finitely many steps.

In Section 3.4.1 we improve Theorem 3.4.2 by showing that it holds with $P(n) = n$ for all vectors of power series on cio-semirings.

In Section 3.4.2 we discuss the relationship between commutative Kleene algebras and cio-semirings in more detail. In particular, we lift our result on the convergence speed of Newton's method on cio-semirings to commutative Kleene algebras, thereby improving the result of [HK99].

3.4.1 Analysis of the Convergence Speed

In this section we analyze how many steps the Newton iteration and, equivalently, the Hopkins-Kozen iteration need to reach $\mu\boldsymbol{f}$ when we consider cio-semirings.

[4] The connection to Newton's sequence was not stated there.

Recall from Section 3.2 the concept of derivation trees (short: trees). A tree t has a height $h(t)$, a dimension $d(t)$, and a yield $Y(t)$. We define yet another tree property.

Definition 3.4.3.
Let t be a tree. We denote by $L(t)$ the number of distinct λ_v-labels in t. We call t *compact* if $d(t) \leq L(t)$. ◇

Now we are ready to prove the key lemma of this section, which states that any tree can be made compact.

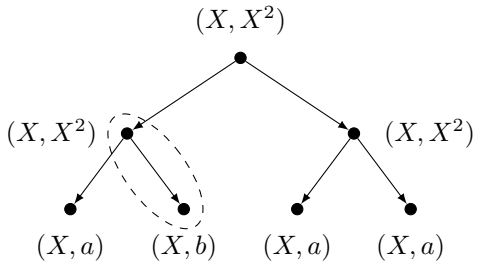
Lemma 3.4.4.
For each tree t there is a compact tree t' with $\lambda_v(t) = \lambda_v(t')$ and $Y(t) = Y(t')$. ◇

We first sketch the proof of the preceding lemma by means of an example:

Example 3.4.5. Consider the following univariate polynomial equation system where $\boldsymbol{f}$ again denotes the right-hand side:

$$X = X^2 + a + b.$$

From $\boldsymbol{f}$ we obtain the following X-tree t: ([5])

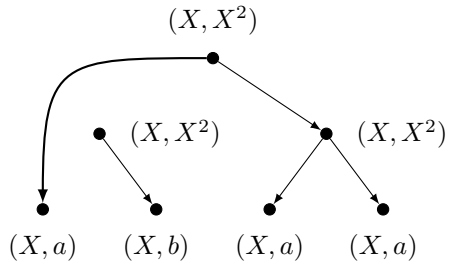


This tree has dimension 2 and is therefore not compact by definition. In order to make it compact, we have to transform it into a derivation tree of $\boldsymbol{f}$ of dimension 1 while preserving its yield up to commutativity.

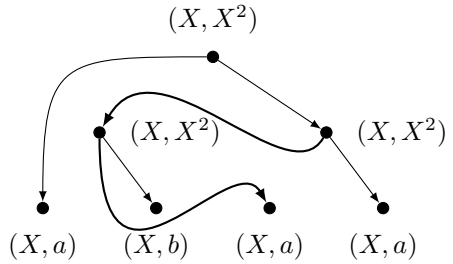
For this, we introduce the notion of *pump tree* ([6]): We obtain a pump tree from an X-tree by removing exactly one X-subtree. The idea is now to reduce the left subtree to a tree of dimension 0 by reallocating pump trees (encircled in the above figure) into the right subtree where we then deal recursively with the right subtree. We first remove such a pump tree from the rest of the tree by deleting the connecting edges and connecting the remaining parts as depicted here:

[5] To improve readability in the following illustrations, we replace the node labels $(X, 1)$, $(X, 2)$, $(X, 3)$ by (X, X^2), (X, a), (X, b), respectively.

[6] The term *pump tree* stems from the pumping lemma for context free languages.



Next, we reallocate the detached pump tree into the right subtree, e.g. as shown here:



It is easy to check that this new tree is indeed a derivation tree of $\boldsymbol{f}$ again and has the same yield as the original one. Further this tree is already compact. In general, we would have to proceed recursively in order to make the right subtree compact.

Note that, as we assume multiplication to be commutative, it is not important where we insert the pump tree into the right subtree. In the following proof we show that we can always find such pump trees and relocate them, i.e., find insertion points, if the tree under consideration is not compact. ◇

We now give a formal proof of Lemma 3.4.4:

Proof. First, let us introduce the following notation: We write $t = t_1 \cdot t_2$ to denote that t is combined from t_1 and t_2 in the following way: The tree t_1 is a "partial" derivation tree which is a regular derivation tree except for one leaf l missing its children. The tree t_2 is a regular derivation tree with $\lambda_v(t_2) = \lambda_v(l)$. The tree t is obtained from t_1 and t_2 by replacing the leaf l of t_1 by the tree t_2.

We proceed by induction on the number of nodes. In the base case, t has just one node, so $d(t) = 0$, and hence t is compact and we are done. In the following, let t have more than one node and assume $d(t) > L(t)$. We give a procedure that constructs a compact tree from t.

Let $s_1, s_2, \ldots, s_r$ be the children of t with $d(t) \geq d(s_1) \geq d(s_2) \geq \ldots \geq d(s_r)$. By induction we can make every child compact, i.e. $d(s_i) \leq L(s_i)$. We then have by definition of dimension

$$L(t) + 1 \leq d(t) \leq d(s_1) + 1 \leq L(s_1) + 1 \leq L(t) + 1.$$

Hence, we have $d(t) = d(s_1)+1$ which, by definition of dimension and compactness, implies $d(s_1) = d(s_2) = L(t) = L(s_1) = L(s_2)$. As $h(s_2) \geq d(s_2) = L(s_2)$ by Remark 3.2.10, we find a path in s_2 from the root to a leaf which passes through at least two nodes with

the same λ_v-label, say X_j. In other words, we may factor s_2 into $t_1^b \cdot (t_2^b \cdot t_3^b)$ such that $\lambda_v(t_2^b) = \lambda_v(t_3^b) = X_j$. As $L(t) = L(s_1) = L(s_2)$, we also find a node of s_1 labeled by X_j which allows us to write $s_1 = t_1^a \cdot t_2^a$ with $\lambda_v(t_2^a) = X_j$.

Now we move the middle part of s_2 to s_1, i.e., let $s_1' = t_1^a \cdot (t_2^b \cdot t_3^a)$ and let $s_2' = t_1^b \cdot t_3^b$. We then have $L(s_1') = L(s_1) = L(s_2) \geq L(s_2')$. By induction, s_1' and s_2' can be made compact, so $d(s_1') \leq d(s_1) = d(s_2) \geq d(s_2')$. Consider the tree t' obtained from t by replacing s_1 by s_1' and s_2 by s_2'. By commutativity, t and t' have the same yield. If $d(s_2') < d(s_2)$ then $d(t') \leq d(t) - 1 = L(t) = L(t')$ and we are done. Otherwise we iterate the described procedure.

This procedure terminates, because the number of nodes of (the current) s_2 strictly decreases in every iteration, and the number of nodes is an upper bound for $h(s_2)$ and, therefore, for $d(s_2)$. □

Now we can prove the main theorem of this section.

Theorem 3.4.6.
Let $\boldsymbol{f}$ be a vector of power series over a cio-semiring $\mathcal{S}$ in the variables $\mathcal{X}$ with $|\mathcal{X}| = n$. Then $\boldsymbol{\nu}^{(n)} = \mu\boldsymbol{f}$. ◇

Proof. We have for all $X \in \mathcal{X}$:

$$\begin{aligned}
(\mu\boldsymbol{f})_X &= \sum_{\text{trees } t \text{ with } \lambda_v(t)=X} Y(t) && \text{(Corollary 3.2.8)} \\
&= \sum_{\substack{\text{trees } t \text{ with } \lambda_v(t)=X \\ \text{and } d(t)\leq n}} Y(t) && \text{(Lemma 3.4.4)} \\
&= (\boldsymbol{\nu}^{(n)})_X && \text{(Theorem 3.2.11)}
\end{aligned}$$ □

Remark 3.4.7.
The bound of this theorem is tight as the following example shows: If $\boldsymbol{f}(X_1, \ldots, X_n) = (X_2^2 + a, X_3^2, \ldots, X_n^2, X_1^2)$, then $(\boldsymbol{\nu}^{(k)})_{X_1} = a$ for $k < n$, but $a^{2^n} \leq (\boldsymbol{\nu}^{(n)})_{X_1} = (\mu\boldsymbol{f})_{X_1}$. ◇

In terms of languages, combining Theorem 3.4.6 with Theorem 3.3.4 we obtain the following refined statement of Parikh's theorem:

Corollary 3.4.8.
Let $G = (\mathcal{X}, \Sigma, P, X_0)$ be a context-free grammar in CNF. Let $|\mathcal{X}| = n$. Then the commutative image of the index-$(n+1)$ approximation $L_{n+1}(G)$ equals the commutative image of $L(G)$. ◇

In Section 3.4.3 we discuss in more detail how our proof of Parikh's theorem is related to previous proofs of it.

3.4.2 Generalization to Commutative Kleene Algebras

In this subsection we generalize Theorem 3.4.6 to commutative Kleene algebras. A *commutative Kleene algebra* $\langle K, +, \cdot, ^*, 0, 1\rangle$ is commutative idempotent naturally ordered, but not necessarily ω-chain complete semiring $\langle K, +, \cdot, 0, 1\rangle$ where the *-operator is only required to satisfy these two axioms for all $a, b, c \in K$:

$$1 + aa^* \sqsubseteq a^* \quad \text{and} \quad a + bc \leq c \to b^*a \sqsubseteq c\ .$$

Notice that for a Kleene algebra there may not exist a notion of countable summation, as the *-operator is defined axiomatically. Thus, the axioms of commutative Kleene algebras are weaker than those of cio-semirings. In particular, the following example from [Koz90] shows there are commutative Kleene algebras which are not cio-semirings:

Example 3.4.9. Consider the Kleene algebra with carrier $\omega^2 := \mathbb{N}^2 \cup \{\bot, \top\}$, i.e., the set of ordered pairs of natural numbers extended by a bottom and a top element. We assume that ω^2 is totally ordered by $\prec$ with $\bot$ the minimum element, $\top$ the maximum element, and the lexicographic order on $\mathbb{N}^2$. Addition is defined to be the supremum of the elements w.r.t. $\prec$. Thus, $\prec$ becomes the natural order, and the additive neutral element is $\bot$. Note that this also gives us a notion of countable summation on ω^2. Multiplication is defined by

$$\begin{aligned} x \cdot \bot &= \bot \cdot x = \bot \\ x \cdot \top &= \top \cdot x = \top \quad (x \neq \bot) \\ (a, b) \cdot (c, d) &= (a + c, b + d) \end{aligned}$$

with neutral element $(0, 0)$. Finally, the Kleene-star is defined by

$$a^* = \begin{cases} \bot & \text{if } a = \bot \vee a = (0, 0) \\ \top & \text{else.} \end{cases}$$

This definition satisfies the axioms stated above. But obviously, we do not have a cio-semiring as

$$\sum_{i \in \mathbb{N}} (0, 1)^n = \sup\{(0, 1)^i | i \in \mathbb{N}\} = (1, 0) \prec \top = (0, 1)^*. \qquad \diamond$$

Thus, the fact that our result carries over to this more general setting is not obvious.

In the rest of the section we prove the following theorem which improves Hopkins and Kozen's result [HK99] from $\mathcal{O}(3^n)$ to n.

Theorem 3.4.10.
Let $\boldsymbol{f} \in \mathsf{RExp}^{\mathcal{X}}_{K \cup \mathcal{X}}$ be a vector of regular expressions ([7]) over a commutative Kleene algebra $\langle K, +, \cdot, ^*, 0, 1\rangle$. Let $|\mathcal{X}| = n$. Then $\boldsymbol{\nu}^{(n)} = \mu\boldsymbol{f}$. $\diamond$

[7] Recall that RExp_M denotes the set of regular expressions generated by the set M.

We have not yet defined $\boldsymbol{\nu}^{(i)}$ over a commutative Kleene algebra. We take the equations $\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\mathbf{0})$ and $\boldsymbol{\nu}^{(i+1)} = D\boldsymbol{f}|^*_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\nu}^{(i)})$ (cf. Proposition 3.3.1 (b)) as definition. For convenience, we define the *Hopkins-Kozen operator* $\mathsf{H}_{\boldsymbol{f}}$ by

$$\mathsf{H}_{\boldsymbol{f}}(\boldsymbol{X}) = D\boldsymbol{f}|^*_{\boldsymbol{X}}(\boldsymbol{X}) \ .$$

Then $\boldsymbol{\nu}^{(i)}$ is obtained by applying $\mathsf{H}_{\boldsymbol{f}}$ to $\boldsymbol{f}(\mathbf{0})$ i times:

$$\boldsymbol{\nu}^{(i)} = \mathsf{H}^i_{\boldsymbol{f}}(\boldsymbol{f}(\mathbf{0})) \ .$$

However, we still need to adapt some definitions for ω-continuous semirings to commutative Kleene algebras. In Kleene algebras, the Kleene star replaces ω-summation. So we modify the definition of differentials (see Remark 3.1.8) by replacing the equation for the $\sum$-operator by the definition of [HK99]:

$$\partial_X g^*|_{\boldsymbol{v}} = g^*(\boldsymbol{v}) \cdot \partial_X g|_{\boldsymbol{v}} \ . \tag{3.14}$$

(Note that this equation is satisfied by the Kleene star on cio-semirings.) Further, [HK99] gives, implicitly, a definition of $D\boldsymbol{f}|^*_{\boldsymbol{u}}(\boldsymbol{v})$ in commutative Kleene algebras, i.e., without expressing * using ω-summation. As we do not need the formal definition in the following, we restrict ourselves to this very brief description.

With those notations, and using the fact that [HK99] shows $\boldsymbol{\nu}^{(n)} \sqsubseteq \mu\boldsymbol{f}$, proving Theorem 3.4.10 amounts to showing the equation

$$\boldsymbol{f}(\mathsf{H}^n_{\boldsymbol{f}}(\boldsymbol{f}(\mathbf{0}))) = \mathsf{H}^n_{\boldsymbol{f}}(\boldsymbol{f}(\mathbf{0})) \ . \tag{3.15}$$

In order to prove (3.15) we appeal to Redko's theorem (see [Con71]). This theorem essentially states that an equation of terms over any commutative Kleene algebra holds if it holds under the *canonical commutative interpretation*. See Appendix A.3 for a technical justification of this fact. Let Σ be the finite set of elements of K appearing in $\boldsymbol{f}$. The canonical commutative interpretation $\mathsf{c}_\Sigma : \mathsf{RExp}_\Sigma \to 2^{\mathbb{N}^\Sigma}$ is defined by

$$\mathsf{c}_\Sigma(\alpha) = \{\#w \mid w \in L_\Sigma(\alpha)\} \ ,$$

where $\#w$ is the Parikh-vector of $w \in \Sigma^*$, i.e., $a \in \Sigma$ appears exactly $(\#w)_a$ times in w. We omit the subscript of c_Σ in the following. The cio-semiring of sets of Parikh-vectors $\mathcal{C}_\Sigma$ is defined by $\mathcal{C}_\Sigma = \langle 2^{\mathbb{N}^\Sigma}, \cup, \cdot, \emptyset, \{\mathbf{0}\}\rangle$ with $A \cdot B = \{a + b \mid a \in A, b \in B\}$ for all $A, B \subseteq \mathbb{N}^\Sigma$ and $\sum S = \bigcup S$ for all $S \subseteq 2^{\mathbb{N}^\Sigma}$. In particular we have $\mathsf{c}(\alpha^*) = \bigcup_{i\in\mathbb{N}} \mathsf{c}(\alpha)^i$. By Redko's theorem, we can prove (3.15) by showing $\mathsf{c}(\boldsymbol{f}(\mathsf{H}^n_{\boldsymbol{f}}(\boldsymbol{f}(\mathbf{0})))) = \mathsf{c}(\mathsf{H}^n_{\boldsymbol{f}}(\boldsymbol{f}(\mathbf{0})))$ over $\mathcal{C}_\Sigma$.

For $g \in \mathsf{RExp}_{\Sigma\cup\mathcal{X}}$ let g^{c} be its interpretation as map on $\mathcal{C}_\Sigma$, i.e.:

- If $\alpha \in \mathcal{X}$, then $\alpha^{\mathsf{c}} = \alpha$.
- If $\alpha \in \Sigma$, then $\alpha^{\mathsf{c}} = \mathsf{c}(\alpha)$.
- If $\alpha = \beta + \gamma$, then $\alpha^{\mathsf{c}} = \beta^{\mathsf{c}} \cup \gamma^{\mathsf{c}}$.
- If $\alpha = \beta \cdot \gamma$, then $\alpha^{\mathsf{c}} = \beta^{\mathsf{c}} \cdot \gamma^{\mathsf{c}}$.
- If $\alpha = \beta^*$, then $\alpha^{\mathsf{c}} = (\beta^{\mathsf{c}})^* = \bigcup_{i\in\mathbb{N}} (\beta^i)^{\mathsf{c}}$.

Define $\boldsymbol{f}^{\mathsf{c}}$ by applying above definition componentwise. Then $\boldsymbol{f}^{\mathsf{c}}$ is a vector of power series on $\mathcal{C}_\Sigma$ with $\mathsf{c}(\boldsymbol{f}(\boldsymbol{\alpha})) = \boldsymbol{f}^{\mathsf{c}}(\mathsf{c}(\boldsymbol{\alpha}))$ for all $\boldsymbol{\alpha} \in \mathsf{RExp}_\Sigma^{\mathcal{X}}$.

Assume $(\mathsf{H}_{\boldsymbol{f}})^{\mathsf{c}} = \mathsf{H}_{\boldsymbol{f}^{\mathsf{c}}}$. By Theorem 3.4.6, $\mathsf{H}^n_{\boldsymbol{f}^{\mathsf{c}}}(\boldsymbol{f}^{\mathsf{c}}(\emptyset))$ solves the equation system $\boldsymbol{X} = \boldsymbol{f}^{\mathsf{c}}(\boldsymbol{X})$ over $\mathcal{C}_\Sigma$. This implies:

$$\begin{aligned}\mathsf{c}(\boldsymbol{f}(\mathsf{H}^n_{\boldsymbol{f}}(\boldsymbol{f}(\boldsymbol{0})))) &= \boldsymbol{f}^{\mathsf{c}}((\mathsf{H}^n_{\boldsymbol{f}})^{\mathsf{c}}(\boldsymbol{f}^{\mathsf{c}}(\emptyset))) = \boldsymbol{f}^{\mathsf{c}}(\mathsf{H}^n_{\boldsymbol{f}^{\mathsf{c}}}(\boldsymbol{f}^{\mathsf{c}}(\emptyset))) = \mathsf{H}^n_{\boldsymbol{f}^{\mathsf{c}}}(\boldsymbol{f}^{\mathsf{c}}(\emptyset)) \\ &= \mathsf{c}(\mathsf{H}^n_{\boldsymbol{f}}(\boldsymbol{f}(\boldsymbol{0}))) \,.\end{aligned}$$

Then (3.15) follows by Redko's theorem.

So it remains to show that $(\mathsf{H}_{\boldsymbol{f}})^{\mathsf{c}} = \mathsf{H}_{\boldsymbol{f}^{\mathsf{c}}}$ indeed holds, which is equivalent to

$$\mathsf{c}(D\boldsymbol{f}|^*_{\boldsymbol{X}}(\boldsymbol{X})) = D\boldsymbol{f}^{\mathsf{c}}|^*_{\mathsf{c}(\boldsymbol{X})}(\mathsf{c}(\boldsymbol{X})) \,. \tag{3.16}$$

First we show the following lemma.

Lemma 3.4.11.
The following equation holds for all $\boldsymbol{u}, \boldsymbol{v} \in \mathsf{RExp}_\Sigma^{\mathcal{X}}$:

$$\mathsf{c}(D\boldsymbol{f}|_{\boldsymbol{u}}(\boldsymbol{v})) = D\boldsymbol{f}^{\mathsf{c}}|_{\mathsf{c}(\boldsymbol{u})}(\mathsf{c}(\boldsymbol{v})) \,. \qquad \diamond$$

Proof. One can prove this vector equation for each component separately, so we can assume $\boldsymbol{f} = f \in \mathsf{RExp}_{\Sigma\cup\mathcal{X}}$. Moreover, it suffices to show $\mathsf{c}(D_X f|_{\boldsymbol{u}}(\boldsymbol{v})) = D_X f^{\mathsf{c}}|_{\mathsf{c}(\boldsymbol{u})}(\mathsf{c}(\boldsymbol{v}))$ for all $X \in \mathcal{X}$. By Remark 3.1.8 this is equivalent to proving

$$\mathsf{c}\,(\partial_X f|_{\boldsymbol{u}}) = \partial_X f^{\mathsf{c}}\big|_{\mathsf{c}(\boldsymbol{u})} \,.$$

We proceed by induction on the structure of f. Only the case $f = g^*$ is interesting. We have:

$$\begin{aligned}\mathsf{c}\,(\partial_X g^*|_{\boldsymbol{u}}) &= \mathsf{c}\,(g^*(\boldsymbol{u}) \cdot \partial_X g|_{\boldsymbol{u}}) && \text{(Equation (3.14))} \\ &= \bigcup_{i\in\mathbb{N}} \mathsf{c}(g(\boldsymbol{u}))^i \cdot \mathsf{c}\,(\partial_X g|_{\boldsymbol{u}}) && \text{(definition of } \mathsf{c}) \\ &= \bigcup_{i\in\mathbb{N}} \mathsf{c}(g(\boldsymbol{u}))^i \cdot \partial_X g^{\mathsf{c}}\big|_{\mathsf{c}(\boldsymbol{u})} && \text{(induction)} \\ &= \bigcup_{i\geq 1} \big(g^{\mathsf{c}}(\mathsf{c}(\boldsymbol{u}))\big)^{i-1} \cdot \partial_X g^{\mathsf{c}}\big|_{\mathsf{c}(\boldsymbol{u})} && \text{(definition of } g^{\mathsf{c}})\end{aligned}$$

$$
\begin{aligned}
&= \bigcup_{i\in\mathbb{N}} \left(\partial_X (g^{\mathsf{c}})^i\big|_{\mathsf{c}(\boldsymbol{u})}\right) && \text{(idempotence of } \cup\text{,}\\
&&& \text{Remark 3.1.8: equation for } \cdot\ \text{)}\\
&= \partial_X \bigcup_{i\in\mathbb{N}} (g^{\mathsf{c}})^i \Big|_{\mathsf{c}(\boldsymbol{u})} && \text{(Remark 3.1.8: equation for } +\ \text{)}\\
&= \partial_X (g^*)^{\mathsf{c}}\big|_{\mathsf{c}(\boldsymbol{u})} && \text{(definition of } \mathsf{c}\text{)} \qquad \square
\end{aligned}
$$

As mentioned above, [HK99] implicitly defines $D\boldsymbol{f}|^*_{\boldsymbol{u}}(\boldsymbol{v})$ in commutative Kleene algebras. In particular, their definition satisfies

$$\mathsf{c}\left(D\boldsymbol{f}|^*_{\boldsymbol{u}}(\boldsymbol{v})\right) = \bigcup_{i\in\mathbb{N}} \mathsf{c}\left(D\boldsymbol{f}|^i_{\boldsymbol{u}}(\boldsymbol{v})\right) \ . \tag{3.17}$$

Now we can prove (3.16):

$$
\begin{aligned}
\mathsf{c}\left(D\boldsymbol{f}|^*_{\boldsymbol{X}}(\boldsymbol{X})\right) &= \bigcup_{i\in\mathbb{N}} \mathsf{c}\left(D\boldsymbol{f}|^i_{\boldsymbol{X}}(\boldsymbol{X})\right) && \text{(Equation (3.17))}\\
&= \bigcup_{i\in\mathbb{N}} D\boldsymbol{f}^{\mathsf{c}}\big|^i_{\mathsf{c}(\boldsymbol{X})}(\mathsf{c}(\boldsymbol{X})) && \text{(Lemma 3.4.11)}\\
&= D\boldsymbol{f}^{\mathsf{c}}\big|^*_{\mathsf{c}(\boldsymbol{X})}(\mathsf{c}(\boldsymbol{X})) && \text{(Lemma 3.1.13)}
\end{aligned}
$$

This concludes the proof of Theorem 3.4.10.

3.4.3 Comparison with previous proofs of Parikh's theorem

Neglecting the results regarding the convergence speed of the Newton's sequence, Theorem 3.4.6, resp. Theorem 3.4.10 give another proof of Parikh's result that the commutative image of a context-free language can be represented by a regular language. In this subsection we briefly sketch how our proof relate to the proofs given by Parikh [Par66], resp. Hopkins and Kozen [HK99], resp. Aceto, Ésik, and Ingólfsdóttir [AEI01].

For a context-free grammar G let $L(G)$ be the language generated by G, and define $L'(G)$ to be the subset of $L(G)$ such that for every $w \in L'(G)$ there is a derivation tree t such that every variable (non-terminal) of G appears at least once in t. Parikh reduces the problem of calculating the commutative image of $L(G)$ to that of $L'(G)$. He then uses the side condition imposed on $L'(G)$ to show that its commutative image is semi-linear set can be obtained

from the derivation trees, and the *partial* derivation trees (representing linear monomials) of height at most n^2 ([8]).

On the other hand, the proof by Hopkins and Kozen [HK99] relies completely on the axiomatic definitions of commutative Kleene algebras, and combines these with generalizations of results known from (vector) calculus. The idea of using concepts from calculus becomes obvious in the idea of using partial derivatives, but also the proof of Theorem 5.1 in [HK99] bears similarities to the proof of the implicit, resp. inverse function theorem.

Finally, Aceto, Ésik, and Ingólfsdóttir identify in [AEI01] a set of axioms describing the properties of the Kleene star ([9]) over some given semiring which allow to show Parikh's theorem. The interesting point here is that these axioms are purely equational, while the axioms used in [HK99] involve inequalities and implications (see the beginning of Subsection 3.4.2).

In comparison, our proof, combining both transformation of derivation trees and methods motivated from algebra and calculus, can be filed between the original proof by Parikh, and the one by Hopkins and Kozen. In particular, Parikh's side condition that all variables should appear in a derivation tree is similar to our notion of compact tree introduced in Definition 3.4.3. Further, the partial derivation trees used by Parikh are in our case obtained via the differential of $\boldsymbol{f}$. On the other hand, we borrow from calculus Newton's method and the concepts of iterative linearization, similar to [HK99].

Compared to [AEI01], it is still an open question if there is a set of purely equational axioms which allow to proof that $\boldsymbol{\nu}^{(n)} = \mu\boldsymbol{f}$.

3.5 Non-Distributive Program Analyses

We have seen in the introduction that system of polynomials naturally arise in the setting of program analysis. Up to now we have only considered the case where the maps induced by these systems operate on semirings, thus assuming that multiplication distributes over addition. As several interesting analyses of programs are only subdistributive, i.e., only satisfy

$$a \cdot (b + c) \sqsupseteq a \cdot b + a \cdot c \text{ and } (b + c) \cdot a \sqsupseteq b \cdot a + c \cdot a,$$

[8] Here, with partial derivation tree we denote every tree we obtain from a derivation tree by removing exactly one leaf, i.e., a partial derivation tree yields a linear monomial.

[9] More precisely, in [AEI01] least fixed-point expressions (μ-terms), like $\mu z.xz + y$, are considered, generalizing the Kleene star.

we discuss in this section how the Newton sequence generalizes to the algebraic structure we obtain from semirings by replacing the axiom of distributivity by above axiom of subdistributivity. We first give a brief summary of the role of distributivity in program analysis:

For programs without procedures, distributive program analyses were first considered in [Kil73]. Recall from the introduction that, given a program and the distributive transfer functions of a program analysis, one can construct a vector $\boldsymbol{f}$ of polynomials such that, for every program point p, the p-component of the least fixed point $\mu\boldsymbol{f}$ coincides with the so-called *meet-on-all-paths value* or short *MOP-value* ([10]) of p, the sum of the dataflow values of all program paths leading to p.

The framework of [Kil73] was generalized to non-distributive transfer functions in [KU77]. Non-distributivity means, in our terms, that only *subdistributivity* holds ([11]). There are interesting program analyses, such as constant propagation, which are non-distributive, see e.g. [KU77, NNH99]. In those cases, the least fixed point does not necessarily coincide with the MOP-value, but safely overapproximates it.

For procedural programs, [SP81] considered only distributive analyses. [KS92] refined this approach for programs with local variables, and also showed that, like in the case without procedures, the least fixed point is an overapproximation of the MOP-value.

To be precise we define the MOP-value as the vector $\boldsymbol{M}$ with $\boldsymbol{M}_p = \mathsf{Y}(\mathcal{T}_p)$ where $\mathcal{T}_p$ is the set of trees labeled with p. Notice that a depth-first traversal of a tree labeled with p precisely corresponds to an interprocedural path from the beginning of the procedure of p to the program point p, i.e., the MOP-value $\boldsymbol{M}_p = \mathsf{Y}(\mathcal{T}_p)$ is in fact the sum of the dataflow values of all paths to p.

Corollary 3.2.8 states for the distributive case that $\boldsymbol{M} = \mu\boldsymbol{f}$, and we have seen in Theorem 2.2.12 and Theorem 3.1.10 that the Kleene and Newton sequences converge to this value. For the non-distributive case, the least fixed point overapproximates the MOP-value, i.e., $\boldsymbol{M} \sqsubseteq \mu\boldsymbol{f}$, cf. [KS92]. In the following we try to sketch that in a natural extension from semirings to "subdistributive semirings", Newton's method is defined as well. Moreover, the

[10] We keep the term MOP-value for historical reasons.

[11] If addition is idempotent (as for lattice joins) this condition is equivalent to the monotonicity of multiplication, or, in traditional terms, to the monotonicity of the transfer functions [KU77]. The stricter distributivity condition, on the other hand, amounts to requiring the transfer functions to be homomorphisms.

Kleene and Newton sequences both converge to overapproximations of $\boldsymbol{M}$, more precisely, we show $\boldsymbol{M} \sqsubseteq \bigsqcup_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)} \sqsubseteq \bigsqcup_{i\in\mathbb{N}} \boldsymbol{\nu}^{(i)}$.

For this we first define *subdistributive (ω-chain complete) semirings* ([12]):

Definition 3.5.1.
A subdistributive semiring is given by a tuple $\langle S, +, \cdot, 0, 1\rangle$ where the following properties are required to hold:

(1) $\langle S, +, 0\rangle$ is a commutative monoid.

(2) $\langle S, \cdot, 1\rangle$ is a monoid.

(3) $0 \cdot a = a \cdot 0 = 0$ for all $a \in S$.

(4) $a \cdot (b + c) \sqsupseteq a \cdot b + a \cdot c$ and $(a + b) \cdot c \sqsupseteq a \cdot c + b \cdot c$ for all $a, b, c \in S$.

(5) The relation $\sqsubseteq := \{(a, b) \in S \times S \mid \exists d \in S : a + d = b\}$ is a partial order.

(6) For all *ω-chains* $(a_i)_{i\in\mathbb{N}}$ $\bigsqcup_{i\in\mathbb{N}} a_i$ exists.

For *any* sequence $(b_i)_{i\in\mathbb{N}}$ define $\sum_{i\in\mathbb{N}} b_i := \bigsqcup\{a_0 + a_1 + \ldots + a_i \mid i \in \mathbb{N}\}$. ◇

Remark 3.5.2.
We obtain the definition of subdistributive semiring from the definition of ω-continuous semiring by removing (7), and replacing distributivity with subdistributivity (see (4)). ◇

In the remainder of this section $\langle S, +, \cdot, 0, 1\rangle$ always denotes a subdistributive semiring. Polynomials, vectors, differential, etc. are then defined analogously as in the distributive setting.

Note that we still have in any subdistributive semiring that the following inequalities hold for all sequences $(a_i)_{i\in\mathbb{N}}$, $c \in S$, and partitions $(I_j)_{j\in J}$ of $\mathbb{N}$:

$$c\cdot\left(\sum_{i\in\mathbb{N}} a_i\right) \sqsupseteq \sum_{i\in\mathbb{N}}(c\cdot a_i), \quad \left(\sum_{i\in\mathbb{N}} a_i\right)\cdot c \sqsupseteq \sum_{i\in\mathbb{N}}(a_i\cdot c), \quad \sum_{j\in J}\left(\sum_{i\in I_j} a_j\right) \sqsupseteq \sum_{i\in\mathbb{N}} a_i\,.$$

Thus, any polynomial p is still monotone, but not necessarily ω-continuous. Still, for any sequence $(\boldsymbol{v}_i)_{i\in\mathbb{N}}$ (of vectors) on a subdistributive semiring we have $p(\sum_{i\in\mathbb{N}} \boldsymbol{v}_i) \sqsupseteq \sum_{i\in\mathbb{N}} p(\boldsymbol{v}_i)$. Hence, the Kleene sequence of a polynomial system $\boldsymbol{f}$ still converges, but not necessarily to the least fixed point of $\boldsymbol{f}$:

[12] We drop *ω-chain-complete* in the following.

Corollary 3.5.3.
For any system $\boldsymbol{f}$ of polynomials we have that the Kleene sequence $(\boldsymbol{\kappa}^{(i)})_{i\in\mathbb{N}}$ is an ω-chain. If $\boldsymbol{f}$ has a least solution $\mu\boldsymbol{f}$, then $\bigsqcup_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)} \sqsubseteq \mu\boldsymbol{f}$. ◇

Still, the limit of the Kleene sequence exists as it is an ω-chain, and this limit is a safe approximation of the MOP-value:

Proposition 3.5.4.
For any polynomial system $\boldsymbol{f}$ we have $\left(\boldsymbol{\kappa}^{(i)}\right)_X \sqsupseteq \mathsf{Y}(\mathcal{H}^i_X)$, and, hence, $\left(\bigsqcup_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)}\right)_X \sqsupseteq \mathsf{Y}(\mathcal{T}_X)$ where $\mathcal{T}_X$ is the set of trees labeled with X. ◇

We skip the proof of this proposition as it is almost identical to the one of Proposition 3.2.5. The only difference is that when expanding the components of $\boldsymbol{\kappa}^{(i)}$ into a sum of products of coefficients the subdistributivity only guarantees that $\boldsymbol{\kappa}^{(i)}$ is an upper bound. Similarly, subdistributivity only allows us to generalize the lower bound from Lemma 3.1.14, i.e., we have

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

for a polynomial system $\boldsymbol{f}$ and vectors $\boldsymbol{u}, \boldsymbol{v}$.

We now turn to the definition of *Newton sequence*.

Definition 3.5.5.
For $\boldsymbol{f}$ a polynomial system in the variables $\boldsymbol{X}$, and $\boldsymbol{a}, \boldsymbol{b}$ vectors we set

$$L_{\boldsymbol{f};\boldsymbol{a};\boldsymbol{b}}(\boldsymbol{X}) := \boldsymbol{b} + D\boldsymbol{f}|_{\boldsymbol{a}}(\boldsymbol{X}).$$ ◇

Definition 3.5.6.
Let $\boldsymbol{f}$ be a polynomial system. For $i \in \mathbb{N}$, an i-th *Newton approximant* $\boldsymbol{\nu}^{(i)}$ is inductively defined by

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\vec{0}) \quad \text{and} \quad \boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)} ,$$

where $\boldsymbol{\Delta}^{(i)}$ has to satisfy $\sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}) \sqsubseteq \boldsymbol{\Delta}^{(i)} \sqsubseteq L_{\boldsymbol{f};\boldsymbol{\nu}^{(i)};\boldsymbol{\delta}^{(i)}}\left(\boldsymbol{\Delta}^{(i)}\right)$. Any such sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ of Newton approximants is called *Newton sequence*. ◇

Remark 3.5.7.
If $\boldsymbol{\delta}^{(i)}$ exists, then possible choices for $\boldsymbol{\Delta}^{(i)}$ are

$$\sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}),\ \bigsqcup_{k\in\mathbb{N}} L^k_{\boldsymbol{f};\boldsymbol{\nu}^{(i)};\boldsymbol{\delta}^{(i)}}(\boldsymbol{0}) \text{ or (if it exists) } \mu L_{\boldsymbol{f};\boldsymbol{\nu}^{(i)};\boldsymbol{\delta}^{(i)}}.$$

Note that in the distributive setting all three values coincide. ◇

Proposition 3.5.8.
Let $\boldsymbol{f}\colon V \to V$ be a vector of power series.

- For every Newton approximant $\boldsymbol{\nu}^{(i)}$ there exists a vector $\boldsymbol{\delta}^{(i)}$ such that $\boldsymbol{f}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)}$. So there is at least one Newton sequence.
- Every Newton sequence $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ satisfies $\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) \sqsubseteq \boldsymbol{\nu}^{(i+1)}$ for all $i \in \mathbb{N}$. ◇

Proof. First we prove for all $i \in \mathbb{N}$ that a suitable $\boldsymbol{\delta}^{(i)}$ exists and, at the same time, that the inequality $\boldsymbol{\kappa}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)})$ holds. We proceed by induction on i. For the base case $i = 0$ we have:

$$\boldsymbol{\nu}^{(0)} = \boldsymbol{f}(\mathbf{0}) = \boldsymbol{\kappa}^{(0)} \sqsubseteq \boldsymbol{\kappa}^{(1)} = \boldsymbol{f}(\boldsymbol{\kappa}^{(0)}) = \boldsymbol{f}(\boldsymbol{\nu}^{(0)}).$$

So, there exists a $\boldsymbol{\delta}^{(0)}$ with $\boldsymbol{\nu}^{(0)} + \boldsymbol{\delta}^{(0)} = \boldsymbol{f}(\boldsymbol{\nu}^{(0)})$, and hence we have:

$$\boldsymbol{\nu}^{(1)} = \boldsymbol{\nu}^{(0)} + \boldsymbol{\Delta}^{(0)} \sqsupseteq \boldsymbol{\nu}^{(0)} + \sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(0)}}(\boldsymbol{\delta}^{(0)}) \sqsupseteq \boldsymbol{\nu}^{(0)} + \boldsymbol{\delta}^{(0)} = \boldsymbol{f}(\boldsymbol{\nu}^{(0)}).$$

For the induction step, let $i \geq 0$.

$$\boldsymbol{\kappa}^{(i+1)} = \boldsymbol{f}(\boldsymbol{\kappa}^{(i)}) \sqsubseteq \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) = \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)} \sqsubseteq \boldsymbol{\nu}^{(i)} + \sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}).$$

As we require that $\sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)}) \sqsubseteq \boldsymbol{\Delta}^{(i)}$, it now immediately follows that

$$\boldsymbol{\kappa}^{(i+1)} \sqsubseteq \boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)} = \boldsymbol{\nu}^{(i+1)}.$$

By definition of $\boldsymbol{\Delta}^{(i)}$ we have $\boldsymbol{\Delta}^{(i)} \sqsubseteq L_{\boldsymbol{f};\boldsymbol{\nu}^{(i)};\boldsymbol{\delta}^{(i)}}(\boldsymbol{\Delta}^{(i)}))$; it therefore follows:

$$\begin{aligned}\boldsymbol{\nu}^{(i+1)} = \boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)} &\sqsubseteq \boldsymbol{\nu}^{(i)} + \boldsymbol{\delta}^{(i)} + D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}\big(\boldsymbol{\Delta}^{(i)}\big) \\ &= \boldsymbol{f}(\boldsymbol{\nu}^{(i)}) + D\boldsymbol{f}|_{\boldsymbol{\nu}^{(i)}}\big(\boldsymbol{\Delta}^{(i)}\big) \sqsubseteq \boldsymbol{f}\big(\boldsymbol{\nu}^{(i)} + \boldsymbol{\Delta}^{(i)}\big) = \boldsymbol{f}(\boldsymbol{\nu}^{(i+1)}).\end{aligned}$$

We complete our proof by

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{\nu}^{(i+1)}) = \boldsymbol{\nu}^{(i+1)} + \boldsymbol{\delta}^{(i+1)} &\sqsubseteq \boldsymbol{\nu}^{(i+1)} + \sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(i+1)}}(\boldsymbol{\delta}^{(i+1)}) \\ &\sqsubseteq \boldsymbol{\nu}^{(i+1)} + \boldsymbol{\Delta}^{(i+1)} = \boldsymbol{\nu}^{(i+2)}.\end{aligned}$$

□

Proposition 3.5.9.
Let $\boldsymbol{M}$ be the MOP-value, i.e., the vector $\boldsymbol{M}$ with $\boldsymbol{M}_X = \mathsf{Y}(\mathcal{T}_X)$. Then $\boldsymbol{M} \sqsubseteq \bigsqcup_{i\in\mathbb{N}} \boldsymbol{\kappa}^{(i)} \sqsubseteq \bigsqcup_{i\in\mathbb{N}} \boldsymbol{\nu}^{(i)}$. ◇

Proof. Follows directly from Propositions 3.5.4 and 3.5.8. □

Proposition 3.5.10.
For $\boldsymbol{\Delta}^{(i)} = \sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{\nu}^{(i)}}(\boldsymbol{\delta}^{(i)})$ we have $\bigsqcup_{i\in\mathbb{N}} \boldsymbol{\nu}^{(i)} \sqsubseteq \mu\boldsymbol{f}$, if $\mu\boldsymbol{f}$ exists. ◇

Proof. The proof is almost identical to the one of Proposition 3.1.11. Note that the proof of Lemma 3.1.15 does not use distributivity. □

Theorem 3.5.11 (Tree Characterization of the Newton Sequence)**.**
Let $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ be a Newton sequence of $\boldsymbol{f}$. For every $X \in \mathcal{X}$ and every $i \geq 0$ we have $\left(\boldsymbol{\nu}^{(i)}\right)_X \sqsupseteq \mathsf{Y}(\mathcal{D}^i_X)$, i.e., the X-component of the i-th Newton approximant is a safe approximation of the yield of $\mathcal{D}^i_X$. $\diamond$

Proof. In the distributive setting we proved this theorem via induction where we expanded the the terms we obtained using distributivity. In the subdistributive case the same proof still guarantees that $\left(\boldsymbol{\nu}^{(i)}\right)_X \sqsupseteq \mathsf{Y}(\mathcal{D}^i_X)$. $\square$

Chapter 4

Derivation Tree Analysis

4.1 Introduction

In the last chapter we showed that on cio-semirings the Newton sequence reaches the least fixed point after a linear number of steps (cf. Theorem 3.4.6). The principle underlying the proof of this result can be paraphrased as follows: We first analyzed the structure of the derivation trees corresponding to the Newton approximations $\boldsymbol{\nu}^{(i)}$ (see Proposition 3.3.1) which led us to the notion of tree dimension, and the result that $\boldsymbol{\nu}^{(i)}$ is given by the yield of the derivation trees $\mathcal{D}^i$ of dimension at most i (see Theorem 3.2.11). By definition, the set $\mathcal{D}^i$ was a subset of the derivations trees associated with $\boldsymbol{f}$, i.e., $\mathcal{D}^i \subseteq \mathcal{T}$. To obtain Theorem 3.4.6 it then remained to show that we have $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{D}^n_X)$ for any tree $t \in \mathcal{T}_X$, as by idempotence and ω-continuity of summation([1]) we then could conclude that

$$\sum_{t \in \mathcal{T}_X} \mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{D}^n_X) = \left(\boldsymbol{\nu}^{(n)}\right)_X.$$

The proof of Theorem 3.4.6 can thus be broken down into these two steps:

(1) Identify the derivation trees $\mathcal{D}^i$ of $\boldsymbol{f}$ corresponding to $\boldsymbol{\nu}^{(i)}$.

(2) Show that all derivation trees $\mathcal{T}$ of $\boldsymbol{f}$ can be *embedded* into $\mathcal{D}^n$.

We call this proof principle *derivation tree analysis*.

[1] More precisely, for any sequence $(a_i)_{i\in\mathbb{N}}$ with $a_i \sqsubseteq d$ we have by idempotence $a_0 + a_1 + \ldots + a_i \sqsubseteq d + d + \ldots d = d$ for all $i \in \mathbb{N}$. Now, as $\sum_{i\in\mathbb{N}} a_i = \bigsqcup\{a_0 + a_1 + \ldots + a_i | i \in \mathbb{N}\}$ it immediately follows that $\sum_{i\in\mathbb{N}} a_i \sqsubseteq d$.

Definition 4.1.1.
Fix a polynomial system $\boldsymbol{f}$ in the variables $\mathcal{X}$ (with $n := |\mathcal{X}|$) over some idempotent ω-continuous semiring. We say that a set T_X of X-trees satisfies the *embedding property* if $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(T_X)$ holds for every X-tree t. ◇

Proposition 4.1.2.
Let $\boldsymbol{f}$ be a system of polynomials over an io-semiring, and let X be a variable of $\boldsymbol{f}$. If a set T_X of X-trees of $\boldsymbol{f}$ satisfies the embedding property, then $(\mu\boldsymbol{f})_X = \mathsf{Y}(T_X)$. ◇

Derivation tree analysis is used in this chapter for showing that in the following special cases of io-semirings the least fixed point of a polynomial system $\boldsymbol{f}$ can be calculated more efficiently than by means of Newton's method or the Kleene sequence where the latter is not guaranteed to reach $\mu\boldsymbol{f}$ after a finite number of steps. In particular, we consider the following classes of idempotent ω-continuous semirings:

Definition 4.1.3.
Let $\mathcal{S} = \langle S, +, \cdot, 0, 1\rangle$ be an idempotent ω-continuous semiring.

(1) $\mathcal{S}$ is a *star-distributive semiring* if multiplication is commutative and the Kleene-star distributes over finite sums, i.e.,

$$x \cdot y = y \cdot x \text{ and } (x + y)^* = x^* + y^*$$

holds for all $x, y \in S$.

(2) $\mathcal{S}$ is a *lossy semiring*, if

$$1 \sqsubseteq x$$

holds for all $x \in S \setminus \{0\}$.

(3) $\mathcal{S}$ is a 1-*bounded semiring*, if

$$x \sqsubseteq 1$$

holds for all $x \in S$. ◇

In the case of star-distributive, resp. lossy semirings we show that for any polynomial system $\boldsymbol{f}$ in n variables $\boldsymbol{X}$ its linearization

$$\boldsymbol{f}_{\mathcal{B}}(\boldsymbol{X}) := \boldsymbol{f}^n(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}(\boldsymbol{X})$$

inherits the least fixed point from $\boldsymbol{f}$, i.e.,

$$\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}} = D\boldsymbol{f}|^*_{\boldsymbol{f}^n(\boldsymbol{0})}(\boldsymbol{f}^n(\boldsymbol{0})).$$

In the case of 1-bounded semirings we can show that $\mu\boldsymbol{f}$ is always obtained after at most n Kleene steps, i.e., $\mu\boldsymbol{f} = \boldsymbol{f}^n(\boldsymbol{0})$.

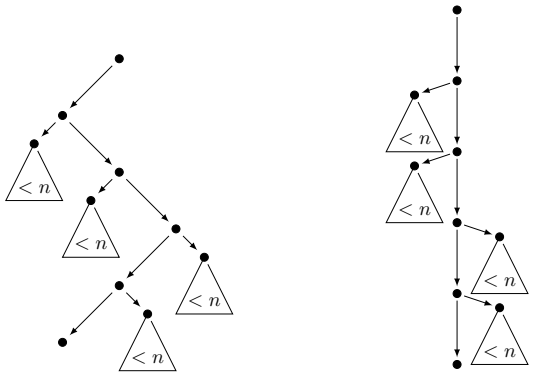


Figure 4.1: An example of the structure of a bamboo: it consists of a stem of unbounded length from which subtrees of height less than n sprout; on the right it is shown with its stem straightened.

4.2 Bamboos and their Yield

The difficulty of derivation tree analysis lies in finding a set T_X exhibiting a good balance between the contradictory requirements "easy to compute" and "relevant": if $T_X = \emptyset$ then the yield is trivial to compute, but T_X does not satisfy the embedding property in any interesting case. Conversely, $T_X = \mathcal{T}_X$ trivially satisfies the embedding property for every io-semiring, but its yield is not easy to compute. In the case of 1-bounded semirings we will show that one can take for T_X simply the set of X-trees of height at most n. For star-distributive, resp. lossy semirings surprisingly the same set T_X of trees, called *bamboos* (see below), can be taken. In this section we define bamboos and show that their yield is the least solution of a system of *linear* equations easily derivable from $\boldsymbol{f}$. The "easy to compute" part is justified by the fact that in most semirings used in practice linear equations are far easier to solve than polynomial equations (e.g. in the real semiring or the language semiring with union and concatenation as operations). The "relevance" of bamboos is justified in the next three sections.

Definition 4.2.1.
Let $\boldsymbol{f}$ be a system of polynomials. A tree $t \in \mathcal{T}_{\boldsymbol{f},X}$ is an X-*bamboo* if there is a path leading from the root to some leaf of t, the *stem*, such that the height of every subtree of t not containing a node of the stem is at most $n-1$. The set of all X-bamboos of $\boldsymbol{f}$ is denoted by $\mathcal{B}_{f,X}$, or just by $\mathcal{B}_X$ if $\boldsymbol{f}$ is clear from the context. ◇

Figure 4.1 depicts the basic structure of a bamboo.

Definition 4.2.2.
Let $\boldsymbol{f}$ be a system of n polynomials. The *bamboo system* $\boldsymbol{f}_{\mathcal{B}}$ associated to $\boldsymbol{f}$ is the linear system $\boldsymbol{f}_{\mathcal{B}}(\boldsymbol{X}) = D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}(\boldsymbol{X}) + \boldsymbol{f}(\mathbf{0})$. The least solution of the system of equations $\boldsymbol{X} = \boldsymbol{f}_{\mathcal{B}}(\boldsymbol{X})$ is denoted by $\mu\boldsymbol{f}_{\mathcal{B}}$. ◇

Now we can state the relation between bamboos and bamboo systems.

Theorem 4.2.3.
Let $\boldsymbol{f}$ be a system of polynomials over an io-semiring. For every variable X of $\boldsymbol{f}$ we have $\mathsf{Y}(\mathcal{B}_X) = (\mu\boldsymbol{f}_{\mathcal{B}})_X$, i.e., the yield of the X-bamboos is equal to the X-component of the least solution of the bamboo system. ◇

Proof sketch. The proof idea is similar to the one sketched in Example 3.2.16: Every tree t associated with the bamboo system $\boldsymbol{f}_{\mathcal{B}}$ is a chain where every inner node of t is labeled by a monomial of $D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}(\boldsymbol{X})$, and its leaf is labeled by a component of $\boldsymbol{f}^n(\mathbf{0})$. As $\boldsymbol{f}^n(\mathbf{0})$ corresponds to the trees $\mathcal{H}_X^{n-1}$ w.r.t. $\boldsymbol{f}$ of height less than n, we may identify t with the set of trees we obtain from t by "replacing" the semiring element $\boldsymbol{f}^n(\mathbf{0})_X$ by the trees of $\mathcal{H}_X^{n-1}$. It is not hard to show that every such tree is indeed a tree w.r.t. $\boldsymbol{f}$. □

Together with Proposition 4.1.2 we get the following corollary.

Corollary 4.2.4 (derivation tree analysis for bamboos)**.**
Let $\boldsymbol{f}$ be a system of polynomials over an io-semiring. If $\mathcal{B}_X$ satisfies the embedding property for all X, i.e., for all X-trees t it holds $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$, then $\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$. ◇

4.3 Star-Distributive Semirings

We first recall the definition of star-distributive semiring:

Definition 4.3.1.
A cio-semiring $\mathcal{S}$ is *star-distributive* if for all $a, b \in S$ we have

$$(a + b)^* = a^* + b^*.$$ ◇

Proposition 4.3.2.
Any totally ordered cio-semiring is star-distributive. ◇

Proof. Let w.l.o.g. $a \sqsubseteq b$. Then $(a + b)^* = b^* \sqsubseteq a^* + b^* \sqsubseteq (a + b)^*$. □

In particular, the $(\min, +)$-semiring is star-distributive.

We have already considered cio-semirings in Section 3.4, where we showed that $\mu\boldsymbol{f}$ can be computed by solving n *linear* equation systems by means of a Newton-like method, improving the $\mathcal{O}(3^n)$ bound of Hopkins and Kozen [HK99]. In this section we improve this result even further for star-distributive semirings: One single linear system, the bamboo system $\boldsymbol{f}_{\mathcal{B}}$, needs to be solved. This leads to an efficient algorithm for computing $\mu\boldsymbol{f}$ in arbitrary star-distributive semirings. In Section 4.3.1 we instantiate this algorithm for the $(\min, +)$-semiring; in Section 4.3.2 we use it to improve the algorithm of [CCFR07] for computing the throughput of a context-free grammar.

We start by stating two useful properties of star-distributive semirings.

Proposition 4.3.3.
In any star-distributive semiring the following equations hold:
(1) $a^*b^* = a^* + b^*$, and (2) $(ab^*)^* = a^* + ab^*$. ◇

We can now state and prove our result:

Theorem 4.3.4.
For any polynomial system $\boldsymbol{f}$ over a star-distributive semiring $\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$ holds. ◇

Proof sketch (see the appendix for a complete proof). The proof is by derivation tree analysis. So it suffices to discharge the precondition of Corollary 4.2.4. More precisely we show for any X-tree t that $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$ holds. It suffices to consider the case where t is not an X-bamboo. Then the height of t is at least n, and so t is "pumpable", i.e., one can choose a path p in t from the root to a leaf such that two different nodes on the path share the same variable-label. So t can be decomposed into three (partial) trees with yields a, b, c, respectively, such that $\mathsf{Y}(t) = abc$, see Figure 4.2(a). Notice that, by commutativity of product, ab^*c is the yield of a set of trees obtained by "pumping" t. We show $ab^*c \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$ which implies $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$. As t is not an X-bamboo, t has a pumpable subtree disjoint from p. In this sketch we assume that it is a subtree of that part of t whose yield is a, see Figure 4.2(b). Now we have $a = a_1a_2a_3$, and so $ab^*c = a_1a_2a_3b^*c \sqsubseteq a_1a_2^*a_3b^*c = a_1a_3b^*c + a_1a_2^*a_3c$, where we used commutativity and Proposition 4.3.3(1) in the last step. Both summands in above sum are yields of sets of trees obtained by pumping pumpable trees smaller than t, see Figure 4.2(c) and (d). An inductive argument then shows that both $a_1a_3b^*c$ and $a_1a_2^*a_3c$ are less than $\mathsf{Y}(\mathcal{B}_X)$. As addition is idempotent, we therefore also obtain $a_1a_3b^*c + a_1a_2^*a_3c \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$. □

We have sketched in Example 2.2.16 how the Kleene star of a matrix on an io-semiring can be calculated. As star-distributive semirings are commutative, the bamboo system $\boldsymbol{f}_{\mathcal{B}}$ can be represented by means of a matrix, and

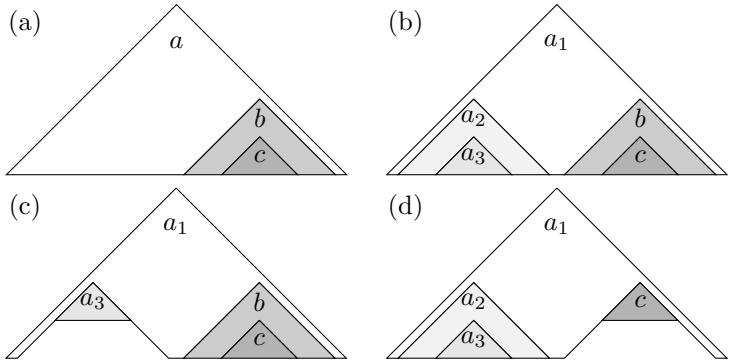


Figure 4.2: "Unpumping" trees to make them bamboos

so its least fixed point $\boldsymbol{f}_{\mathcal{B}}$ can be represented by means of a regular expression interpreted on the underlying star-distributive semiring. The following subsection shows that on $\mathcal{S}_{\min}$ we also may use the Bellman-Ford algorithm to obtain $\mu\boldsymbol{f}_{\mathcal{B}}$ directly.

4.3.1 The $(\min, +)$-Semiring

Consider the $(\min, +)$-semiring $\mathcal{S}_{\min} = \langle \mathbb{R} \cup \{-\infty, \infty\}, \sqcap, +, \infty, 0\rangle$ with $a \sqcap b := \min\{a, b\}$. Then the natural order $\sqsubseteq$ is the order $\geq$ on the reals extended by ∞, resp. $-\infty$ as top, resp. bottom element.([2]) As $\mathcal{S}_{\min}$ is totally ordered, Proposition 4.3.2 implies that $\mathcal{S}_{\min}$ is star-distributive. Assume for the rest of this section that $\boldsymbol{f}$ is a polynomial system on $\mathcal{S}_{\min}$.We can apply Theorem 4.3.4, i.e., $\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$ holds. This immediately suggests a polynomial algorithm to compute the least fixed-point: Compute $\boldsymbol{f}^n(\boldsymbol{\infty})$ by performing n Kleene iterations, and solve the linear system $\boldsymbol{X} = D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{\infty})}(\boldsymbol{X}) \sqcap \boldsymbol{f}(\boldsymbol{\infty})$. The latter can be done by means of the Bellman-Ford algorithm.

Example 4.3.5. Consider the following equation system.

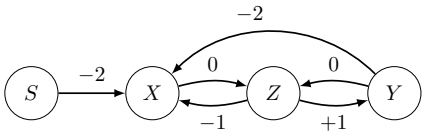
$$\big(X, \quad Y, \quad Z\big) = \big(-2 \sqcap (Y + Z), \quad Z + 1, \quad X \sqcap Y\big) =: \boldsymbol{f}(\boldsymbol{X})$$

We have $\boldsymbol{f}(\boldsymbol{\infty}) = (-2, \infty, \infty)$, $\boldsymbol{f}^2(\boldsymbol{\infty}) = (-2, \infty, -2)$, $\boldsymbol{f}^3(\boldsymbol{\infty}) = (-2, -1, -2)$. The linear system $\boldsymbol{X} = D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{\infty})}(\boldsymbol{X}) \sqcap \boldsymbol{f}(\boldsymbol{\infty}) = \boldsymbol{f}_{\mathcal{B}}(\boldsymbol{X})$ looks as follows:

$$\big(X, \quad Y, \quad Z\big) = \big(-2 \sqcap (-1 + Z) \sqcap (Y + -2), \quad Z + 1, \quad X \sqcap Y\big).$$

This equation system corresponds in a straightforward way to the following graph.

[2] By symmetry, we could equivalently consider maximum instead of minimum.



We claim that the V-component of $\mu \boldsymbol{f}_{\mathcal{B}}$ is equal to the least weight of any path from S to V where $V \in \{X, Y, Z\}$. To see this, notice that $(\boldsymbol{f}^k_{\mathcal{B}}(\boldsymbol{\infty}))_V$ corresponds to the least weight of any path from S to V of length at most k. The claim then follows by Kleene's fixed-point theorem. So we can compute $\mu \boldsymbol{f}_{\mathcal{B}}$ with the Bellman-Ford algorithm. In our example, X, Y, Z are all reachable from S via a negative cycle, so $\mu \boldsymbol{f}_{\mathcal{B}} = (-\infty, -\infty, -\infty)$. By Theorem 4.3.4, $\mu \boldsymbol{f} = \mu \boldsymbol{f}_{\mathcal{B}} = (-\infty, -\infty, -\infty)$. ◇

The Bellman-Ford algorithm can be used here as it handles negative cycles correctly. The overall runtime of our algorithm to compute $\mu \boldsymbol{f}$ is dominated by the Bellman-Ford algorithm. Its runtime is in $\mathcal{O}(n \cdot m)$, where m is the number of monomials appearing in $\boldsymbol{f}$. We conclude that our algorithm has the same asymptotic complexity as the "generalized Bellman-Ford" algorithm of [GS07]. Note that applying Newton's method to this problem would result in solving n linear systems (cf. Theorem 3.4.6) instead of only one.

In Chapter 5 we explore totally ordered star-distributive semirings, like the $(\min, +)$-semiring considered here, in more detail. On idempotent totally ordered semirings addition becomes the maximum w.r.t. the natural order, and we may consider *min-max-systems*, i.e., polynomial systems which do not only use maximum (addition) and multiplication, but also minimum. We study the existence and calculation of the least fixed point of such min-max-systems in Chapter 5.

4.3.2 Throughput of Grammars

In [CCFR07], a polynomial algorithm for computing the *throughput* of a context-free grammar was given. Now we show that the algorithm can be both simplified and accelerated by computing least fixed-points according to Theorem 4.3.4.

Let us define the problem following [CCFR07]. Let Σ be a finite alphabet and $\rho : \Sigma \to \mathbb{N}$ a weight function. We extend ρ to words $a_1 \cdots a_k \in \Sigma^*$ by setting $\rho(a_1 \cdots a_k) := \rho(a_1) + \ldots + \rho(a_k)$.[3] The mean weight of a non-empty word w is defined as $\overline{\rho}(w) := \rho(w)/|w|$. The throughput of a non-empty language $L \subseteq \Sigma^+$ is defined as the infimum of the mean weights of the words in L: $tp(L) := \inf\{\overline{\rho}(w) \mid w \in L\}$. Let $G = (\Sigma, \mathcal{X}, P, S)$ be a context-free

[3] We write + for the addition of reals in this section.

grammar and $L = L(G)$ its language. The problem is to compute $tp(L)$. As in [CCFR07] we assume that G has at most 2 symbols on the right hand side of every production and that L is non-empty and contains only non-empty words.

Note that we cannot simply construct a polynomial system having $tp(L)$ as its least fixed-point, as the throughput of two non-terminals is not additive. In [CCFR07] an ingenious algorithm is proposed to avoid this problem. Assume we already know a routine, the *comparing routine*, that decides for a given $t \in \mathbb{Q}$ whether $tp(L) \geq t$ holds. Assume further that this routine has $\mathcal{O}(N^k)$ time complexity for some k. Using the comparing routine we can approximate $tp(L)$ up to any given accuracy by means of binary search. Let $d = \max_{a\in\Sigma} \rho(a) - \min_{a\in\Sigma} \rho(a)$. A dichotomy result of [CCFR07] shows that $\mathcal{O}(N + \log d)$ iterations of binary search suffice to approximate $tp(L)$ up to an ε that allows to compute the exact value of $tp(L)$ in time $\mathcal{O}(N^3)$. This is proved by showing that, once a value t has been determined such that $t - \varepsilon < tp(L) \leq t$, one can:

- transform G in $\mathcal{O}(N^3)$ time into a grammar G' of size $\mathcal{O}(N^3)$ generating a finite language, and having the same throughput as G (this construction does not yet depend on $tp(L)$);
- compute the throughput of G' in linear time in the size of G', i.e., in $\mathcal{O}(N^3)$ time.

The full algorithm for the throughput runs then in $\mathcal{O}(N^k(N + \log d)) + \mathcal{O}(N^3)$ time.

The algorithm of [CCFR07] and our new algorithm differ in the comparing routine. In the routine of [CCFR07] the transformation of G into the grammar G' is done *before* $tp(L)$ has been determined. Then a linear time algorithm can be applied to G' to decide whether $tp(L) \geq t$ holds. (This algorithm does not work for arbitrary context-free grammars, and that is why one needs to transform G into G'.) Since G' has size $\mathcal{O}(N^3)$, the comparing routine has $k = 3$, and so the full algorithm runs in $\mathcal{O}(N^4 + N^3 \log d)$ time.

We give a more efficient comparing routine with $k = 2$. Given a $t \in \mathbb{Q}$, assign to each word $w \in \Sigma^+$ its *throughput balance* $\sigma_t(w) = \rho(w) - |w| \cdot t$. Notice that $\sigma_t(w) \geq 0$ if and only if $\overline{\rho}(w) \geq t$. Further, for two words w, u we now have $\sigma_t(wu) = \sigma_t(w) + \sigma_t(u)$. So we can set up a polynomial system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ over the tropical semiring $\mathcal{S}_{\min}$ where $\boldsymbol{f}$ is constructed such that each variable $X \in \mathcal{X}$ in the equation system corresponds to the minimum (infimum) throughput balance of the words derivable from X. More formally, define a map m by setting $m(a) = \rho(a) - t$ for $a \in \Sigma$ and $m(X) = X$

for $X \in \mathcal{X}$. Extend m to words in $(\Sigma \cup \mathcal{X})^*$ by setting $m(\alpha_1 \cdots \alpha_k) = m(\alpha_1) + \cdots + m(\alpha_k)$. Let P_X be the productions of G with X on the left hand side. Then set $\boldsymbol{f}_X(\boldsymbol{X}) := \min\{m(w) \mid (X \to w) \in P_X\}$. For instance, if P_X consists of the rules $X \to aXY$ and $X \to bZ$, we have $\boldsymbol{f}_X(\boldsymbol{X}) = \rho(a) - t + X + Y \sqcap \rho(b) - t + Z$.

It is easy to see that the relevant solution of the system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ is the least one w.r.t. $\sqsubseteq$, i.e., $(\mu\boldsymbol{f})_S \geq 0$ if and only if $tp(L) \geq t$. So we can use the algorithm from Section 4.3.1 as our comparing routine. This takes time $\mathcal{O}(N^2)$ where N is the size of the grammar. With that comparing routine we obtain an algorithm for computing the throughput with $\mathcal{O}(N^3 + N^2 \log d)$ runtime.

4.4 Lossy Semirings

Definition 4.4.1.
An io-semiring $\mathcal{S}$ is called *lossy* if $1 \sqsubseteq a$ holds for all $a \neq 0$. $\diamond$

Note that by definition of natural order the requirement $1 \sqsubseteq a$ is equivalent to $a = a+1$. If we interpret this equation on the language semiring generated by some alphabet Σ, this becomes $\{a\} = \{a, \varepsilon\}$ which means that we may replace any letter a by ε. Hence, every language $L \subseteq \Sigma^*$ is "downward closed", i.e., for every word $w = a_1 a_2 \ldots a_l \in L$ all possible subwords $\{a'_1 a'_2 \ldots a'_l \mid a'_i \in \{\varepsilon, a_i\}\}$ are also included in L. By virtue of Higman's lemma [Hig52] the downward-closure of a context-free language is regular. This has been used in [ABJ98] for an efficient analysis of systems with unbounded, lossy FIFO channels. Downward closure was used there to model the loss of messages due to transmission errors.

Recall that a system $\boldsymbol{f}$ of polynomials is *clean* if $\mu\boldsymbol{f}_X \neq 0$ for all $X \in \mathcal{X}$. Every system can be *cleaned* in linear time by removing the equations of all variables X such that $\mu\boldsymbol{f}_X = 0$ and setting these variables to 0 in the other equations. We consider only clean systems, and introduce a normal form for them.

Definition 4.4.2.
Let $\boldsymbol{f} \in \mathcal{S}[\mathcal{X}]^{\mathcal{X}}$ be a system of polynomials over a lossy semiring. $\boldsymbol{f}$ is in *quadratic normal form* if every polynomial $\boldsymbol{f}_X$ has the form

$$c + \sum_{Y,Z \in \mathcal{X}} a_{Y,Z} \cdot Y \cdot Z + \sum_{Y \in \mathcal{X}} b_{l,Y} \cdot Y \cdot b_{r,Y}$$

where (i) $c \in S \setminus \{0\}$, (ii) $a_{Y,Z} \in \{0,1\}$, and (iii) if $\sum_{Z \in \mathcal{X}} a_{Y,Z} \neq 0$, then $b_{l,Y} \neq 0 \neq b_{r,Y}$ for all $Y, Z \in \mathcal{X}$. ◇

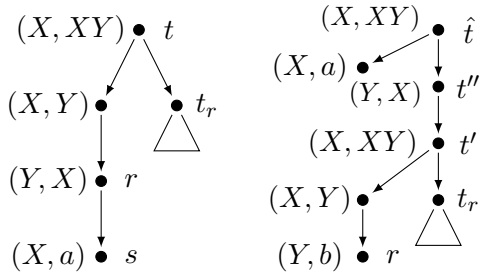
Lemma 4.4.3.
For every clean $\boldsymbol{g} \in \mathcal{S}[\mathcal{X}]^{\mathcal{X}}$ we can construct in linear time a system $\boldsymbol{f} \in \mathcal{S}[\mathcal{X}']^{\mathcal{X}'}$ in quadratic normal form such that $\mathcal{X} \subseteq \mathcal{X}'$ and $\mu\boldsymbol{g}_X = \mu\boldsymbol{f}_X$ for all $X \in \mathcal{X}$. ◇

Proof sketch. Note that, as $\boldsymbol{g}$ is clean, we have $\mathbf{1} \sqsubseteq \mu\boldsymbol{g}$. Hence, requirement (i) is no restriction. The transformation that normalizes a system is similar to the one that brings a context-free grammar into Chomsky normal-form (CNF). The superset $\mathcal{X}' \supset \mathcal{X}$ results from the introduction of new variables by this transformation into CNF. □

Our main result in this section is that for *strongly-connected* systems $\boldsymbol{f}$ (cf. Definition 2.2.14) in quadratic normal form we again have that $\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$. We then show how this result leads to an algorithm for arbitrary systems.

Theorem 4.4.4.
$\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$ holds for strongly-connected polynomial systems $\boldsymbol{f}$ in quadratic normal form over lossy semirings. ◇

Proof sketch. We consider a concrete example of a tree t that is not a bamboo, and show how to construct a bamboo $\hat{t}$ such that $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\hat{t})$. The general procedure for all non-bamboos can be found in the appendix. Let $\mathcal{X} = \{X, Y\}$, and $\boldsymbol{f}$ with $\boldsymbol{f}_X = XY + X + Y + a$, and $\boldsymbol{f}_Y = X + Y + b$. Consider the X-tree t depicted on the left of the picture below, where t_r is some bamboo of height at least 2 (we inductively assume that the original subtree has already been replaced by a bamboo with at least the same yield). Since the left subtree of t has height 2, t itself is not a bamboo.



Let s denote the left-most leaf of t, and let r be the parent of s. In our example, we assume that r has s as its only child. Then we proceed as follows:

(i) We remove from t the leaf s, and turn its father r into a leaf. Here, we make use of the assumption that $\boldsymbol{f}$ is in quadratic normal form, and so every polynomial of $\boldsymbol{f}$ contains a constant monomial, in our example b. We change the monomial-label of r to b, and obtain the tree t', which is a derivation tree of $\boldsymbol{f}$. Moreover, t' is a bamboo, because its left subtree has now height 1, and its right subtree t_r is a bamboo.

(ii) We prepend a (partial) derivation tree on top of t_r having two linear chains as subtrees: the left chain leads to the leaf s, and the right chain leads to t'. This gives us the tree $\hat{t}$ depicted on the right of the picture above. The proof of Theorem 4.4.4 shows that these chains exist and have at most length $n-1$ (in our example $n-1=1$). It follows that $\hat{t}$ is a bamboo itself, and so $\mathsf{Y}(\hat{t}) \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$.

We have $\mathsf{Y}(t) = a \cdot \mathsf{Y}(t_r)$ and $\mathsf{Y}(\hat{t}) = a \cdot b \cdot \mathsf{Y}(t_r)$. Since the semiring is lossy, we have $1 \sqsubseteq b$ and so $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\hat{t})$. Notice that, since product is not necessarily commutative, it is important that a is the first factor of both yields. □

Because of the preceding theorem, given a strongly connected system $\boldsymbol{f}$, we may use the linear system $\boldsymbol{f}_{\mathcal{B}}(\boldsymbol{X}) = \boldsymbol{f}(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}(\boldsymbol{X})$ for calculating $\mu\boldsymbol{f}$. As $\boldsymbol{f}$ is strongly connected, $\boldsymbol{f}_{\mathcal{B}}$ is also strongly connected. The least fixed point of such a strongly connected linear system $\boldsymbol{f}_{\mathcal{B}}$ is easily calculated: all non-constant monomials appearing in $\boldsymbol{f}_{\mathcal{B}}$ have the form $b_l X b_r$ for some $X \in \mathcal{X}$, and $b_l, b_r \in S \setminus \{0\}$. As $\boldsymbol{f}_{\mathcal{B}}$ is strongly connected, every polynomial $(\boldsymbol{f}_{\mathcal{B}})_Y$ is substituted for Y in $(\boldsymbol{f}_{\mathcal{B}})_X$ again and again when calculating the Kleene sequence $(\boldsymbol{f}_{\mathcal{B}}^k(\boldsymbol{0}))_{k\in\mathbb{N}}$. So, let l be the sum of all left-handed coefficients b_l (appearing in *any* $\boldsymbol{f}_X$), and similarly define r. We then have

$$(\mu\boldsymbol{f}_{\mathcal{B}})_X = l^*\big(\sum_{Y\in\mathcal{X}} \boldsymbol{f}_Y(\boldsymbol{0})\big)r^*$$

for all $X \in \mathcal{X}$.

If $\boldsymbol{f}$ is not strongly connected, we first decompose $\boldsymbol{f}$ into strongly connected subsystems, and then we solve these systems bottom-up. Note that substituting the solutions from underlying SCCs into a given SCC leads to a new system in normal form. As there are at most $n = |\mathcal{X}|$ many strongly connected components for a given system $\boldsymbol{f} \in \mathcal{S}[\mathcal{X}]^{\mathcal{X}}$, we obtain the following theorem which was first stated explicitly for context-free grammars in [Cou91].

Theorem 4.4.5.
The least fixed-point $\mu\boldsymbol{f}$ of a polynomial system $\boldsymbol{f}$ over a lossy semiring is representable by regular expressions over $\mathcal{S}$. If $\boldsymbol{f}$ is in normal form $\mu\boldsymbol{f}$ can be calculated by solving at most n bamboo systems. ◇

4.5 1-bounded Semirings

Definition 4.5.1.
An io-semiring $\mathcal{S}$ is called 1-*bounded* if $a \sqsubseteq 1$ holds for all $a \in S$. ◇

Natural examples are the $(\min, +)$-semiring restricted to the natural numbers $\langle \mathbb{N} \cup \{\infty\}, \sqcap, +, 0, \infty \rangle$ and the "maximum-probability" semiring $\langle [0,1], \sqcup, \cdot, 0, 1 \rangle$, where $\sqcap$ and $\sqcup$ denote minimum and maximum, respectively. Notice that any commutative 1-bounded semiring is star-distributive (as $a^* = 1$ for all a), but not all 1-bounded semirings have commutative multiplication. Consider for example the semiring of those languages L over Σ that are *upward-closed*, i.e., $w \in L$ implies $u \in L$ for all u such that w is a subword of u. This semiring is 1-bounded and has Σ^* as 1-element. Upward-closed languages form a natural dual to downward-closed languages from the previous section.

We show that $\mu\boldsymbol{f}$ can be computed very easily in the case of 1-bounded semirings:

Theorem 4.5.2.
$\mu\boldsymbol{f} = \boldsymbol{f}^n(\mathbf{0})$ holds for polynomial systems over 1-bounded semirings. $\diamond$

Proof sketch. Recall that, by Proposition 3.2.5, we have $(\boldsymbol{f}^n(\mathbf{0}))_X = \mathsf{Y}(\mathcal{H}_X^{(n-1)})$, where $\mathcal{H}_X^{(n-1)}$ contains all X-trees of height at most $n-1$. We proceed by derivation tree analysis, i.e., we show that for any X-tree t there is an X-tree t' of height at most $n-1$ with $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(t')$. As long as some variable label occurs at least twice along any path, we can construct from t such a tree t' by pruning. $\square$

Theorem 4.5.2 appears to be rather easy from our point of view, i.e., from the point of view of derivation trees. However, even this simple result has very concrete applications in the domain of interprocedural program analysis [RSJM05]. The main algorithms of [RSJM05], the so-called *post** and *pre** algorithms, can be seen as solvers of fixed-point equations over *bounded* semirings, which are semirings that do not have infinite ascending chains. Those solvers are based on Kleene's iteration and the complexity result given there depends on the maximal length of ascending chains in the semiring (cf. [RSJM05], page 28). Such a bound may not exist, and does not exist for the tropical semiring over the natural numbers $(\mathbb{N} \cup \{\infty\}, \sqcap, +, \infty, 0)$ which is considered as an example in [RSJM05], pages 13 and 18. However, Theorem 4.5.2 can be applied to this semiring, which shows that the program analysis algorithms of [RSJM05] applied to 1-bounded semirings are polynomial-time algorithms, independent of the length of chains in the semiring.

Chapter 5

Min-Max-Systems and Strategy Iteration

5.1 Introduction

In the previous chapter we introduced star-distributive semirings, i.e., commutative, idempotent ω-continuous semirings satisfying the additional axiom $(a+b)^* = a^* + b^*$ for all semiring elements a, b. We obtained Theorem 4.3.4 saying that in any star-distributive semiring $\mathcal{S}$ we have

$$\mu\boldsymbol{f} = \mu\big(\lambda\boldsymbol{X}.\boldsymbol{f}^n(\mathbf{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}(\boldsymbol{X})\big) = D\boldsymbol{f}|^n_{\boldsymbol{f}^n(\mathbf{0})}(\boldsymbol{f}^n(\mathbf{0}))$$

for any polynomial system $\boldsymbol{f}$ given in n variables $\boldsymbol{X}$. Specifically, any *totally ordered* cio-semiring, i.e., the natural order $\sqsubseteq$ is total, is by Proposition 4.3.2 star-distributive. As an example we have considered the $(\min, +)$-semiring in Subsection 4.3.1. With $\sqsubseteq$ total, one easily confirms that the addition on the semiring becomes the maximum w.r.t. $\sqsubseteq$, and one may naturally consider the minimum w.r.t. $\sqsubseteq$ as a "dual addition". In this chapter we study *min-max-systems*, i.e., an extension of polynomial systems where both maximum and miminum is used as semiring addition.

In Section 5.2 we explicitly consider totally ordered cio-semirings. By virtue of our result on star-distributive semirings, we immediately obtain that the least fixed point of a min-max-system on a totally ordered cio-semiring exists and can be calculated. These results serve as motivation for the subsequent sections. In Section 5.3 we analyze a well-known technique for iteratively solving min-max-systems, called *strategy iteration*. We then study the con-

nection between linear min-max-systems and games in Section 5.4. We illustrate our results in Section 5.5 by applying them to parity games.

5.2 Min-Max-Systems on Totally Ordered cio-Semirings

Assume for this section that $\mathcal{S} = \langle S, +, \cdot, 0, 1\rangle$ is a totally ordered cio-semiring. We then have $a + b = \bigsqcup\{a, b\}$, and emphasize this by writing $a \sqcup b$ instead of $a + b$ and $\bot$ instead of 0. We then may define the minimum-operation $\sqcap$:

Definition 5.2.1.
Let $\mathcal{S} = \langle S, +, \cdot, 0, 1\rangle$ be a totally ordered semiring. Then the mimimum $a \sqcap b$ of two elements $a, b \in S$ is defined by

$$a \sqcap b := \begin{cases} a & \text{if } a \sqsubseteq b \\ b & \text{else} \end{cases} \qquad \diamond$$

We note some easy to check facts about the minimum for ω-continuous semirings. The proofs can be found in the appendix.

Proposition 5.2.2.
Let $\langle S, \sqcup, \cdot, \bot, 1\rangle$ be a totally ordered cio-semiring. Define $a \sqcap b$ as stated above. Let $a, b, c \in S$ and $(a_i)_{i\in\mathbb{N}}$ an ω-chain. We then have:

(1) $\sqcup$ and $\sqcap$ distribute:

$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c), \text{ and } a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c).$$

(2) $\cdot$ distributes over $\sqcap$: $a \cdot (b \sqcap c) = (a \cdot b) \sqcap (a \cdot c)$.

(3) $\sqcap$ is ω-continuous (w.r.t. $\sqsubseteq$):

$$c \sqcap \left(\bigsqcup_{i\in\mathbb{N}} a_i \right) = \bigsqcup_{i\in\mathbb{N}} (c \sqcap a_i).$$

(4) $\sqcap$ is monotone: $b \sqsubseteq c \Rightarrow a \sqcap b \sqsubseteq a \sqcap c$. $\diamond$

As already stated in the introduction, we extend polynomial systems to min-max-systems by allowing the use of both maximum, i.e., the addition on the semiring, and minimum as defined above.

Definition 5.2.3.
Let $\mathcal{S} = \langle S, \sqcup, \cdot, \bot, 1\rangle$ be a totally ordered semiring, and $\mathcal{X}$ be a finite set of variables. A *min-max-system* $\boldsymbol{F}$ on $\mathcal{S}$ is then defined as follows: For $X \in \mathcal{X}$ we either have

$$\boldsymbol{F}_X(\boldsymbol{X}) = m_1^{(X)}(\boldsymbol{X}) \sqcup m_2^{(X)}(\boldsymbol{X}) \sqcup \ldots \sqcup m_{k_{\boldsymbol{F}}(X)}(\boldsymbol{X})$$

or

$$\boldsymbol{F}_X(\boldsymbol{X}) = m_1^{(X)}(\boldsymbol{X}) \sqcap m_2^{(X)}(\boldsymbol{X}) \sqcap \ldots \sqcap m_{k_{\boldsymbol{F}}(X)}(\boldsymbol{X})$$

where $m_i^{(X)}$ is a monomial, and $k_{\boldsymbol{F}}(X) \geq 1$ is the number of distinct monomials $m_i^{(X)}$ appearing in $\boldsymbol{F}_X$. We assume that at most one monomial $m_i^{(X)}$ is constant.

We say that $X \in \mathcal{X}$ is a *min-variable (or $\sqcap$-variable)* if $\boldsymbol{F}_X$ is non-constant and the minimum of at least two distinct monomials. Otherwise X is a *max-variable (or $\sqcup$-variable)*. We write $\mathcal{X}_\sqcup$, resp. $\mathcal{X}_\sqcap$ for the set of max-, resp. min-variables.

For technical reasons we require that a constant monomial appears explicitly in $\boldsymbol{F}_X$ if X is a max-variable ([1]).

We say that $\boldsymbol{F}$ is *linear* if all monomials $m_i^{(X)}$ have degree at most 1. ◇

Example 5.2.4. Consider the semiring $\langle \mathbb{Z} \cup \{\pm\infty\}, \max, +, -\infty, 0\rangle$. Then $a \sqcap b = \min\{a, b\}$ for all $a, b \in \mathbb{Z} \cup \{\pm\infty\}$. On this semiring we may consider the min-max-system

$$\boldsymbol{F} = (\boldsymbol{F}_X, \boldsymbol{F}_Y) = \big(\min\{Y, 1\}, \max\{X + X, -1\}\big).$$

Note that $\boldsymbol{F}$ is non-linear, as $+$ is the multiplication of the semiring. The least fixed point of this systems is $X = -1, Y = -1$. ◇

We use capital letters for min-max-systems in order to emphasize that these are not simple polynomial systems. Again, we are interested in calculating $\mu\boldsymbol{F}$. By Proposition 5.2.2 we know that min-max-Systems are ω-continuous and monotone. So, Kleene's fixed-point theorem also holds for these, i.e., $\mu\boldsymbol{F}$ always exists with $\mu\boldsymbol{F} = \bigsqcup_{k\in\mathbb{N}} \boldsymbol{F}^k(\bot)$. Consider now the equation $\boldsymbol{F}(\mu\boldsymbol{F}) = \mu\boldsymbol{F}$. By definition of $\sqcap$, we then find for every $X \in \mathcal{X}_\sqcap$ a monomial m_X of $\boldsymbol{F}_X$ such that $\boldsymbol{F}_X(\mu\boldsymbol{F}) = m_X(\mu\boldsymbol{F}) = (\mu\boldsymbol{F})_X$. Define $\boldsymbol{G}$ by $\boldsymbol{G}_X = \boldsymbol{F}_X$ for $X \in \mathcal{X}_\sqcup$, and $\boldsymbol{G}_X = m_X$ for $X \in \mathcal{X}_\sqcap$. We then have $\boldsymbol{F} \sqsubseteq \boldsymbol{G}$ on $S^{\mathcal{X}}$ by construction, and so $\mu\boldsymbol{F} \sqsubseteq \mu\boldsymbol{G}$ follows. On the other hand, we also have $\boldsymbol{G}(\mu\boldsymbol{F}) = \mu\boldsymbol{F}$ by construction. This yields $\mu\boldsymbol{F} = \mu\boldsymbol{G}$. As $\boldsymbol{G}$ is a polynomial system, i.e., $\sqcap$ does not appear in $\boldsymbol{G}$, we know by Theorem 4.3.4 how to

[1] We may always add the constant $\bot$ if necessary.

calculate $\mu\boldsymbol{G}$ efficiently. From this we obtain the result that $\mu\boldsymbol{F}$ can be represented by means of regular expressions.

In order to formalize the results we have just sketched, we introduce *deterministic $\sqcap$-strategies* which capture the construction of $\boldsymbol{G}$.

Definition 5.2.5.
For a given min-max-system $\boldsymbol{F}$ a *deterministic ([2]) $\sqcap$-strategy (or min-strategy)* τ chooses for any $X \in \mathcal{X}_\sqcap$ exactly one monomial of $\boldsymbol{F}_X$. Given a $\sqcap$-strategy τ we denote by $\boldsymbol{F}_\tau$ the operator we obtain from $\boldsymbol{F}$ by replacing the component $\boldsymbol{F}_X$ by the monomial $m^{(X)}_{\tau(X)}$ for any $X \in \mathcal{X}_\sqcap$, i.e.,

$$\forall X \in \mathcal{X} : (\boldsymbol{F}_\tau)_X := \begin{cases} \boldsymbol{F}_X & \text{if } X \in \mathcal{X}_\sqcup \\ \tau(X) & \text{if } X \in \mathcal{X}_\sqcap \end{cases}$$

We say that τ is optimal if $\mu\boldsymbol{F} = \mu\boldsymbol{F}_\tau$. ◇

Remark 5.2.6.
Deterministic $\sqcup$-strategies are defined analogously. ◇

Example 5.2.7. Consider the min-max-system of Example 5.2.4. Here, we only have two possibilities for defining a deterministic $\sqcap$-strategy, i.e., τ chooses for X either the monomial Y or the monomial 1. In the latter case we obtain the system

$$X = 1 \quad Y = \max\{X + X, -1\}).$$

Obviously, the least solution is $X = 1, Y = 2$. Assume now that τ chooses Y for X, i.e.,

$$X = Y \quad Y = \max\{X + X, -1\}.$$

Here, the least solution is $X = -1, Y = -1$, i.e., the optimal $\sqcap$-strategy is to choose Y for X. ◇

Our result on the existence of $\mu\boldsymbol{F}$ can therefore be stated as follows:

Theorem 5.2.8.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be min-max-system on a totally ordered cio-semiring with carrier S and $n := |\mathcal{X}|$. Then there exists an optimal $\sqcap$-strategy τ with

$$\mu\boldsymbol{F} = \mu\boldsymbol{F}_\tau = D\boldsymbol{F}_\tau \,|^n_{\boldsymbol{F}^n_\tau(\bot)} (\boldsymbol{F}^n_\tau(\bot)).$$ ◇

As there are only finitely many $\sqcap$-strategies, we always can find an optimal $\sqcap$-strategy. A bound on the number of $\sqcap$-strategies is e.g.

$$\prod_{X \in \mathcal{X}_\sqcap} k_{\boldsymbol{F}}(X) \leq \left(\frac{k^\sqcap_{\boldsymbol{F}}}{|\mathcal{X}_\sqcap|}\right)^{|\mathcal{X}_\sqcap|} \text{ with } k^\sqcap_{\boldsymbol{F}} := \sum_{X \in \mathcal{X}_\sqcap} k_{\boldsymbol{F}}(X).$$

[2] We drop "deterministic" for the rest of this section.

Instead of naively enumerating all $\sqcap$-strategies, one approach often used is called *strategy iteration* or *strategy improvement*. In our setting of $\sqcap$-strategies, this approach can be described in its basic form as follows:

Given a ($\sqcap$-)strategy τ, one tries to deduce from the $\mu\boldsymbol{F}_\tau$ a new strategy τ' yielding a "better" approximation, i.e., $\mu\boldsymbol{F}_{\tau'} \sqsubseteq \mu\boldsymbol{F}_\tau$. For this, consider $\boldsymbol{F}(\mu\boldsymbol{F}_\tau)$. If $\mu\boldsymbol{F}_\tau$ is a fixed-point of $\boldsymbol{F}$, then we stop; otherwise we have $\boldsymbol{F}(\mu\boldsymbol{F}_\tau) \sqsubset \mu\boldsymbol{F}_\tau$, i.e., there is at least one $\sqcap$-variable X with $\boldsymbol{F}(\mu\boldsymbol{F}_\tau)_X \sqsubset (\mu\boldsymbol{F}_\tau)_X$ (equality has to hold for all $\sqcup$-variables obviously). So, we may change τ on these $\sqcap$-variables where equality does not hold, and obtain a new strategy τ' with $\boldsymbol{F}_{\tau'}(\mu\boldsymbol{F}_\tau) \sqsubset \mu\boldsymbol{F}_\tau$. We then calculate $\mu\boldsymbol{F}_{\tau'}$ in order to obtain a new and better approximation of $\mu\boldsymbol{F}$.

This iteration is guaranteed to terminate, as there are only finitely many $\sqcap$-strategies, and every $\sqcap$-strategy is considered at most once, as the sequence of approximations $\mu\boldsymbol{F}_\tau$ is strictly decreasing. The weak spot here is that the end result is some fixed point of $\boldsymbol{F}$, but not necessarily the least fixed point, as shown in the following example taken from [GS07]:

Example 5.2.9. We return to the min-max-system of Example 5.2.4

$$\boldsymbol{F} = (\boldsymbol{F}_X, \boldsymbol{F}_Y) = \big(\min\{Y, 1\}, \max\{X + X, -1\}\big) \text{ with } \mu\boldsymbol{F} = (-1, -1).$$

Consider the $\sqcap$-strategy τ which chooses 1 for the min-variable X. This yields the polynomial system

$$\boldsymbol{F}_\tau = \big(1, \max\{X + X, -1\}\big) \text{ with } \mu\boldsymbol{F}_\tau = (1, 2).$$

Further, $\mu\boldsymbol{F}_\tau$ is already a fixed point of $\boldsymbol{F}$. So, $\mu\boldsymbol{F}_\tau$ does not indicate any improvements of τ, i.e., the strategy iteration terminates yielding a fixed point of $\boldsymbol{F}$, but not the least. $\diamond$

In the next section we discuss how the heuristic of strategy iteration may be refined in order to yield the least fixed point.

Before doing so, a short remark on the usefulness of strategy iteration is due, as strategy iteration itself might still lead to inspecting all, i.e., exponentially many strategies, thus gaining no advantage over inspecting all possible strategies by a brute-force approach:
The technique of strategy iteration is encountered quite often in literature, especially in the context of games, like Markov decision processes [How60], stochastic games [HK66], discounted payoff games [Pur95], mean payoff games [ZP96], or parity games [VJ00, Sch07], but also in the context of static analysis [GS07, GS08]. Judging by the experiments done in these articles, it seems that "hard" instances of min-max-systems are quite rare in practice. For example, in [GS08] it is reported that all instances encountered in their experiments needed only $O(|\mathcal{X}|)$ many iterations. It was also

a long standing problem if the strategy improvement algorithm by Jurdzinski and Vöge [VJ00] runs in polynomial time. Only recently, it was shown in [Fri09] that there is indeed a family of parity games where both the algorithm of [VJ00] and the algorithm of [Sch07] require a superpolynomial number of iterations until $\mu\boldsymbol{F}$ is reached. On the other hand, for parity games it is only known that solving parity games is included in NP∩co-NP, or more precisely UP∩co-UP [Jur98], but it is still an open problem whether parity games can be solved within polynomial time. As we will see, parity games are a special case of min-max-system (see Section 5.5), so solving min-max-systems in general is at least as hard as solving parity games. In absence of an efficient algorithm and motivated by the applicability of strategy iteration in practice, it therefore seems worthwhile to study the question of how and when strategy iteration may be used for solving min-max-systems.

5.3 Strategy Iteration and Semirings

In the previous section we have introduced the idea of iteratively improving a strategy in order to calculate the least fixed point of a min-max-system. We have also seen that the most basic realization of this idea does not work in general. In the following we consider a class of totally ordered semirings, which we simply call *strategy-iteration semirings* or short *si-semirings*, and we show that on these $\sqcup$-strategies can be iteratively improved in order to calculate $\mu\boldsymbol{F}$. We give a formal definition later on, for now we only like to remark that si-semirings are not required to be ω-continuous. Hence, the existence of the least fixed point of a min-max-system is not guaranteed. We also note that the semiring $\langle \mathbb{Z} \cup \{\pm\infty\}, \max, +, -\infty, 0\rangle$ is a semiring. From Example 5.2.9 it therefore follows that also on si-semirings ($\sqcap$-)strategy iteration in its basic form does not yield the least fixed point in general.

The strategy iteration proposed in this section is based on the work by Gawlitza and Seidel [GS08] for solving min-max-systems on the integers. We extend their approach to the more general setting of si-semirings, and further consider a more permissive class of ($\sqcup$-)strategies in contrast to [GS08]: We allow that a $\sqcup$-, resp. $\sqcap$-strategy not only selects exactly one monomial of $\boldsymbol{F}_X$ for every $X \in \mathcal{X}_\sqcup$, resp. $X \in \mathcal{X}_\sqcap$, but we allow that it selects a nonempty subset of these monomials. We call these strategies *nondeterministic* (see Definition 5.3.10) and denote by $\boldsymbol{F}_\sigma$ again the min-max-system induced by σ.

We proceed as follows: In the next Subsection 5.3.1 we define si-semirings

formally, and discuss several properties of min-max-systems on this particular class of semirings. We then turn to nondeterministic strategies and introduce the subclass of *reasonable* strategies. A reasonable strategy σ has several nice properties: Its greatest fixed point $\nu\boldsymbol{F}_\sigma$ exists and can be easily calculated; further $\nu\boldsymbol{F}_\sigma$ is a lower bound on any fixed point of the given min-max-system $\boldsymbol{F}$. This is done in Subsection 5.3.2. In Subsection 5.3.3 we then describe our strategy iteration and show that the strategies appearing in a strategy iteration are all reasonable. From this we then obtain our main result that the least fixed point of a min-max-system on a si-semiring exists and can be calucated by means of the proposed strategy iteration.

5.3.1 Si-Semirings and Min-Max-Systems

We start with the definition of si-semiring:

Definition 5.3.1.
$\mathcal{S} = \langle S, \cdot, \sqsubseteq, 1, \bot, \top\rangle$ is a *si-semiring* if it satisfies the following conditions:

(1) $\langle S, \sqsubseteq\rangle$ is a totally ordered set with $\bot, \top \in S$ such that

$$\bot \sqsubseteq a \sqsubseteq \top \text{ for all } a \in S.$$

(2) $\langle S, \cdot, 1\rangle$ is a commutative monoid with

(a) $\bot \cdot a = \bot$ for all $a \in S$.

(b) $\top \cdot a = \top$ for all $a \in S \setminus \{\bot\}$.

(c) $a \sqsubseteq b \Rightarrow c \cdot a \sqsubseteq c \cdot b$ for all $a, b, c \in S$.

(d) $a \sqsubset b \Rightarrow c \cdot a \sqsubset c \cdot b$ for all $a, b \in S$ and $c \in S \setminus \{\bot, \top\}$.

We denote by $a \sqcup b$, resp. $a \sqcap b$ the maximum, resp. minimum of $a, b \in S$ w.r.t. the total order $\sqsubseteq$ on S. ◇

Remark 5.3.2.
Note that (c) and (d) together imply that

$$c \cdot a = c \cdot b \Rightarrow a = b \text{ if } c \neq \bot, \top.$$

This means that $\langle S \setminus \{\bot, \top\}, \cdot, 1\rangle$ has the cancellation property. As multiplication is commutative it can therefore be embedded into a linearly-ordered abelian group via the Grothendieck construction [Ati67]. It is known that linearly-ordered abelian groups which have the *Archimedean property*[3]

[3] I.e., there are no elements x, y s.t. $x^n < y$ for all $n \in \mathbb{N}$

are isomorphic to a subgroup of $\langle \mathbb{R}, +, 0 \rangle$. In general, si-semirings are not Archimedean, for an example see Definition 5.5.4. In particular, there are no zero divisors in a si-semiring. Please not that for most of the following proofs we only need this weaker version of (d):

$$1 \sqsubseteq b \Rightarrow c \sqsubseteq c \cdot b \text{ for } b \in S, c \in S \setminus \{\bot, \top\}. \qquad \diamond$$

The following properties of si-semirings can easily be checked by the reader. A proof can be found in the appendix.

Proposition 5.3.3.
Every si-semiring $\langle S, \cdot, \sqsubseteq, 1, \bot, \top \rangle$ has the following properties:

(1) $\cdot$ distributes both over $\sqcup$ and $\sqcap$:

$$a \cdot (b \sqcup c) = (a \cdot b) \sqcup (a \cdot c) \text{ and } a \cdot (b \sqcap c) = (a \cdot b) \sqcap (a \cdot c).$$

(2) $\sqcup$ and $\sqcap$ distribute:

$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c) \text{ and } a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c).$$

(3) $(1 \sqsubset a \wedge a \cdot x \sqsubseteq x) \Rightarrow x \in \{\bot, \top\}$

(4) $x \sqsubset a \cdot x \Rightarrow (x \in S \setminus \{\bot, \top\} \wedge 1 \sqsubset a)$. $\diamond$

Remark 5.3.4.
Every si-semiring is a totally-ordered idempontent and commutative semiring w.r.t. maximum $\sqcup$ as addition. In particular, min-max-systems (Definition 5.2.3) on si-semirings are monotone, but not necessarily ω-continuous. So we may not apply Kleene's fixed point theorem in the following in order to reason about the least fixed point. $\diamond$

We next extend the definition of the dependency relation (see Definition 2.2.14) to min-max-systems and introduce a graphical representation of it.

Definition 5.3.5.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a min-max-system. The *dependency graph* $\mathcal{G}_{\boldsymbol{F}}$ *of* $\boldsymbol{F}$ is the directed, edge labeled graph whose nodes are given by $\mathcal{X}$. There is an edge from X to Y labeled by m (short: $X \xrightarrow{m} Y$) if m is a non-constant monomial of the polynomial $\boldsymbol{F}_X$ and Y appears in m. $\diamond$

Remark 5.3.6.
The shape of a node is used to encode whether it corresponds to a $\sqcup$-variable ($\bullet$) or to a $\sqcap$-variable ($\blacksquare$). See Figure 5.1 for an example. $\diamond$

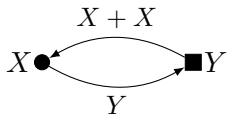


Figure 5.1: The dependency graph of the min-max-system $\boldsymbol{F} = (\boldsymbol{F}_X, \boldsymbol{F}_Y) = \big(\min\{Y, 1\}, \max\{X + X, -1\}\big)$. Circular nodes represent $\sqcup$-variables, box shaped nodes $\sqcap$-variables.

Assume that we are given a min-max-system $\boldsymbol{F}$ and that its least fixed point $\mu\boldsymbol{F}$ exists. Then we can find a $\sqcup$-strategy σ wich chooses for every $X \in \mathcal{X}_\sqcup$ a monomial m from $\boldsymbol{F}_X$ such that $m(\mu\boldsymbol{F}) = (\mu\boldsymbol{F})_X$. The dependency graph of $\boldsymbol{F}_\sigma$ is then a subgraph of $\mathcal{G}_{\boldsymbol{F}}$, and, as we will see, calculating $\mu\boldsymbol{F}$ boils down to identifying the cycles of $\mathcal{G}_{\boldsymbol{F}}$ which also exists in $\mathcal{G}_{\boldsymbol{F}_\sigma}$. The following lemma shows that we can immediately solve a min-max-system whose dependency graph is acyclic.

Lemma 5.3.7.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a min-max-system on some si-semiring such that $\mathcal{G}_{\boldsymbol{F}}$ is acyclic. Then $\boldsymbol{F}$ has a unique fixed point which is given by $\boldsymbol{F}^{|\mathcal{X}|}(\boldsymbol{v})$ for any $\boldsymbol{v} \in S^{\mathcal{X}}$. ◇

Proof. We proceed by induction on $n := |\mathcal{X}|$.

$(n = 1)$. Let $\mathcal{X} = \{X\}$. If $\mathcal{G}_{\boldsymbol{F}}$ is acyclic, then $\boldsymbol{F}_X$ is the minimum or maximum of some finite number of constant monomials. Obviously, we then have $\boldsymbol{F}(\boldsymbol{v}) = \mu\boldsymbol{F} = \nu\boldsymbol{F}$ for all $\boldsymbol{v} \in S^{\mathcal{X}}$.

$(n \to n + 1)$. Let $|\mathcal{X}| = n + 1$. As $\mathcal{G}_{\boldsymbol{F}}$ is acyclic, there exists some variable X which does not have any incoming edges in $\mathcal{G}_{\boldsymbol{F}}$, i.e., there is no monomial in $\boldsymbol{F}$ in which X appears. We therefore may remove $\boldsymbol{F}_X$ from $\boldsymbol{F}$, and obtain a system $\boldsymbol{G}$ given in the variables $\mathcal{X}' = \mathcal{X} \setminus \{X\}$. By induction we have $\mu\boldsymbol{G} = \nu\boldsymbol{G} = \boldsymbol{G}^n(\boldsymbol{v})$. Obviously, we then have that $\mu\boldsymbol{F} = \nu\boldsymbol{F} = \boldsymbol{F}^{n+1}(\boldsymbol{v})$, too. □

In order to be able to control all cycles of $\mathcal{G}_{\boldsymbol{F}}$ by means of max-strategies, we often assume in the following that there are no cycles which only consist of min-variables.

Definition 5.3.8.
A min-max-system $\boldsymbol{F}$ is *min-cycle-free* if $\mathcal{G}_{\boldsymbol{F}}$ restricted to $\mathcal{X}_\sqcap$ is acyclic. ◇

Remark 5.3.9.
In Lemma 5.3.23 we show that w.l.o.g. we may assume that any min-max-system is min-cycle-free. ◇

5.3.2 Nondeterministic and Reasonable Strategies

We define nondeterministic $\sqcup$-strategies. In contrast to deterministic strategies, a nondeterministic $\sqcup$-strategy σ is allowed to choose a nonempty subset $\sigma(X)$ of the monomials appearing in $\boldsymbol{F}_X$.

Definition 5.3.10.
Let $\boldsymbol{F}$ be a min-max-system. A *nondeterministic* max-strategy σ (min-strategy τ) maps every max-variable $X \in \mathcal{X}_\sqcup$ (every min-variable $X \in \mathcal{X}_\sqcap$) to a nonempty subset $\sigma(X)$ ($\tau(X)$) of the monomials of $\boldsymbol{F}_X$. We call σ *deterministic* if $|\sigma(X)| = 1$ for all $X \in \mathcal{X}_\sqcup$. Similarly, deterministic min-strategies are defined.

We denote by $\boldsymbol{F}_\sigma$ the min-max-system induced by the max-strategy σ, i.e.:

$$(\boldsymbol{F}_\sigma)_X := \begin{cases} \sqcup\sigma(X) & \text{if } X \in \mathcal{X}_\sqcup \\ \boldsymbol{F}_X & \text{else} \end{cases}$$

Similarly, a min-strategy τ induces the min-max-system $\boldsymbol{F}_\tau$. ◇

We next introduce a particular class of max-strategies σ called *reasonable* which have the particular property that $\nu\boldsymbol{F}_\sigma$ exists, is given by $\boldsymbol{F}_\sigma^{|\mathcal{X}|}(\top)$ and is always a lower bound on any fixed point of $\boldsymbol{F}$ (cf. Lemma 5.3.12).

Definition 5.3.11.
Let $\boldsymbol{F}$ be a min-max-system. A max-strategy σ is *reasonable*, if there is some *witness* $\boldsymbol{v} \in S^{\mathcal{X}}$ such that:

(1) $\boldsymbol{v} \sqsubseteq \boldsymbol{a}$ for any fixed point $\boldsymbol{a}$ of $\boldsymbol{F}$.

(2) For any cycle $X_0 \xrightarrow{m_0} \dots X_l \xrightarrow{m_l} X_{l+1}$ (with $X_0 = X_{l+1}$) in $\mathcal{G}_{\boldsymbol{F}_\sigma}$:

$$\bot \sqsubset \boldsymbol{v}_{X_0} \text{ and } 1 \sqsubset \prod_{i=0}^{l} D_{X_{i+1}} m_i|_{\boldsymbol{v}}.$$

(3) If $\boldsymbol{F}_\sigma$ is nonlinear, then $\boldsymbol{v} \sqsubseteq \boldsymbol{F}_\sigma(\top)^k$ for all $k \in \mathbb{N}$. ◇

Lemma 5.3.12.
Let $\boldsymbol{F}$ be a min-max-system with $n := |\mathcal{X}|$. For a reasonable max-strategy σ with witness $\boldsymbol{v} \in S^{\mathcal{X}}$ it holds that $\boldsymbol{F}_\sigma^n(\top)$ is the greatest fixed point of $\boldsymbol{F}_\sigma$, i.e., $\nu\boldsymbol{F}_\sigma = \boldsymbol{F}_\sigma^n(\top)$. Further, $\nu\boldsymbol{F}_\sigma \sqsubseteq \boldsymbol{a}$ for any fixed point $\boldsymbol{a}$ of $\boldsymbol{F}$. ◇

Proof. We first show that $\boldsymbol{F}_\sigma^n(\top)$ is indeed the greatest fixed point of $\boldsymbol{F}_\sigma$. For this, we define a context-free grammar G with non-terminals $\mathcal{X}$, terminals the coefficients of $\boldsymbol{F}_\sigma$ extended by $\{\top, \sqcup\}$, and rules as follows:

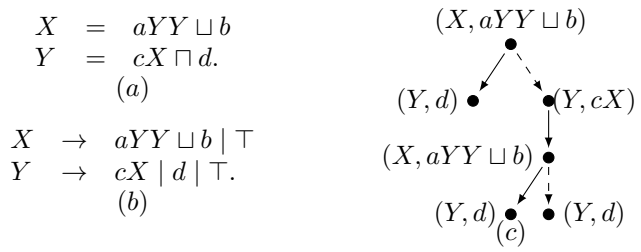


Figure 5.2: From the min-max-system depicted in (a) we obtain the rules of (b). An X-tree of height 3 with yield $adc(add \sqcup b) \sqcup b$ is shown in (c). The dashed arrows indicate a possible factorization of this tree into three trees t_h, t_g and t_r with $g(Y) = adY \sqcup b$, $h(Y) = c(adY \sqcup b)$ and $r = d$. As we know that $\boldsymbol{v}_Y \sqsubseteq d$, we also have $h(Y) \sqsupseteq ca\boldsymbol{v}_Y \cdot Y$. By condition (2) we then may conclude that $ca\boldsymbol{v}_Y \sqsupset 1$, i.e., $h(Y) \sqsupseteq Y$.

- For every $X \in \mathcal{X}$ there is the rule $X \to \top$.
- If $X \in \mathcal{X}_\sqcap$, then for every monomial m of $\boldsymbol{F}_X$ $(= (\boldsymbol{F}_\sigma)_X)$ we have a rule $X \to m$.
- If $X \in \mathcal{X}_\sqcup$, then there we have the rule $X \to (\boldsymbol{F}_\sigma)_X$.

We use this grammar to associate with $\boldsymbol{F}_\sigma$ derivation trees as defined Chapter 3. The yield $\mathsf{Y}(t)$ of a (derivation) tree t is defined to be the value we obtain by evaluting the functions represented by the nodes of the tree in a bottom-up manner. See Figure 5.2 for an example. For $X \in \mathcal{X}$, let $\mathcal{H}_X^k$ denote the X-trees of height at most k. Similarly, we write $\mathcal{H}_{X,\alpha}^k$ for all X-trees of height at most k whose root is labeled by the rule (X, α).

We claim that

$$\boldsymbol{F}_\sigma^{k+1}(\top) = \sqcap \underbrace{\{\mathsf{Y}(t) \mid t \in \mathcal{H}_X^k\}}_{=:\mathsf{Y}(\mathcal{H}_X^k)} \text{ for all } k \in \mathbb{N}.$$

Note that the minimum on the right-hand side always exists, as $\mathcal{H}_X^k$ is a finite set.

We proceed by induction on k:

- $(k = 0)$:

  Consider any min-variable $X \in \mathcal{X}_\sqcap$. There are at most two distinct X-trees of height 0, namely $(X, \top)$ and (X, c) with c the constant in $\boldsymbol{F}_X$. So, $\sqcap \mathsf{Y}(\mathcal{H}_X^0) = \top \sqcap c = c = \boldsymbol{F}_X(\top) = \boldsymbol{F}_\sigma(\top)_X$.

  Similarly, for $X \in \mathcal{X}_\sqcup$ there also at most two distinct X-trees, this time $(X, \top)$ and $(X, \boldsymbol{F}_\sigma(\top)_X)$. Obviously, we have $\sqcap \mathsf{Y}(\mathcal{H}_X^0) = \boldsymbol{F}_\sigma(\top)_X$, too.

- $(k \to k + 1)$:

  We write $\sqcap \mathsf{Y}(\mathcal{H}^k)$ for the vector whose X-component is given by $\sqcap \mathsf{Y}(\mathcal{H}_X^k)$.

  Consider first a min-variable $X \in \mathcal{X}_\sqcap$ and let m be a (non-constant) monomial of $\boldsymbol{F}_X$. As multiplication is assume to be commutative, we may write m as $cX_1 \cdots X_r$ for some $\bot \sqsubset c$ and $r \geq 1$ with $X_1, \ldots, X_r \in \mathcal{X}$. We then have

  $$m(\sqcap \mathsf{Y}(\mathcal{H}^k)) = c \cdot \left(\sqcap \mathsf{Y}(\mathcal{H}_{X_1}^k)\right) \cdots \left(\sqcap \mathsf{Y}(\mathcal{H}_{X_r}^k)\right)$$

$$
\begin{aligned}
&= \big(\sqcap\{c \cdot \mathsf{Y}(t_1) \mid t_1 \in \mathcal{H}^k_{X_1}\}\big) \cdot \big(\sqcap \mathsf{Y}(\mathcal{H}^k_{X_2})\big) \cdots \big(\sqcap \mathsf{Y}(\mathcal{H}^k_{X_r})\big)\\
&= \big(\sqcap\{c \cdot \mathsf{Y}(t_1) \cdot \big(\sqcap \mathsf{Y}(\mathcal{H}^k_{X_2})\big) \mid t_1 \in \mathcal{H}^k_{X_1}\}\big) \cdots \big(\sqcap \mathsf{Y}(\mathcal{H}^k_{X_r})\big)\\
&= \big(\sqcap\{c \cdot \mathsf{Y}(t_1) \cdot \mathsf{Y}(t_2) \mid t_1 \in \mathcal{H}^k_{X_1}, t_2 \in \mathcal{H}^k_{X_2}\}\big) \cdots \big(\sqcap \mathsf{Y}(\mathcal{H}^k_{X_r})\big)\\
&= \ldots\\
&= \sqcap\{c \cdot \mathsf{Y}(t_1) \cdots \mathsf{Y}(t_r) \mid t_1 \in \mathcal{H}^k_{X_1}, \ldots, t_r \in \mathcal{H}^k_{X_r}\}\\
&= \sqcap \mathsf{Y}(\mathcal{H}^{k+1}_{X,m}).
\end{aligned}
$$

It now immediately follows that for $X \in \mathcal{X}_\sqcap$:

$$
\begin{aligned}
\boldsymbol{F}^{k+2}_\sigma(\top)_X &= \boldsymbol{F}_X(\boldsymbol{F}^{k+1}_\sigma(\top))\\
&= \boldsymbol{F}_X(\sqcap \mathsf{Y}(\mathcal{H}^k))\\
&= \sqcap\{m(\sqcap \mathsf{Y}(\mathcal{H}^k)) \mid m \text{ is a non-constant monomial of } \boldsymbol{F}_X\} \sqcap \boldsymbol{F}_X(\top)\\
&= \sqcap\{\sqcap \mathsf{Y}(\mathcal{H}^{k+1}_{X,m}) \mid m \text{ is a non-constant monomial of } \boldsymbol{F}_X\} \sqcap \mathsf{Y}(\mathcal{H}^0_X)\\
&= \sqcap \mathsf{Y}(\mathcal{H}^{k+1}_X).
\end{aligned}
$$

Let us now consider a max-variable $X \in \mathcal{X}_\sqcup$. Here, we have

$$\boldsymbol{F}^{k+2}_\sigma(\top)_X = (\boldsymbol{F}_\sigma)_X(\sqcap \mathsf{Y}(\mathcal{H}^k))$$

We want to show that

$$(\boldsymbol{F}_\sigma)_X(\sqcap \mathsf{Y}(\mathcal{H}^k)) = \sqcap \mathsf{Y}(\mathcal{H}^{k+1}_{X,(\boldsymbol{F}_\sigma)_X}).$$

This follows by structural induction on $(\boldsymbol{F}_\sigma)_X$: If $(\boldsymbol{F}_\sigma)_X$ is a constant, then any X-tree has height 0. We have also already consider the case that it is a non-constant monomial. If $(\boldsymbol{F}_\sigma)_X$ is the maximum of at least two monomials, we split it up into two shorter functions f and g such that $(\boldsymbol{F}_\sigma)_X = f \sqcup g$. By induction we then have

$$f(\sqcap \mathsf{Y}(\mathcal{H}^k)) = \sqcap \mathsf{Y}(\mathcal{H}^{k+1}_{X,f}) \text{ and } g(\sqcap \mathsf{Y}(\mathcal{H}^k)) = \sqcap \mathsf{Y}(\mathcal{H}^{k+1}_{X,g}) \ ([4]).$$

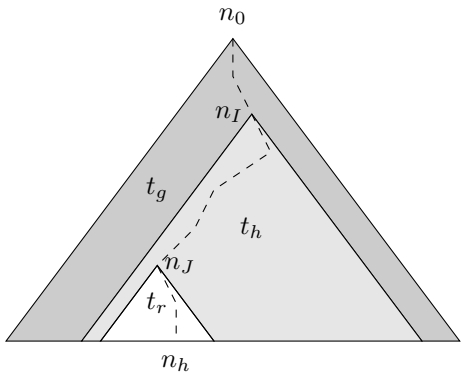
As $\sqcup$ distributes over $\sqcap$, we then obtain

$$(\boldsymbol{F}_\sigma)_X(\sqcap \mathsf{Y}(\mathcal{H}^k)) = \sqcap\{\mathsf{Y}(t) \sqcup \mathsf{Y}(t') \mid t \in \mathcal{H}^{k+1}_{X,f}, t' \in \mathcal{H}^{k+1}_{X,g}\} = \sqcap \mathsf{Y}(\mathcal{H}^{k+1}_{X,(\boldsymbol{F}_\sigma)_X}).$$

With this at hand, we are going to show that for every X-tree t of height at least n there is an X-tree t' of height at most $n-1$ with $\mathsf{Y}(t') \sqsubseteq \mathsf{Y}(t)$. This in turn then implies that $\sqcap \mathsf{Y}(\mathcal{H}^n_X) = \sqcap \mathsf{Y}(\mathcal{H}^{n-1}_X)$, i.e., $\boldsymbol{F}^{n+1}_\sigma(\top) = \boldsymbol{F}^n_\sigma(\top)$.

Consider now any X-tree t of height $h \geq n$. Then there is a path $n_0, \ldots, n_h$ of length h from the root of t to one of its leaves. Along this path $h+1$ variables appear, i.e., we find two nodes n_I and n_J $(I < J)$ labeled by the same variable, say Y.

[4] Here $\mathcal{H}^{k+1}_{X,f}$ denotes the set of trees whose root is labeled by (X, f) and every occurrence of a variable Y in f gives rise to a Y-(sub)-tree of height at most k. Analogously, $\mathcal{H}^{k+1}_{X,g}$ is defined. Technically, the trees of $\mathcal{H}^{k+1}_{X,f}$, resp. $\mathcal{H}^{k+1}_{X,g}$ are no derivation trees w.r.t. $\boldsymbol{F}_\sigma$, but we may "factorize" any tree of $\mathcal{H}^{k+1}_{X,(\boldsymbol{F}_\sigma)_X}$ into two trees t and t' with $t \in \mathcal{H}^{k+1}_{X,f}$ and $t' \in \mathcal{H}^{k+1}_{X,g}$.



Let t_g denote the tree we obtain from t by removing the subtree rooted in n_I (including the node n_I; t_g might therefore be empty). Further, let t_r denote the subtree of t with root n_J. Finally, let t_h be the rest of t after removing both t_g and t_r. Note that t_g and t_h then both represent functions $g(Y)$, resp. $h(Y)$ depending only on Y, while t_r yields some constant $\mathsf{Y}(t_r) = r$. (If t_g is empty, then set $g(Y) := Y$.) In particular, we have $\mathsf{Y}(t) = g(h(r))$. See also Figure 5.2. We show that $h(Y) \sqsupseteq Y$:

Consider the path from the root of t_g to the node which links to t_r, i.e., the subpath $n_I, \ldots, n_{J-1}$. Let $Y_0, \ldots, Y_l$ be the sequence of variables encountered along this path $n_I, \ldots, n_{J-1}$. For every Y_j we choose a monomial m_j as follows: if $Y_j \in \mathcal{X}_\sqcap$, then m_j is simply the monomial represented by the node n_{I+j}; otherwise we choose from $\sigma(Y_j)$ the unique monomial giving rise to the subtree rooted in n_{I+j+1}. We then have that m_j is a non-constant monomial depending on variable Y_{j+1} for $j = 0, \ldots, l$ with $Y_{l+1} = Y_0 = Y$. Further, the yield of the subtree having root n_{I+j} is then always at least the value we obtain from m_j by evaluating the variables of m_j according to yields of the subtrees originating from m_j.

Assume first that some of the monomials m_j are nonlinear. Then by condition (3) of Definition 5.3.11 we know that $\boldsymbol{v}$ is a lower bound on $\boldsymbol{F}^k_\sigma(\top)$ for all $k \in \mathbb{N}$, i.e., $\boldsymbol{v}_X$ is a lower bound on the yield of any X-tree. We therefore may underapproximate the yield of any subtree originating from some monomial m_j, whose root is not located on the path $n_I, \ldots, n_{J-1}$, by means of $\boldsymbol{v}$. It follows that

$$h(Y) \sqsupseteq Y \cdot \prod_{j=0}^{l} D_{Y_{j+1}} m_j|_{\boldsymbol{v}}$$

holds. By condition (2) it also follows that

$$1 \sqsubset \prod_{j=0}^{l} D_{Y_{j+1}} m_j|_{\boldsymbol{v}}.$$

On the other hand, if all monomials m_j are linear, then we do not need to underapproximate the value of any subtree. In particular, $D_{Y_{j+1}} m_j|_{\boldsymbol{v}}$ is simply the coefficient of m_j and thus idependent of the value of $\boldsymbol{v}$. In this case we therefore do not need to require that $\boldsymbol{v}$ is a lowerbound on $\boldsymbol{F}^k_\sigma(\top)$.

Hence, $h(Y) \sqsupseteq Y$ on S, and so

$$\mathsf{Y}(t) = g(h(r)) \sqsupseteq g(r).$$

In particular, by removing the tree t_h from t we obtain a tree t' having exactly yield $g(r)$. We now may prune t' again in this way, if it has height at least n, while not increasing its yield. Eventually, we obtain a tree of height at most $n-1$, whose yield is a lower bound on the yield of the original tree t. So, $\sqcap\mathsf{Y}(\mathcal{H}_X^{n-1}) = \sqcap\mathsf{Y}(\mathcal{H}_X)$ which concludes the first part of the proof that $\boldsymbol{F}_\sigma^n(\top)$ is a fixed point of $\boldsymbol{F}_\sigma$. By monotonicity of $\boldsymbol{F}_\sigma$, it is also the greatest fixed point $\nu\boldsymbol{F}_\sigma$.

We next show that $\nu\boldsymbol{F}_\sigma$ is a lower bound on any fixed point of $\boldsymbol{F}$. For this assume that $\boldsymbol{a}$ is some fixed point of $\boldsymbol{F}$. Recall that condition (1) requires that $\boldsymbol{v} \sqsubseteq \boldsymbol{a}$ holds.

We now choose some deterministic $\sqcap$-strategy τ satisfying

$$\boldsymbol{F}_{\sigma\tau}(\boldsymbol{a}) = \boldsymbol{F}_\sigma(\boldsymbol{a}).$$

Then $\mathcal{G}_{\boldsymbol{F}_{\sigma\tau}}$ is a subgraph of $\mathcal{G}_{\boldsymbol{F}_\sigma}$ and our first goal is to show that for any $X \in \mathcal{X}$ which is located in some cycle of $\mathcal{G}_{\boldsymbol{F}_{\sigma\tau}}$ we have $\boldsymbol{a}_X = \top$.

Let $X_0 \xrightarrow{m_0} X_1 \xrightarrow{m_1} \dots X_l \xrightarrow{m_l} X_{l+1}$ be a simple cycle in $\mathcal{G}_{\boldsymbol{F}_{\sigma\tau}}$ ($X_0 = X_{l+1}$). In particular all monomial m_i are non-constant and X_{i+1} appears in m_i.

Let $\boldsymbol{g}$ denote the following function on $S^{\mathcal{X}}$:

$$\begin{array}{lcll} \boldsymbol{g}_{X_i} & := & D_{X_{i+1}} m_i|_{\boldsymbol{v}} \cdot X_{i+1} & \text{for } i \in \{0,\dots,l\} \\ \boldsymbol{g}_X & := & \bot & \text{for } X \notin \{X_0,\dots,X_l\}. \end{array}$$

Obviously, we have $\boldsymbol{g}(\boldsymbol{w}) \sqsubseteq \boldsymbol{F}_{\sigma\tau}(\boldsymbol{w})$ for all $\boldsymbol{w} \in S^{\mathcal{X}}$ with $\boldsymbol{v} \sqsubseteq \boldsymbol{w}$, and so, as $\boldsymbol{v} \sqsubseteq \boldsymbol{a}$:

$$\boldsymbol{g}(\boldsymbol{a}) \sqsubseteq \boldsymbol{F}_{\sigma\tau}(\boldsymbol{a}) = \boldsymbol{F}_\sigma(\boldsymbol{a}) \sqsubseteq \boldsymbol{F}(\boldsymbol{a}) = \boldsymbol{a}.$$

This means that

$$\begin{array}{rcl} \boldsymbol{a}_{X_0} & \sqsupseteq & D_{X_1} m_0|_{\boldsymbol{v}} \cdot \boldsymbol{a}_{X_1} \\ \boldsymbol{a}_{X_1} & \sqsupseteq & D_{X_2} m_1|_{\boldsymbol{v}} \cdot \boldsymbol{a}_{X_2} \\ & \vdots & \\ \boldsymbol{a}_{X_l} & \sqsupseteq & D_{X_0} m_l|_{\boldsymbol{v}} \cdot \boldsymbol{a}_{X_0}. \end{array}$$

In particular, we have

$$\boldsymbol{a}_{X_0} \sqsupseteq \boldsymbol{a}_{X_0} \cdot \prod_{i=0}^{l} D_{X_{i+1}} m_i|_{\boldsymbol{v}}.$$

Recall that by condition (2)

$$\bot \sqsubset \boldsymbol{v}_{X_0} \text{ and } 1 \sqsubset \prod_{i=0}^{l} D_{X_{i+1}} m_i|_{\boldsymbol{v}}$$

along any cycle of $\mathcal{G}_{\boldsymbol{F}_\sigma}$. By condition (1) we also have $\boldsymbol{v} \sqsubseteq \boldsymbol{a}$, so that $\bot \sqsubset \boldsymbol{a}_{X_0}$ follows. As we require that multiplication preserves strict inequations except when multiplying with $\bot$ or $\top$, it follows that $\boldsymbol{a}_{X_0} = \top$ has to hold. Hence, $\boldsymbol{a}_X = \top$ has to hold for any variable X which is located in some cycle of $\mathcal{G}_{\boldsymbol{F}_{\sigma\tau}}$.

Let $\boldsymbol{G}$ now denote the following min-max-system:

$$\boldsymbol{G}_X := \begin{cases} \top & \text{if } X \text{ is located in some cycle of } \mathcal{G}_{\boldsymbol{F}_{\sigma\tau}} \\ (\boldsymbol{F}_{\sigma\tau})_X & \text{else.} \end{cases}$$

We have $\boldsymbol{F}_{\sigma\tau} \sqsubseteq \boldsymbol{G}$ on $S^{\mathcal{X}}$. Further, the dependency graph of $\boldsymbol{G}$ is acyclic by construction. Hence, $\boldsymbol{G}$ has a unique fixed point which basically can be calculated by constant propagation, i.e., $\boldsymbol{G}^n(\boldsymbol{w})$ is this unique fixed point for any $\boldsymbol{w} \in S^{\mathcal{X}}$. In particular, $\boldsymbol{G}^n(\boldsymbol{a}) = \boldsymbol{G}^n(\top)$. See Lemma 5.3.7.

Note that both $\boldsymbol{F}^k_{\sigma\tau}(\top) = \boldsymbol{G}^k(\top)$ and $\boldsymbol{F}^k_{\sigma\tau}(\boldsymbol{a}) = \boldsymbol{G}^k(\boldsymbol{a})$ hold. For this to see we only have to consider variables X which are located in some cycle, as $\boldsymbol{G}$ and $\boldsymbol{F}_{\sigma\tau}$ coincide on the remaining variables. For every such X we find a successor (w.r.t. $\mathcal{G}_{\boldsymbol{F}_{\sigma\tau}}$) also located on some cycle. By induction it then follows that this successor has value $\top$, so as both $\boldsymbol{G}$ and $\boldsymbol{F}_{\sigma\tau}$ are pure max-systems, also X has value $\top$.

By monotonicity, we therefore have $\boldsymbol{F}^n_\sigma(\top) \sqsubseteq \boldsymbol{F}^n_{\sigma\tau}(\top)$ on the one hand, and on the other hand we have $\boldsymbol{F}^n_{\sigma\tau}(\boldsymbol{a}) \sqsubseteq \boldsymbol{a}$ as $\boldsymbol{F}_{\sigma\tau}(\boldsymbol{a}) \sqsubseteq \boldsymbol{a}$ by choice of τ. With $\boldsymbol{G}^n(\top) = \boldsymbol{G}^n(\boldsymbol{a})$ we conclude that $\boldsymbol{F}^n_\sigma(\top) \sqsubseteq \boldsymbol{a}$ for any fixed point $\boldsymbol{a}$ of $\boldsymbol{F}$. □

Motivated by the preceding result on reasonable strategies we define the set of improvements of a given max-strategy σ as follows:

Definition 5.3.13.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a min-max-system, and σ some max-strategy. Set $\boldsymbol{v} := \boldsymbol{F}^{|\mathcal{X}|}_\sigma(\top)$.

Then the set $S_\sigma(X)$ of *strict improvements*, resp. the set $I_\sigma(X)$ of *improvements* of σ at $X \in \mathcal{X}_\sqcup$ w.r.t. $\boldsymbol{F}$ are defined by

$$\begin{aligned} S_\sigma(X) &:= \{m \mid m \text{ is a monomial of } \boldsymbol{F}_X \text{ with } \boldsymbol{v}_X \sqsubset m(\boldsymbol{v})\} \\ I_\sigma(X) &:= S_\sigma(X) \cup \{m \in \sigma(X) \mid \boldsymbol{v}_X = m(\boldsymbol{v})\}. \end{aligned}$$

We call any max-strategy σ' with $\sigma'(X) \subseteq I_\sigma(X)$ (for all $X \in \mathcal{X}_\sqcup$) a *successor strategy* of σ. ◇

Remark 5.3.14.
I_σ is itself a successor strategy of σ. ◇

We close this subsection by showing that for any nondeterministic reasonable strategy σ there is a deterministic reasonable strategy ρ with $\nu\boldsymbol{F}_\sigma = \nu\boldsymbol{F}_\rho$. We will use this result in the next subsection to get a better bound on the number of strategy iterations needed to reach $\mu\boldsymbol{F}$.

Definition 5.3.15.
Let $\boldsymbol{F}$ be a min-max-system $\boldsymbol{F}$, and σ a reasonable non-deterministic max-strategy. We call any deterministic max-strategy ρ satisfying

$$\boldsymbol{F}_\rho(\nu\boldsymbol{F}_\sigma) = \nu\boldsymbol{F}_\sigma$$

a *determinization* of $\boldsymbol{F}$. ◇

Lemma 5.3.16.
The determinization ρ of σ is also reasonable with $\nu\boldsymbol{F}_\rho = \nu\boldsymbol{F}_\sigma$. ◇

Proof. Note that $\boldsymbol{F}_\rho \sqsubseteq \boldsymbol{F}_\sigma$. By construction $\nu\boldsymbol{F}_\sigma$ is a fixed point of $\boldsymbol{F}_\rho$, and thus also its greatest fixed point, i.e., $\nu\boldsymbol{F}_\sigma = \nu\boldsymbol{F}_\rho$. We claim that $\nu\boldsymbol{F}_\sigma$ is a witness of ρ being reasonable:

As σ is reasonable, we have $\nu\boldsymbol{F}_\sigma \sqsubseteq \boldsymbol{a}$ for any fixed point of $\boldsymbol{F}$ (Lemma 5.3.12). So, $\nu\boldsymbol{F}_\sigma$ satisfies condition (1). Condition (3) holds because $\nu\boldsymbol{F}_\sigma$ is a fixed point of $\boldsymbol{F}_\rho$. Finally, condition (2) is satisfied as $\mathcal{G}_{\boldsymbol{F}_\rho}$ is a subgraph of $\mathcal{G}_{\boldsymbol{F}_\sigma}$ and $\boldsymbol{v} \sqsubseteq \nu\boldsymbol{F}_\sigma$ with $\boldsymbol{v}$ the witness of σ being reasonable. □

5.3.3 Nondeterministic ⊔-Strategy Iteration

We now can give a formal definition of strategy iteration using nondeterministic ⊔-strategies.

Definition 5.3.17.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a min-max-system with $n := |\mathcal{X}|$.

A *strategy iteration sequence* is any maximal sequence $(\sigma_i)_{i\in I}$ (with either $I = \{0, 1, \ldots, L\}$ for some $L \in \mathbb{N}$ or $I = \mathbb{N}$) of max-strategies satisfying for all $i \in I$:

(1) $\sigma_0(X) := \{\boldsymbol{F}_X(\bot)\}$ for every max-variable X ([5]).

(2) For $\boldsymbol{v}_{i+1} := \boldsymbol{F}^n_{\sigma_i}(\top)$ the ⊔-strategy σ_{i+1} satisfies:

  (a) $\boldsymbol{v}_{i+1} \sqsubset \boldsymbol{F}_{\sigma_{i+1}}(\boldsymbol{v}_{i+1}) \sqsubseteq \boldsymbol{F}(\boldsymbol{v}_{i+1})$

  (b) σ_{i+1} is a successor strategy of σ_i. ◇

The goal of the remainder of this subsection is to show the following result:

Theorem 5.3.19.
Let $\boldsymbol{F}$ be a min-cycle-free min-max-system $\boldsymbol{F}$ and $(\sigma_i)_{i\in I}$ a strategy iteration sequence. Then $(\sigma_i)_{i\in I}$ is finite and $\nu\boldsymbol{F}_{\sigma_L}$ is the least fixed point of $\boldsymbol{F}$ (with $L = \max I$). Further, let $N^{\boldsymbol{F}}_{\sqcup}$ be the number of distinct deterministic max-strategies w.r.t. $\boldsymbol{F}$, then $L \leq N^{\boldsymbol{F}}_{\sqcup}$. ◇

In order to prove this we show that every strategy of a strategy iteration sequence is reasonable, if the given min-max-system is min-cycle-free:

[5] I.e., σ_0 maps every max-variable to the constant term of $\boldsymbol{F}_X$.

Lemma 5.3.18.
Let $\boldsymbol{F}$ be a min-cycle-free min-max-system with variables $\mathcal{X}$ ($n := |\mathcal{X}|$). Further, assume that $(\sigma_i)_{i\in I}$ is a strategy iteration sequence (with either $I = \{0, 1, \dots, L\}$ for some $L \in \mathbb{N}$ or $I = \mathbb{N}$).

Set $\boldsymbol{v}_0 := \bot$, resp. $\boldsymbol{v}_{i+1} := \boldsymbol{F}^n_{\sigma_i}(\top)$ for $i \geq 0$. Then σ_i is reasonable with witness $\boldsymbol{v}_i$. Further $\boldsymbol{v}_i \sqsubset \boldsymbol{v}_{i+1}$. ◇

Proof. We proceed by induction on i.

($i = 0$) By definition σ_0 maps every max-variable X to the constant term of $\boldsymbol{F}_X$. Every max-variable $X \in \mathcal{X}_\sqcup$ therefore has no outgoing edge in $\mathcal{G}_{\boldsymbol{F}_{\sigma_0}}$. By our assumption that there are no cycles which only consist of min-variable, the dependency graph of $\boldsymbol{F}_{\sigma_0}$ is acyclic. So condition (2) of Definition 5.3.11 is trivially satisfied. Obviously, $\boldsymbol{v}_0$ also satisfies conditions (1) and (3) as $\boldsymbol{v}_0 = \bot$ by definition. So, σ_0 is reasonable with witness $\boldsymbol{v}_0$. As $\boldsymbol{F}_{\sigma_0}$ is acyclic, it has a unique fixed point given both by $\boldsymbol{v}_1 = \boldsymbol{F}^n_{\sigma_0}(\top)$ and $\boldsymbol{F}^n_{\sigma_0}(\bot)$. If $\bot = \boldsymbol{F}^n_{\sigma_0}(\top)$, then $\bot$ is also the least fixed point of $\boldsymbol{F}$, and $I = \{0\}$. Otherwise, we have

$$\boldsymbol{v}_0 = \bot \sqsubset \boldsymbol{F}^n_{\sigma_0}(\bot) = \boldsymbol{v}_1.$$

($i \to i{+}1$) Assume that σ_i is reasonable with witness $\boldsymbol{v}_i$ and that σ_{i+1} exists, i.e., $i{+}1 \in I$. We show that σ_{i+1} is reasonable with witness $\boldsymbol{v}_{i+1} = \boldsymbol{F}^n_{\sigma_i}(\top)$.

As σ_i is reasonable, we know that $\boldsymbol{v}_{i+1} := \boldsymbol{F}^n_{\sigma_i}(\top)$ is the greatest fixed point of $\boldsymbol{F}_{\sigma_i}$, i.e. $\boldsymbol{v}_{i+1} = \nu\boldsymbol{F}_{\sigma_i}$ and $\boldsymbol{v}_{i+1} \sqsubseteq \boldsymbol{a}$ for any fixed point $\boldsymbol{a}$ of $\boldsymbol{F}$. Hence $\boldsymbol{v}_{i+1}$ satisfies condition (1) of Definition 5.3.11. Further, $\boldsymbol{v}_{i+1} = \boldsymbol{F}_{\sigma_i}(\boldsymbol{v}_{i+1}) \sqsubseteq \boldsymbol{F}(\boldsymbol{v}_{i+1})$. By Definition 5.3.17 (2a) σ_{i+1} is chosen such that

$$\boldsymbol{v}_{i+1} \sqsubset \boldsymbol{F}_{\sigma_{i+1}}(\boldsymbol{v}_{i+1}) \sqsubseteq \boldsymbol{F}(\boldsymbol{v}_{i+1}).$$

From this and the monotonicity of $\boldsymbol{F}_\sigma$, we immediately obtain

$$\boldsymbol{v}_{i+1} \sqsubset \boldsymbol{F}^k_{\sigma_{i+1}}(\boldsymbol{v}_{i+1}) \sqsubseteq \boldsymbol{F}^k_{\sigma_{i+1}}(\top) \text{ for all } k \in \mathbb{N},$$

i.e., $\boldsymbol{v}_{i+1}$ also satisfies condition (3) of Definition 5.3.11. Further we obtain $\boldsymbol{v}_{i+1} \sqsubset \boldsymbol{F}^n_{\sigma_{i+1}}(\top) = \boldsymbol{v}_{i+2}$.

If $\mathcal{G}_{\boldsymbol{F}_{\sigma_{i+1}}}$ is acyclic, we are done. Hence, let $X_0 \xrightarrow{m_0} \dots X_l \xrightarrow{m_l} X_{l+1}$ be a cycle in $\mathcal{G}_{\boldsymbol{F}_{\sigma_{i+1}}}$.

If this cycle already exists w.r.t. the preceding strategy σ_i, then we know by induction that

$$\bot \sqsubset (\boldsymbol{v}_i)_{X_0} \text{ and } 1 \sqsubset \prod_{i=0}^{l} D_{X_{i+1}} m_i|_{\boldsymbol{v}_i}.$$

As $\boldsymbol{v}_i \sqsubseteq \boldsymbol{v}_{i+1}$, we have by monotonicity that

$$\bot \sqsubset (\boldsymbol{v}_{i+1})_{X_0} \text{ and } 1 \sqsubset \prod_{i=0}^{l} D_{X_{i+1}} m_i|_{\boldsymbol{v}_{i+1}}.$$

also hold, i.e., the cycle satisfies condition (2) of Definition 5.3.11.

We turn to the case that the cycle only exists w.r.t. σ_{i+1}, i.e., there is some max-variable along the cycle, w.l.o.g. X_0, such that the monomial $m_0 \in \sigma_{i+1}(X_0) \setminus \sigma_i(X_0)$ was newly added to σ_{i+1}. By Definition 5.3.17 (2b) we therefore know that $(\boldsymbol{v}_{i+1})_{X_0} \sqsubset m_0(\boldsymbol{v}_{i+1})$. Consider any other variable X_j along the cycle. If $X_j \in \mathcal{X}_\sqcap$, we have $\boldsymbol{F}_{\sigma_{i+1}}(\boldsymbol{v})_{X_j} \sqsubseteq m_j(\boldsymbol{v})$ for any $\boldsymbol{v} \in S^{\mathcal{X}}$, as m_i is a monomial of $(\boldsymbol{F}_{\sigma_{i+1}})_{X_j}$. And so

$$(\boldsymbol{v}_{i+1})_{X_j} \sqsubseteq m_j(\boldsymbol{v}_{i+1})$$

as $\boldsymbol{v}_{i+1} \sqsubseteq \boldsymbol{F}(\boldsymbol{v}_{i+1})$ and $\boldsymbol{F}_{\sigma_{i+1}}$ and $\boldsymbol{F}$ coincide on min-variables.

If $X_j \in \mathcal{X}_\sqcup$, then we have by Definition 5.3.17 (2b) that all monomials of σ_{i+1} do not decrease in $\boldsymbol{v}_{i+1}$, i.e.,

$$(\boldsymbol{v}_{i+1})_{X_j} \sqsubseteq m_j(\boldsymbol{v}_{i+1})$$

holds too.

So, we obtain:

$$\begin{array}{lcll} (\boldsymbol{v}_{i+1})_{X_0} & \sqsubset & m_0(\boldsymbol{v}_{i+1}) & = D_{X_1} m_0|_{\boldsymbol{v}_{i+1}} \cdot (\boldsymbol{v}_{i+1})_{X_1} \\ (\boldsymbol{v}_{i+1})_{X_1} & \sqsubseteq & m_1(\boldsymbol{v}_{i+1}) & = D_{X_2} m_1|_{\boldsymbol{v}_{i+1}} \cdot (\boldsymbol{v}_{i+1})_{X_2} \\ & \vdots & & \\ (\boldsymbol{v}_{i+1})_{X_l} & \sqsubseteq & m_l(\boldsymbol{v}_{i+1}) & = D_{X_0} m_l|_{\boldsymbol{v}_{i+1}} \cdot (\boldsymbol{v}_{i+1})_{X_0}. \end{array}$$

From this we conclude:

$$(\boldsymbol{v}_{i+1})_{X_0} \sqsubset (\boldsymbol{v}_{i+1})_{X_0} \prod_{j=0}^{l} D_{X_{j+1}} m_j|_{\boldsymbol{v}_{i+1}}.$$

By Proposition 5.3.3(6) it follows that

$$\bot \sqsubset (\boldsymbol{v}_{i+1})_{X_0} \text{ and } 1 \sqsubset \prod_{j=0}^{l} D_{X_{j+1}} m_j|_{\boldsymbol{v}_{i+1}}.$$

From this also $\bot \sqsubset (\boldsymbol{v}_{i+1})_{X_j}$ for all $j = 0, \ldots, l$ follows. Thus, any newly generated cycle also satisfies condition (2) of Definition 5.3.11 w.r.t. $\boldsymbol{v}_{i+1}$. □

We are now ready to prove our main result.

Theorem 5.3.19.
Let $\boldsymbol{F}$ be a min-cycle-free min-max-system $\boldsymbol{F}$ and $(\sigma_i)_{i \in I}$ a strategy iteration sequence. Then $(\sigma_i)_{i \in I}$ is finite and $\nu \boldsymbol{F}_{\sigma_L}$ is the least fixed point of $\boldsymbol{F}$ (with $L = \max I$). Further, let $N_\sqcup^{\boldsymbol{F}}$ be the number of distinct deterministic max-strategies w.r.t. $\boldsymbol{F}$, then $L \leq N_\sqcup^{\boldsymbol{F}}$. ◇

Proof. By the preceding Lemma 5.3.18 we have

$$\nu \boldsymbol{F}_{\sigma_i} \sqsubset \nu \boldsymbol{F}_{\sigma_{i+1}} \sqsubseteq \mu \boldsymbol{F}$$

for all $i, i+1 \in I$. As for every strategy σ_i there is a determinization having the same greatest fixed point, the number of disctinct values $\nu \boldsymbol{F}_{\sigma_i}$ is bounded by the number of

reasonable deterministic max-strategies (w.r.t. $\boldsymbol{F}$). So, $N_{\sqcup}^{\boldsymbol{F}}$ is an upper bound on the length of any strategy iteration sequence, i.e., for every strategy iteration sequence $(\sigma_i)_{i\in I}$ we have $I = \{0, 1, \ldots, L\}$ with $L \leq N_{\sqcup}^{\boldsymbol{F}}$. As strategy iteration sequences are required to be maximal, we have $\nu\boldsymbol{F}_{\sigma_L} = \boldsymbol{F}(\nu\boldsymbol{F}_{\sigma_L})$ as otherwise we could find a successor strategy ρ of σ_L satisfying

$$\nu\boldsymbol{F}_{\sigma_L} \sqsubset \boldsymbol{F}_\rho(\nu\boldsymbol{F}_{\sigma_L}).$$

So, $\nu\boldsymbol{F}_{\sigma_L}$ is a fixed point of $\boldsymbol{F}$, in particular, the least. □

Remark 5.3.20.
For a min-max-system $\boldsymbol{F}$ set $k_{\boldsymbol{F}}^{\sqcup} := \sum_{X\in\mathcal{X}_\sqcup} k_{\boldsymbol{F}}(X)$, i.e., $k_{\boldsymbol{F}}^{\sqcup}$ is the total number of monomials appearing in all $\boldsymbol{F}_X$ with $X \in \mathcal{X}_\sqcup$. As any $\sqcup$-strategy defines some subset of these monomials we have $N_{\boldsymbol{F}}^{\sqcup} \leq 2^{k_{\boldsymbol{F}}^{\sqcup}}$. ◇

Example 5.3.21. In Example 5.2.9 we have seen that $\sqcap$-strategy iteration does not work in general. There we considered the following min-max-system:

$$\boldsymbol{F} = (\boldsymbol{F}_X, \boldsymbol{F}_Y) = \big(\min\{Y, 1\}, \max\{X + X, -1\}\big).$$

By Defintion 5.3.17 we begin with the strategy $\sigma_0 : \{Y\} \to \{-1\}$. So, the induced min-max-system is:

$$\boldsymbol{F}_{\sigma_0} = \big(\min\{Y, 1\}, -1\big).$$

The greatest fixed point of which can easily be calculated to be $(-1, -1)$ which also is already the least fixed point of $\boldsymbol{F}$. ◇

5.3.4 Locally Optimal Successor Strategies

In the previous subsection we introduced $\sqcup$-strategy iteration and showed that it always converges to the least fixed point of a min-max-system. Up to now, we have not commented on the choice of the σ_i in every step of the strategy iteration. Obviously, the successor strategy σ_{i+1} is not uniquely determined in general. Still, there is a unique maximal, or most permissive choice given by I_{σ_i}. The interesting point is that as for any possible choice of σ_{i+1} we have $\sigma_{i+1}(X) \subseteq I_{\sigma_i}(X)$, and, thus,

$$\boldsymbol{F}_{\sigma_{i+1}} \sqsubseteq \boldsymbol{F}_{I_{\sigma_i}}, \text{ and } \nu\boldsymbol{F}_{\sigma_{i+1}} \sqsubseteq \nu\boldsymbol{F}_{I_{\sigma_i}}.$$

This means that locally, i.e., when only considering σ_i, there is an optimal choice for σ_{i+1}, namely $\sigma_{i+1} := I_{\sigma_i}$.

Definition 5.3.22.
For a given min-max-system $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ the *locally optimal strategy iteration sequence* is the unique strategy iteration sequence with $\sigma_{i+1} := I_{\sigma_i}$. ◇

Although choosing σ_{i+1} to be I_{σ_i} in every step does not necessarily lead to a minimal number of iterations until $\mu\boldsymbol{F}$ is reached, we will show that it leads

to a better bound on the number of iterations at least in the important case of linear min-max-systems. This is done in the following section. Before it we show that we can always assume that the least fixed point of a min-max-system is greater than $\bot$ in any component, and we discuss how to obtain from any min-max-system a min-cycle-free system.

5.3.5 Clean and Min-Cycle-Free Min-Max-Systems

We discuss two consequences of Theorem 5.3.19. We first show that we may assume w.l.o.g. that any min-max-system is min-cycle-free.

Lemma 5.3.23.
Let $\boldsymbol{F}$ be some min-max-system with variables $\mathcal{X}$. By C we denote the variables X of $\boldsymbol{F}$ which are located in some min-cycle of $\mathcal{G}_{\boldsymbol{F}}$, i.e., a cycle consisting only of min-variables.

Then $(\mu\boldsymbol{F})_X = \bot$ for all $X \in C$. $\diamond$

Proof. Assume that $C \neq \emptyset$ and denote by $\hat{C}$ the set $\{\hat{X} \mid X \in C\}$. We assume that $\hat{C} \cap \mathcal{X} = \emptyset$. Let $\boldsymbol{G}$ be the min-max-system we obtain from $\boldsymbol{F}$ as follows: if $X \in \mathcal{X}$, then $\boldsymbol{G}_X$ is obtained from $\boldsymbol{F}_X$ by replacing any variable $X \in C$ by $\hat{X} \in \hat{C}$; otherwise we set $\boldsymbol{G}_{\hat{X}} = X \sqcup \bot$. Then $\boldsymbol{G}$ is min-cycle-free and $\mu\boldsymbol{G}$ exists. For any fixed point $\boldsymbol{a}$ of $\boldsymbol{G}$ we have $\boldsymbol{a}_X = \boldsymbol{a}_{\hat{X}}$ and, thus, every fixed point of $\boldsymbol{G}$ induces a fixed point of $\boldsymbol{F}$. Similarly, we can lift any fixed point $\boldsymbol{b}$ of $\boldsymbol{F}$ to a fixed point of $\boldsymbol{G}$ via the extension $\boldsymbol{b}_{\hat{X}} := \boldsymbol{b}_X$. Further, for any two fixed points $\boldsymbol{a}$ and $\boldsymbol{a}'$ with $\boldsymbol{a} \sqsubseteq \boldsymbol{a}'$ of the one system this inequations also holds for the induced fixed points of the other system, i.e., the partial order on the fixed points is the same. One therefore verifies that $\mu\boldsymbol{G}$ induces $\mu\boldsymbol{F}$ by means of this one-on-one correspondence.

It therefore suffices to show that for any strategy iteration sequence $(\sigma_i)_{i=0,\dots,L}$ we have $\sigma_i(\hat{X}) = \{\bot\}$ for all $\hat{X} \in \hat{C}$. As $\mu\boldsymbol{G} = \nu\boldsymbol{G}_{\sigma_L}$ the result then follows. We proceed by induction on i:

$(i = 0)$ By definition of strategy iteration, $\sigma_0(\hat{X}) = \bot$ for all $\hat{X} \in \hat{C}$.

$(i \to i+1)$ Recall that for any monomial $m \in \sigma_{i+1}(X) \setminus \sigma_i(X)$ we have by definition

$$(\nu\boldsymbol{G}_{\sigma_i})_X \sqsubset m(\nu\boldsymbol{G}_{\sigma_i}).$$

Consider now any variable $\hat{X} \in \hat{C}$. By definition of C, there is some min-variable Y such that $X \to Y$ in $\mathcal{G}_{\boldsymbol{F}}$. By induction, we have $\sigma_i(\hat{Y}) = \{\bot\}$ and thus $(\nu\boldsymbol{G}_{\sigma_i})_{\hat{Y}} = \bot$. As $\hat{Y}$ appears in $\boldsymbol{G}_X$ and $X \in \mathcal{X}_{\sqcap}$, we also have

$$(\nu\boldsymbol{G}_{\sigma_i})_X = \boldsymbol{G}_X(\nu\boldsymbol{G}_{\sigma_i}) = \bot.$$

Hence, we may not include X into $\sigma_{i+1}(\hat{X})$, i.e., we also have $\sigma_{i+1}(\hat{X}) = \{\bot\}$ and thus $(\nu\boldsymbol{G}_{\sigma_{i+1}})_{\hat{X}}$ for any $\hat{X} \in \hat{C}$. $\square$

Remark 5.3.24.
We therefore may obtain from any min-max-system a min-cycle-free system, by substituting $\bot$ for every variable located in some min-cycle. $\diamond$

Similar to polynomial systems, we say that a min-max-system is *clean* if $\mu\boldsymbol{F}$ is greater than $\bot$ in any component. We next show that we can identify those variables X with $(\mu\boldsymbol{F})_X = \bot$ by means of n steps of strategy iteration.

Lemma 5.3.25.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be min-cycle-free min-max-system with $n = |\mathcal{X}|$ and $(\sigma_i)_{i=0,\dots,L}$ a $\sqcup$-strategy iteration sequence. Set $\boldsymbol{v}_{i+1} := \nu\boldsymbol{F}_{\sigma_i}$.

For every $X \in \mathcal{X}$ we have $\left(\mu\boldsymbol{F}\right)_X = \bot$ if $\left(\boldsymbol{v}_n\right)_X = \bot$. $\diamond$

Proof. We proceed by induction on $n = |\mathcal{X}|$.

($n = 1$) We have a single variable X. As $\boldsymbol{F}$ is min-cycle-free, X is a max-variable. By definition, σ_0 then chooses the constant term of $\boldsymbol{F}$ and the claim is obviously true, as $\boldsymbol{v}_1 = \nu\boldsymbol{F}_{\sigma_0}$.

($n \to n+1$) Consider any $X \in \mathcal{X}_\sqcup$ with $\left(\mu\boldsymbol{F}\right)_X \sqsupset \bot$. As $\boldsymbol{v}_{L+1} = \mu\boldsymbol{F}$, there exists some index l such that $\left(\boldsymbol{v}_l\right)_X = \bot \sqsubset \left(\boldsymbol{v}_{l+1}\right)_X$. By monotonicity of the approximations $\boldsymbol{v}_i$, we have $\left(\boldsymbol{v}_i\right)_X = \bot$ for all $i \in \{0, 1, \dots, l\}$. By Definition 5.3.17(2b) it follows that $\sigma_i(X) = \{\bot\}$ for $i \in \{0, \dots, l-1\}$ ([6]). So there is a monomial $m \in \sigma_l(X) \cap S_{\sigma_{l-1}}(X)$ with $\bot = \left(\boldsymbol{v}_l\right)_X \sqsubset m(\boldsymbol{v}_l)$. Let V denote the set of variables appearing in m. As $\bot \sqsubset m(\boldsymbol{v}_l)$ we have $\bot \sqsubset \left(\boldsymbol{v}_l\right)_Y$ for all $Y \in V$. In particular, $X \notin V$. Now, as $\sigma_i(X) = \{\bot\}$ for all $i \in \{0, 1, \dots, l-1\}$ we are basically considering a system in at most $n = |\mathcal{X} \setminus \{X\}|$ variables for the first l iterations. Thus, by induction $\bot \sqsubset \left(\boldsymbol{v}_n\right)_Y$ for all $Y \in V$, and, hence, $l \leq n$.

We turn to the case of min-variables. As $\boldsymbol{F}$ is assumed to be min-cycle-free, every min-variable X ultimately depends only on max-variables, i.e., every path in $\mathcal{G}_{\boldsymbol{F}}$ starting in X eventually hits a max-variable. Let R_X be the set of max-variables reachable from X in $\mathcal{G}_{\boldsymbol{F}}$ when deleting all out-going edges of max-variables.

We define a ranking function $r : \mathcal{X} \to \mathbb{N}$ with $r(X) = 0$ if X is a max-variable; and $r(X) = 1 + \max\{r(Z) \mid Z \text{ appears in } \boldsymbol{F}_X\}$, otherwise. This function is well-defined, as $\boldsymbol{F}$ is min-cycle-free.

As $\boldsymbol{F}$ and $\boldsymbol{F}_{\sigma_{i-1}}$ coincide on min-variables, and $\boldsymbol{v}_i$ is a fixed point of $\boldsymbol{F}_{\sigma_{i-1}}$, one now easily checks by induction on the rank of a min-variable ([7]) that for all $i = 1, \dots, L+1$ we have

[6] By definition σ_0 chooses the constant term. Hence, if $(\boldsymbol{v}_1)_X = \bot$ for some $X \in \mathcal{X}_\sqcup$, then the constant of $\boldsymbol{F}_X$ has to be $\bot$. Further, we only include a new monomial m in $\sigma_{i+1}(X)$ if $(\boldsymbol{v}_{i+1})_X \sqsubset m(\boldsymbol{v}_{i+1})$. From this $\bot \sqsubset \boldsymbol{F}^k_{\sigma_{i+1}}(\boldsymbol{v}_{i+1})_X$ follows for all $k > 0$, which in turn implies that $\bot \sqsubset (\nu\boldsymbol{F}_{\sigma_{i+1}})_X = (\boldsymbol{v}_{i+2})_X$.

[7] We assume that there are no "pathological" equations in $\boldsymbol{F}_X$, i.e., if $\boldsymbol{F}_X(\boldsymbol{v}) = \bot$ for all $\boldsymbol{v} \in S^{\mathcal{X}}$, then $X \in \mathcal{X}_\sqcup$ and $\boldsymbol{F}_X = \bot$.

that $(\boldsymbol{v}_i)_X = \bot$ iff $(\boldsymbol{v}_i)_Y = \bot$ for some $Y \in R_X$. Because we already know that $(\boldsymbol{v}_n)_Y = \bot$ iff $(\mu\boldsymbol{F})_Y = \bot$ for any max-variable Y, we may extend this result to all variables. □

By the preceding results we can always assume that a min-max-system is min-cycle-free and clean. By virtue of these results we introduce a normal form for min-max-systems.

Definition 5.3.26.
We say that a min-max-system $\boldsymbol{F}$ is in *normal form* if it is min-cylce-free, clean and for any max-variable X we have $\bot \sqsubset \boldsymbol{F}_X(\bot)$, i.e., the constant term of $\boldsymbol{F}_X$ is not equal to $\bot$. ◇

Lemma 5.3.27.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a min-cycle-free min-max-system, and $(\sigma_i)_{i\in\{0,\dots,L\}}$ a strategy iteration sequence w.r.t. $\boldsymbol{F}$. Denote by B the set of variables X with $\bot = \big(\boldsymbol{v}_n\big)_X$. Let $\boldsymbol{F}[\bot/B]$ denote the system we obtain from $\boldsymbol{F}$ by substituting every variable in B by $\bot$. Set $\mathcal{X}' := \mathcal{X} \setminus B$.

Define the system $\boldsymbol{G} : S^{\mathcal{X}'} \to S^{\mathcal{X}'}$ by

$$\begin{array}{lcll} \boldsymbol{G}_X & := & \boldsymbol{F}[\bot/B]_X & \text{for } X \in \mathcal{X}'_{\sqcap} \\ \boldsymbol{G}_X & := & \boldsymbol{F}[\bot/B]_X \sqcup \big(\boldsymbol{v}_n\big)_X & \text{for } X \in \mathcal{X}'_{\sqcup}. \end{array}$$

Then $\boldsymbol{G}$ is in normal form with $(\mu\boldsymbol{F})_X = (\mu\boldsymbol{G})_X$ for $X \in \mathcal{X}'$. ◇

Proof. Obviously, $\boldsymbol{G}$ is also min-cycle-free as $\mathcal{G}_{\boldsymbol{G}}$ is a subgraph of $\mathcal{G}_{\boldsymbol{F}}$.

Let $\boldsymbol{a}$ be the restriction of $\mu\boldsymbol{F}$ to $\mathcal{X}'$. We then have for all $X \in \mathcal{X}'_{\sqcup}$:

$$\boldsymbol{G}_X(\boldsymbol{a}) = \boldsymbol{F}_X[\bot/B](\boldsymbol{a}) \sqcup \big(\boldsymbol{v}_n\big)_X = \boldsymbol{F}_X(\mu\boldsymbol{F}) \sqcup \big(\boldsymbol{v}_n\big)_X = (\mu\boldsymbol{F})_X \sqcup \big(\boldsymbol{v}_n\big)_X = (\mu\boldsymbol{F})_X = \boldsymbol{a}_X.$$

Similarly, $\boldsymbol{G}_X(\boldsymbol{a}) = \boldsymbol{a}_X$ for $X \in \mathcal{X}'_{\sqcap}$ can be shown. So, $\boldsymbol{a}$ is a fixed point of $\boldsymbol{G}$, and $\mu\boldsymbol{G} \sqsubseteq \boldsymbol{a}$ follows.

Let $\boldsymbol{w}$ be the restriction of $\boldsymbol{v}_n$ to $\mathcal{X}'$. We claim that $\boldsymbol{w} \sqsubseteq \mu\boldsymbol{G}$. For this let σ be the max-strategy which chooses the constant term of $\boldsymbol{G}$, i.e., σ is the initial strategy considered in any strategy iteration sequence w.r.t. $\boldsymbol{G}$. Then $\nu\boldsymbol{G}_\sigma \sqsubseteq \mu\boldsymbol{G}$.

Let $r : \mathcal{X}' \to \mathbb{N}$ be the ranking function defined by $r(X) = 0$ if $X \in \mathcal{X}'_{\sqcup}$; and $r(X) = 1 + \max\{r(Z) \mid Z \text{ appears in } \boldsymbol{G}_X\}$ otherwise.

We show by induction on the rank of a variable that $\boldsymbol{w} \sqsubseteq \nu\boldsymbol{G}_\sigma$:

If $r(X) = 0$, then $X \in \mathcal{X}_{\sqcup}$ and the claim is obviously true. Otherwise X is a min-variable, and we know by induction for all variables Z appearing in $\boldsymbol{G}_X$ that $\boldsymbol{w}_Z \sqsubseteq (\mu\boldsymbol{G})_Z$. So:

$$(\nu\boldsymbol{G}_\sigma)_X = \boldsymbol{G}_\sigma(\nu\boldsymbol{G}_\sigma)_X = \boldsymbol{G}_X(\nu\boldsymbol{G}_\sigma) \sqsupseteq \boldsymbol{G}_X(\boldsymbol{w}) = \boldsymbol{F}_X[\bot/B](\boldsymbol{w}) = \boldsymbol{F}_X(\boldsymbol{v}_n) \sqsupseteq (\boldsymbol{v}_n)_X.$$

It now also follows that for $X \in \mathcal{X}_{\sqcup}$ we have

$$\boldsymbol{F}_X[\bot/B](\mu\boldsymbol{G}) \sqsupseteq \boldsymbol{F}_X[\bot/B](\boldsymbol{w}) = \boldsymbol{F}_X(\boldsymbol{v}_n) \sqsupseteq (\boldsymbol{v}_n)_X,$$

and thus also

$$(\mu\boldsymbol{G})_X = \boldsymbol{G}_X(\mu\boldsymbol{G}) = \boldsymbol{F}_X[\bot/B](\mu\boldsymbol{G}) \sqcup (\boldsymbol{v}_n)_X = \boldsymbol{F}_X[\bot/B](\mu\boldsymbol{G}). \qquad (*)$$

Let $\boldsymbol{b}$ be the extension $\boldsymbol{b}$ of $\mu\boldsymbol{G}$ to $\mathcal{X}$ by setting $\boldsymbol{b}_X := \bot$ for $X \in B$. By virtue of $(*)$ one now easily checks that $\boldsymbol{b}$ is also a fixed point of $\boldsymbol{F}$, i.e., $\mu\boldsymbol{F} \sqsubseteq \boldsymbol{b}$. So we obtain that $(\mu\boldsymbol{G})_X \sqsubseteq (\mu\boldsymbol{F})_X \sqsubseteq (\mu\boldsymbol{G})_X$ for all $X \in \mathcal{X}'$. □

5.4 Linear Min-Max-Systems and Games

The subject of this section are linear min-max-systems $\boldsymbol{F}$, i.e., min-max-systems where every monomial has at most degree one. We formalize a connection between linear min-max-systems and games played on the associated dependency graph (Subsection 5.4.1). We then introduce *reasonable strategies* w.r.t. to the interpretation of linear min-max systems as games, and obtain an improved bound on the number of steps done by the locally-optimal strategy iteration (Subsection 5.4.2).

For this section we assume that all systems are clean and min-cycle-free.

5.4.1 Interpretation as Games

We first adapt the definition of the dependency graph to the case that all monomials have degree at most one: for every edge $X \xrightarrow{m} Y$ in $\mathcal{G}_{\boldsymbol{F}}$ it holds that $m = D_Y m \cdot Y$ with $D_Y m \sqsupset \bot$. It therefore suffices to label the edges only by $D_Y m$:

Definition 5.4.1.
For $\boldsymbol{F}$ a linear min-max-system in normal form let $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$ denote the directed, edge labeled graph with nodes $\mathcal{X} \cup \{\blacklozenge\}$ (with $\blacklozenge \notin \mathcal{X}$) and edges defined by:

- If $X \xrightarrow{m} Y$ in $\mathcal{G}_{\boldsymbol{F}}$, then $X \xrightarrow{D_Y m} Y$ in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$.
- For all $X \in \mathcal{X}_{\sqcup}$ there is the edge $X \xrightarrow{\boldsymbol{F}_X(\bot)} \blacklozenge$.

The *weight* of a finite path

$$X_0 \xrightarrow{c_0} X_1 \xrightarrow{c_1} \ldots \xrightarrow{c_l} X_{l+1}$$

in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$ is then defined to be the product of the edge labels, i.e., $\prod_{j=0}^{l} c_j$. ◇

On $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$ we can consider the following game played by two players which we simply call player $\sqcup$ (or player max) and player $\sqcap$ (or player min) in the following. This definition is motivated by the *finite cycle-domination games* considered in [VJ00]:

Definition 5.4.2.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a linear min-max-system in normal form. A *play* of the two players $\sqcup$, and $\sqcap$ on the graph $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$ is a path

$$X_0 \xrightarrow{c_0} X_1 \xrightarrow{c_1} \dots X_i \xrightarrow{c_i} X_{i+1} \dots \xrightarrow{c_{l-1}} X_l$$

such that either all nodes X_i are pairwise different with $X_l = \blacklozenge$, i.e., $|\{X_0, \dots, X_l\}| = l+1$, or X_l is the first node visited twice along the path, i.e., $|\{X_0, \dots, X_l\}| = l$ and there is some $j \in \{0, \dots, l-1\}$ such that $X_j = X_l$.

The *value* of such a play depends on whether it hits $\blacklozenge$, or it ends up in a cycle:

- If $X_l = \blacklozenge$, then the value of the play is the value of the path.
- If it ends up in a simple cycle of weight greater than 1, then its value is $\top$.
- Otherwise (that is, if the play winds up in a simple cycle of weight at most 1) its value is $\bot$.

The goal of player $\sqcup$ is now to maximize the value of the play, while player $\sqcap$ tries to minimize the value. ◇

Note that every $\sqcup$-strategy σ induces the subgraph $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\sigma}$, i.e., it can be interpreted as player $\sqcup$ disabling some edges. Similarly, a $\sqcap$-strategy τ corresponds to disabling edges leaving $\sqcap$-nodes. In particular, if a player chooses a deterministic strategy, every play in the resulting game is determined purely by his opponent.

Example 5.4.3. Consider the following min-max-system $\boldsymbol{F}$ on the semiring $\langle \mathbb{Z} \cup \{\pm\infty\}, \max, +, -\infty, 0\rangle$ with $\sqcup = \max$ and $\sqcap = \min$:

$$\begin{aligned}
\boldsymbol{F}_U &= 2 + V \sqcap 2 + W \\
\boldsymbol{F}_V &= -1 + X \sqcup 0 \\
\boldsymbol{F}_W &= 4 + U \sqcup 4 + Y \sqcup 0 \\
\boldsymbol{F}_X &= 2 + W \sqcup 0 \\
\boldsymbol{F}_Y &= -3 + X \sqcap -3 + Z \\
\boldsymbol{F}_Z &= -1 + X \sqcup 0.
\end{aligned}$$

The set of variables is $\mathcal{X} = \{U, \dots, Z\}$ with $\mathcal{X}_\sqcup = \{V, W, X, Z\}$. The system is clean. We obtain from it the game graph $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$ depicted in Figure 5.3. ◇

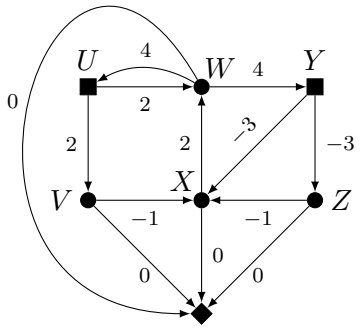


Figure 5.3: The game graph associated with the min-max-system of Example 5.4.3.

Lemma 5.4.4.
Let $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ be a linear min-max-system in normal form, and ρ a determinization of some terminal $\sqcup$-strategy σ obtained by $\sqcup$-strategy iteration. By restricting his moves to $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\rho}$ any play starting in $X \in \mathcal{X}$ has at least the value $\big(\mu\boldsymbol{F}\big)_X$.

Similarly, choose τ to be any deterministic $\sqcap$-strategy with $\boldsymbol{F}_\tau(\mu\boldsymbol{F}) = \mu\boldsymbol{F}$. Then player $\sqcap$ can bound from above the value of any play starting in $X \in \mathcal{X}$ to $\big(\mu\boldsymbol{F}\big)_X$ by sticking to τ. ◇

Proof. Consider $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\sigma}$. We already know that $\mu\boldsymbol{F} = \nu\boldsymbol{F}_\sigma = \boldsymbol{F}^n_\sigma(\top)$ and that all cycles $X_0 \xrightarrow{m_0} \ldots \xrightarrow{m_l} X_{l+1}$ (with $X_0 = X_{l+1}$) in $\mathcal{G}_{\boldsymbol{F}_\sigma}$ satisfy

$$1 \sqsubset \prod_{j=0}^{l} D_{m_j} X_{j+1}$$

by Lemma 5.3.18. Here, we have used that $\boldsymbol{F}$ is linear, and so the derivatives $D_{m_j}X_{j+1}$ are constant. This means that all cycles in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\sigma}$ also have weight greater than 1. Hence, every play in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_{\sigma^*}}$ that does not end up in $\blacklozenge$ has value $\top$. As ρ is a determinization of σ, these properties carry over to ρ ([8]). We therefore only have to show that the value of any play starting from some $X \in \mathcal{X}$ to $\blacklozenge$ is bound from below by $\boldsymbol{F}^n_\rho(\top)_X$. Fix some $X_0 \in \mathcal{X}$ and consider any play $X_0 \xrightarrow{c_0} X_1 \xrightarrow{c_1} \ldots \xrightarrow{c_l} \blacklozenge$ from X_0 to $\blacklozenge$ (with $l < n$) in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\rho}$. We show by induction on l that the value of such a play is at least $\boldsymbol{F}^{l+1}_\rho(\top)_{X_0}$:

[8] In particular $\nu\boldsymbol{F}_\sigma = \nu\boldsymbol{F}_\rho \sqsubseteq \boldsymbol{F}^n_\rho(\top) \sqsubseteq \boldsymbol{F}^n_\sigma(\top) = \nu\boldsymbol{F}_\sigma$.

- Assume $l = 0$. Then we have $X_1 = \blacklozenge$ and, so, $X_0 \in \mathcal{X}_\sqcup$. As ρ is deterministic, this is the only, and, thus, minimal play starting from X_0 in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\rho}$. The play has $c_0 = \boldsymbol{F}_{X_0}(\bot)$ as value by definition. As ρ is deterministic, we also have $(\boldsymbol{F}_\rho)_{X_0} = \boldsymbol{F}_{X_0}(\bot)$.
- Assume now $l > 0$. Then $X_1 \in \mathcal{X}$ and $X_1 \xrightarrow{c_1} \ldots \xrightarrow{c_l} \blacklozenge$ is a play from X_1 to $\blacklozenge$ of length $l-1$. By induction on l we have $\prod_{j=1}^{l} c_j \sqsupseteq \boldsymbol{F}^l_\rho(\top)_{X_1}$. So we obtain:

$$\prod_{j=0}^{l} c_j \sqsupseteq c_0 \cdot \boldsymbol{F}^l_\rho(\top)_{X_1} \sqsupseteq \boldsymbol{F}^{l+1}_\rho(\top)$$

  as $(\boldsymbol{F}_\rho)_{X_0}$ is the minimum of its monomials, one of them $c_0 \cdot X_1$.

As $\boldsymbol{F}^n_\rho(\top) = \nu\boldsymbol{F}_\rho = \mu\boldsymbol{F}$, the claim follows.

Consider now any deterministic $\sqcap$-strategy with $\boldsymbol{F}_\tau(\mu\boldsymbol{F}) = \mu\boldsymbol{F}$. Obviously, there are only finitely many plays in a given game, as the set of plays starting in X_0 is a subset of paths of length at most $n = |\mathcal{X}|$ in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\tau}$. So, there is a play of maximal value in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\tau}$ for any given $X_0 \in \mathcal{X}$. Let $X_0 \xrightarrow{c_0} X_1 \xrightarrow{c_1} \ldots \xrightarrow{c_l} X_{l+1}$ be such a play of maximal value. Assume that its value is greater than $\big(\mu\boldsymbol{F}\big)_{X_0}$.

Consider first the case that the play ends up in a cycle, so the value of the play is either $\top$ or $\bot$. As $\boldsymbol{F}$ is clean, we have $\big(\mu\boldsymbol{F}\big)_{X_0} \sqsupset \bot$, and so by our assumption the value of the play has to be $\top$, i.e., the weight of the cycle is greater than 1. Define $\boldsymbol{g} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ by

$$\begin{array}{lcll} \boldsymbol{g}_{X_j} & := & c_j \cdot X_{j+1} & \text{if } X_j \in \mathcal{X}_\sqcup \\ \boldsymbol{g}_{X_j} & := & c_j \cdot X_{j+1} & \text{if } X_j \in \mathcal{X}_\sqcap \\ \boldsymbol{g}_X & := & \bot & \text{if } X \in \mathcal{X} \setminus \{X_0, X_1, \ldots, X_l\}. \end{array}$$

As $\boldsymbol{F}_\tau$ is a pure max-system, we have $\boldsymbol{g} \sqsubseteq \boldsymbol{F}_\tau$, and so $\boldsymbol{g}(\mu\boldsymbol{F}) \sqsubseteq \boldsymbol{F}_\tau(\mu\boldsymbol{F}) = \mu\boldsymbol{F}$. So, as the cycle of the play has weight greater than 1, it immediately follows that $\top = \big(\mu\boldsymbol{F}\big)_{X_j}$ for all variables of the play. Hence, the value of the play is not greater than $\big(\mu\boldsymbol{F}\big)_{X_0}$.

We turn to the case that $X_{l+1} = \blacklozenge$. We again define a system $\boldsymbol{g} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ using the play:

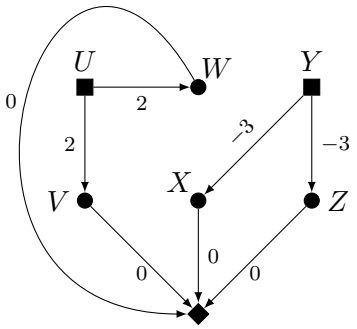
$$\begin{array}{lcll} \boldsymbol{g}_{X_j} & := & c_j \cdot X_{j+1} & \text{for } j \in \{0, \ldots, l-1\} \\ \boldsymbol{g}_{X_l} & := & c_l & \\ \boldsymbol{g}_X & := & \bot & \text{if } X \in \mathcal{X} \setminus \{X_0, X_1, \ldots, X_l\}. \end{array}$$

Again, we have $\boldsymbol{g} \sqsubseteq \boldsymbol{F}_\tau$, and so $\boldsymbol{g}(\mu\boldsymbol{F}) \sqsubseteq \boldsymbol{F}_\tau(\mu\boldsymbol{F}) = \mu\boldsymbol{F}$. As $\boldsymbol{g}$ is acylic, it has a unique fixed point given, e.g., by $\boldsymbol{g}^n(\mu\boldsymbol{F})$. In particular, $\boldsymbol{g}^n(\mu\boldsymbol{F})_{X_0}$ is the value of the play considered. By monotonicity we have $\boldsymbol{g}^n(\mu\boldsymbol{F}) \sqsubseteq \mu\boldsymbol{F}$, and so the value of the play cannot be greater than $\big(\mu\boldsymbol{F}\big)_{X_0}$. □

Corollary 5.4.5.
With the notations and assumptions of Lemma 5.4.4 it follows that $\mu\boldsymbol{F}$ corresponds to the optimal play values. Both players can use their deterministic strategies ρ, resp. τ in order to achieve these values. ◇

Example 5.4.6. We solve the system of Example 5.4.3 using locally optimal $\sqcup$-strategy iteration. By definition the strategy σ_0 chooses the constant monomials, i.e., we have

$$\begin{aligned}
\big(\boldsymbol{F}_{\sigma_0}\big)_U &= 2 + V \sqcap 2 + W \\
\big(\boldsymbol{F}_{\sigma_0}\big)_V &= 0 \\
\big(\boldsymbol{F}_{\sigma_0}\big)_W &= 0 \\
\big(\boldsymbol{F}_{\sigma_0}\big)_X &= 0 \\
\big(\boldsymbol{F}_{\sigma_0}\big)_Y &= -3 + X \sqcap -3 + Z \\
\big(\boldsymbol{F}_{\sigma_0}\big)_Z &= 0.
\end{aligned}$$

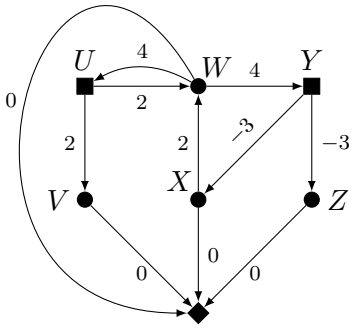


On the right, the graph $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_{\sigma_0}}$ is shown. Obviously, this system is acyclic. Its unique fixed point is

$$\nu\boldsymbol{F}_{\sigma_0} = (2, 0, 0, 0, -3, 0) \sqsubseteq (2, 0, 6, 2, -3, 0) = \boldsymbol{F}(\nu\boldsymbol{F}_{\sigma_0}).$$

The reader can easily check that the values of $\nu\boldsymbol{F}_{\sigma_0}$ correspond to the weight of paths, i.e., plays, from the respective variable (node) to $\blacklozenge$. By comparing $\nu\boldsymbol{F}_{\sigma_1}$ to $\boldsymbol{F}(\nu\boldsymbol{F}_{\sigma_0})$ we see that we can improve the strategy at the variables W, X. By definition of the locally optimal $\sqcup$-strategy iteration we obtain:

$$\begin{aligned}
\big(\boldsymbol{F}_{\sigma_1}\big)_U &= 2 + V \sqcap 2 + W \\
\big(\boldsymbol{F}_{\sigma_1}\big)_V &= 0 \\
\big(\boldsymbol{F}_{\sigma_1}\big)_W &= 4 + U \sqcup 4 + Y \sqcup 0 \\
\big(\boldsymbol{F}_{\sigma_1}\big)_X &= 2 + W \sqcup 0 \\
\big(\boldsymbol{F}_{\sigma_1}\big)_Y &= -3 + X \sqcap -3 + Z \\
\big(\boldsymbol{F}_{\sigma_1}\big)_Z &= 0.
\end{aligned}$$



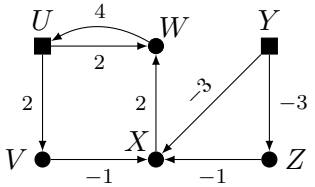
As shown, $\nu\boldsymbol{F}_{\sigma_1}$ is given by $\boldsymbol{F}^6_{\sigma_1}(\infty)$. This yields:

$$\nu\boldsymbol{F}_{\sigma_1} = (2, 0, 6, 8, -3, 0) \sqsubseteq (2, 7, 6, 8, -3, 7) = \boldsymbol{F}(\nu\boldsymbol{F}_{\sigma_1}).$$

Note that every cycle in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_{\sigma_1}}$ has weight > 0. From this it immediately follows that the best that player $\sqcap$ can do in any game starting from Y is to play directly to Z resulting in the play value -3 – as otherwise player $\sqcup$ could play into the cycle $\{W, Y, X\}$. Similarly, player $\sqcup$ is always forced to play from U to V if he wants to optimize, that is minimize the value of any play visiting U.

By comparing again $\nu\boldsymbol{F}_{\sigma_1}$ to $\boldsymbol{F}(\boldsymbol{F}_{\sigma_1})$, we notice that this time we have to adapt the strategy at all $\sqcup$-variables:

$$
\begin{aligned}
\big(\boldsymbol{F}_{\sigma_2}\big)_U &= 2 + V \sqcap 2 + W \\
\big(\boldsymbol{F}_{\sigma_2}\big)_V &= -1 + X \\
\big(\boldsymbol{F}_{\sigma_2}\big)_W &= 4 + U \\
\big(\boldsymbol{F}_{\sigma_2}\big)_X &= 2 + W \\
\big(\boldsymbol{F}_{\sigma_2}\big)_Y &= -3 + X \sqcap -3 + Z \\
\big(\boldsymbol{F}_{\sigma_2}\big)_Z &= -1 + X.
\end{aligned}
$$



◆

W.r.t. to this last strategy now, every play ends up either in the cycle $U \to V \to X \to W \to V$ or $X \to U \to X$. In both cases, the weight of these cycles is positive, hence, the weight of any play is ∞. One easily checks that $\nu\boldsymbol{F}_{\sigma_2}$ is indeed ∞ in every component by calculating the sequence $\boldsymbol{F}^k_{\sigma_2}(\infty)$.

We conclude that $\mu\boldsymbol{F} = \nu\boldsymbol{F}_{\sigma_2}$, and $\mu\boldsymbol{F}$ indeed gives us the optimal play values both players can achieve in the game played on $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$. ◇

In Section 5.5 we apply these results to parity games. For now note that from the interpretation of a clean linear min-max-system $\boldsymbol{F}$ as a game on $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}}$ it is *reasonable* for player $\sqcup$ to only consider strategies σ such that every cycle in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}_\sigma}$ has weight greater than 1 as this forces player $\sqcap$ to try to play to $\blacklozenge$ in order to minimize the value of the play. Note that for a clean min-cycle-free linear min-max-system any such strategy is also reasonable w.r.t. Definition 5.3.11 as we may simply take $\mu\boldsymbol{F}$ as witness:

Proposition 5.4.7.
Let $\boldsymbol{F}$ be a linear min-max-system in normal form and σ a max-strategy.

Then σ is reasonable if every cycle of $\mathcal{G}_{\boldsymbol{F}_\sigma}$ has weight greater than 1. ◇

In the following subsection we use the notion of reasonable strategies in order to obtain an improved bound on the length of locally optimal strategy iteration sequence when applied to linear min-max-systems.

5.4.2 An Improved Bound on the Number of Iterations

In the rest of this subsection we use the fact that player $\sqcup$ only needs to consider *reasonable* strategies in order to obtain a better bound on the number of iterations needed to calculate $\mu\boldsymbol{F}$ in the case of a linear min-max-system in normal form where we additionally assume that every $\sqcup$-equation of $\boldsymbol{F}$ consists of at most two monomials.

For such an $\boldsymbol{F}$ the number $N_\sqcup$ is then trivially bounded from above by $2^{|\mathcal{X}_\sqcup|}$, as any deterministic $\sqcup$-strategy chooses exactly one of the at most 2 monomials

for any $X \in \mathcal{X}_{\sqcup}$. In the following we improve this trivial upper bound. When using the heuristic described in the previous paragraph, i.e., $\sigma_{i+1} = I_{\sigma_i}$ (cf. Definition 5.3.22), we can improve the upper bound to $O(1.724^{|\mathcal{X}_{\sqcup}|})$.

Fix now the locally optimal strategy iteration sequence $(\sigma_i)_{i\in\{0,1,\dots,L\}}$. We have already seen that for every σ_i there is a determinization ρ_i such that $\nu\boldsymbol{F}_{\sigma_i} = \nu\boldsymbol{F}_{\rho_i}$, and $\rho_i(X) \subseteq \sigma_i(X)$ for all $X \in \mathcal{X}_{\sqcup}$. In particular, as σ_i is reasonable, so is ρ_i (cf. Proposition 5.3.16).

For a $\sqcup$-strategy σ, let $\mathrm{src}(S_\sigma)$ be the set of $\sqcup$-variables for which there exists at least one monomial which strictly increases at $\nu\boldsymbol{F}_\sigma$, i.e.,

$$\mathrm{src}(S_\sigma) := \{X \in \mathcal{X}_{\sqcup} \mid S_\sigma(X) \neq \emptyset\}.$$

We then set $s(\sigma) := |\mathrm{src}(S_\sigma)|$ and $s_i := s(\sigma_i) = s(\rho_i)$.

We obtain $2^{s_i} - 1$ many new *deterministic* strategies δ by changing ρ_i at exactly one of the variables $\mathrm{src}(S_{\sigma_i}) = \mathrm{src}(S_{\rho_i})$. As we assume that $\boldsymbol{F}_X$ consists of at most two monomials for every $X \in \mathcal{X}_{\sqcup}$ there is exactly one way to change ρ_i at every $X \in \mathrm{src}(S_{\sigma_i})$. Note that ρ_{i+1} does not need to be to one of these strategies δ. For every such δ we then have

$$\nu\boldsymbol{F}_{\rho_i} = \nu\boldsymbol{F}_{\sigma_i} \sqsubset \nu\boldsymbol{F}_\delta \sqsubseteq \nu\boldsymbol{F}_{\sigma_{i+1}}$$

as $\delta(X) \subseteq \sigma_{i+1}(X)$ for all $X \in \mathcal{X}_{\sqcup}$.

As the approximations $\nu\boldsymbol{F}_{\sigma_i}$ strictly increase until the sequence terminates, we know that none of these strategies appears along the sequence $(\sigma_i)_{i=0,\dots,L}$. Therefore, at least $2^{s_i} - 1$ new deterministic strategies can be ruled out as candidates for optimal winning strategies.

Hence, if S_k is the number of deterministic strategies which have at most k nodes at which there exists at least one strict improvement, we get as an upper bound for the number of improvement steps

$$S_k + \frac{2^{|\mathcal{X}_{\sqcup}|}}{2^{k+1} - 1} \leq S_k + 2^{|\mathcal{X}_{\sqcup}|-k}.$$

We next bound the number of strategies σ_i having the same value s_i.

Lemma 5.4.8.
Let $\boldsymbol{F}$ be a linear min-max-system in normal form, and σ a reasonable $\sqcup$-strategy with $\boldsymbol{F}_X(\bot) \sqsubseteq \left(\nu\boldsymbol{F}_\sigma\right)_X$ for all $X \in \mathcal{X}_{\sqcup}$. Let $\boldsymbol{G}$ be the system we obtain from $\boldsymbol{F}$ by removing from $\boldsymbol{F}_X$ all monomials of $S_\sigma(X)$ for $X \in \mathcal{X}_{\sqcup}$ (with S_σ taken w.r.t. $\boldsymbol{F}$).

Then, $\boldsymbol{G}$ is also in normal form, and σ a reasonable strategy for $\boldsymbol{G}$ with $\nu\boldsymbol{G}_\sigma = \mu\boldsymbol{G}$. Further, for any reasonable $\sqcup$-strategy ρ w.r.t. $\boldsymbol{F}$ with $\rho(X) \cap S_\sigma(X) = \emptyset$ for all $X \in \mathcal{X}_\sqcup$, ρ is a also reasonable w.r.t. $\boldsymbol{G}$ with $\nu\boldsymbol{G}_\rho \sqsubseteq \nu\boldsymbol{G}_\sigma$. $\diamond$

Proof. We first show that $\boldsymbol{G}$ is also clean. As we require that $\boldsymbol{F}_X(\bot) \sqsubseteq (\nu\boldsymbol{F}_\sigma)_X$ for all $X \in \mathcal{X}_\sqcup$, we have $\boldsymbol{F}_X(\bot) \notin S_\sigma(X)$. So, $\boldsymbol{F}_X(\bot) = \boldsymbol{G}_X(\bot)$ follows. By definition of normal form, we also have $\bot \sqsubset \boldsymbol{F}_X(\bot)$ for $X \in \mathcal{X}_\sqcup$. Finally, as $\boldsymbol{F}$ is min-cycle-free, so is $\boldsymbol{G}$, and one easily shows that $\boldsymbol{G}$ is clean.

We next show that σ is reasonable. For this note that $\sigma(X) \cap S_\sigma(X) = \emptyset$ for all $X \in \mathcal{X}_\sqcup$. So, σ can still be applied to $\boldsymbol{G}$ with $\boldsymbol{G}_\sigma = \boldsymbol{F}_\sigma$. In particular, every cycle of $\boldsymbol{G}_\sigma$ has weight greater than 1. As $\boldsymbol{G}$ is clean, σ is therefore reasonable w.r.t. $\boldsymbol{G}$. Analogously, one shows that ρ is reasonable w.r.t. $\boldsymbol{G}$.

Therefore by Lemma 5.3.12, we have $\nu\boldsymbol{G}_\sigma \sqsubseteq \mu\boldsymbol{G}$, and $\nu\boldsymbol{G}_\rho \sqsubseteq \mu\boldsymbol{G}$. In particular, $\nu\boldsymbol{G}_\sigma$ is a fixed point of $\boldsymbol{G}$, as we have removed exactly those monomials which increase at $\nu\boldsymbol{G}_\sigma$. Hence, $\nu\boldsymbol{G}_\rho \sqsubseteq \mu\boldsymbol{G} = \nu\boldsymbol{G}_\sigma$. $\square$

The preceding lemma tells us that if $\sigma_i \cap S_{\sigma_j} = \emptyset$ holds for two strategies of a $\sqcup$-strategy iteration, then $\nu\boldsymbol{F}_{\sigma_i} \sqsubseteq \nu\boldsymbol{F}_{\sigma_j}$, i.e., $i \leq j$.

We use this to show that whenever $\mathrm{src}(S_{\sigma_j}) \subseteq \mathrm{src}(S_{\sigma_i})$, then we have $\nu\boldsymbol{F}_{\sigma_i} \sqsubseteq \nu\boldsymbol{F}_{\sigma_j}$, too. This in turn tells us that if $i \neq j$ and $s_i = s_j$, then $\mathrm{src}(S_{\sigma_j}) \neq \mathrm{src}(S_{\sigma_i})$, which means that the number of strategies with the same value s_i is bounded by the number of distinct subsets of $\mathcal{X}_\sqcup$ of size s_i, i.e., $\binom{|\mathcal{X}_\sqcup|}{s_i}$. We formalize this in the following lemma which is a generalization of a result which can be found in [MS99] for Markov decision processes.

Lemma 5.4.9.
Let $\boldsymbol{F}$ be a linear min-max-system in normal form, and σ_i and σ_j two reasonable $\sqcup$-strategies w.r.t. $\boldsymbol{F}$ of the locally optimal strategy iteration sequence.

If $\mathrm{src}(S_{\sigma_j}) \subset \mathrm{src}(S_{\sigma_i})$, then $\nu\boldsymbol{F}_{\sigma_i} \sqsubseteq \nu\boldsymbol{F}_{\sigma_j}$. $\diamond$

Proof. Set $C = S_{\sigma_j} \cap \sigma_i$. For every $X \in \mathrm{src}(C)$ we find a monomial $m_{C,X}$ such that $(X, m_{X,C}) \in C$, a monomial $m_{X,j}$ with $(X, m_{X,j}) \in \sigma_j$ (as σ_j is a strategy), and a monomial $m_{X,i}$ with $(X, m_{X,i}) \in S_{\sigma_i}$ (as $\mathrm{src}(S_{\sigma_j}) \subseteq \mathrm{src}(S_{\sigma_i})$).

Now, because of $S_\sigma \cap \sigma = \emptyset$ for any strategy σ, we may conclude that both $m_{X,C} \neq m_{X,i}$ and $m_{X,C} \neq m_{X,j}$. Thus, as we assume that $|sE| \leq 2$, we have $m_{X,i} = m_{X,j} =: m_X$. We define therefore $C' = \{(X, m_X) \mid X \in \mathrm{src}(C)\}$, and

$$\sigma' := C' \cup \big(\sigma_i \setminus C\big).$$

As $C' \subseteq S_{\sigma_i}$, we have $\emptyset \neq \sigma'(X) \subseteq I_{\sigma_i}(X)$ for all $X \in \mathcal{X}_\sqcup$. Hence, σ' is a possible successor strategy of σ_i, and so $\nu\boldsymbol{F}_{\sigma_i} \sqsubseteq \nu\boldsymbol{F}_{\sigma'}$.

In particular, σ' is reasonable with $\boldsymbol{F}_X(\bot) \sqsubseteq \left(\nu\boldsymbol{F}_{\sigma_i}\right)_X \sqsubseteq \left(\nu\boldsymbol{F}_{\sigma'}\right)_X$. As $\sigma' \cap S_{\sigma_j} = \emptyset$, we may apply Lemma 5.4.8 yielding $\nu\boldsymbol{F}_{\sigma'} \sqsubseteq \nu\boldsymbol{F}_{\sigma_j}$. □

Corollary 5.4.10.
The number of strategies σ_i along the locally optimal $\sqcup$-strategy iteration having the number $s = s(\sigma_i)$ is bounded by $\binom{|\mathcal{X}_\sqcup|}{s}$. ◇

We now are ready to give an improved upper bound on the number of steps done by the locally optimal $\sqcup$-strategy iteration:

Theorem 5.4.11.
For a given linear min-max-system $\boldsymbol{F} : S^{\mathcal{X}} \to S^{\mathcal{X}}$ in normal form the locally optimal strategy iteration terminates after at most

$$3 \cdot e^{0.545 \cdot |\mathcal{X}_\sqcup|} \leq 3 \cdot 1.724^{|\mathcal{X}_\sqcup|}$$

iterations. ◇

Proof. Recall that by S_k we denote the number of deterministic strategies which have at most k variables for which there is a strict improvement. We have already argued that then $S_k + 2^{|\mathcal{X}_\sqcup|-k}$ is an upper bound on the number of iterations.

As long as $1 \leq k \leq \frac{|\mathcal{X}_\sqcup|}{3}$, it follows from Corollary 5.4.10 that

$$S_k \leq \sum_{k'=0}^{k} \binom{|\mathcal{X}_\sqcup|}{k'} \leq 2\binom{|\mathcal{X}_\sqcup|}{k} \leq 2\left(\frac{|\mathcal{X}_\sqcup|}{k} \cdot e\right)^k.$$

What remains is to find a $1 \leq k \leq \frac{|\mathcal{X}_\sqcup|}{3}$ such that

$$2\left(\frac{|\mathcal{X}_\sqcup|}{k} \cdot e\right)^k + 2^{|\mathcal{X}_\sqcup|-k}$$

is minimal. For this set $b = \frac{|\mathcal{X}_\sqcup|}{k}$ with $b \geq 3$, yielding

$$2 \cdot e^{|\mathcal{X}_\sqcup| \cdot \frac{1+\ln b}{b}} + e^{\ln 2 \cdot |\mathcal{X}_\sqcup| \cdot \frac{b-1}{b}}.$$

As $\frac{1+\ln b}{b}$ is strictly decreasing and $\frac{b-1}{b}$ is strictly increasing, we need to look for the largest $b \geq 3$ such that

$$\frac{1+\ln b}{b} \geq \ln 2 \cdot \frac{b-1}{b}.$$

Using e.g. Newton's method one can easily check that $b \in (4.6, 4.7)$ with $b \approx 4.66438$. □

In [BSV02] Björklung, Sandberg and Vorobyov analyze randomized selection of the successor strategy: they choose uniformly at random a *deterministic* successor strategy $\sigma' \subseteq I_\sigma$, and show that the expected number of iterations needed to reach the optimum is always less than $1.71^{|\mathcal{X}_\sqcup|}$. Similar to

our case, they use that the random selection of the successor strategy leads to skipping an exponential number of strategies. It remains to analyze if their proof allows to lower the upper bound in our setting of locally optimal strategy iteration and linear, or even nonlinear min-max-systems. Note that our bound relies heavily on Lemma 5.4.8. A starting point for improving the trivial upper bound on the number of iterations in the case of nonlinear min-max-systems therefore would be the extension of this lemma.

5.5 Application to Parity Games

In the previous section we have seen that every clean linear min-max-system $\boldsymbol{F}$ can be illustrated by means of a two-person game played on an extension of the dependency graph of $\boldsymbol{F}$. The least fixed point of $\boldsymbol{F}$ then becomes the optimal play values the two players can hope to achieve when both of them are playing optimal (cf. Lemma 5.4.2). We have further seen that for such games the optimal play values can be obtained in at most $O(1.724^{|\mathcal{X}_\sqcup|})$ steps by locally optimal $\sqcup$-strategy iteration, see Theorem 5.4.11. The aim of this section is to apply these results to *parity games*. By virtue of the results of the preceding Section 5.4, we construct from a parity game a clean linear min-max-system $\boldsymbol{FA}$ on a si-semiring over $\mathbb{Z}^k$. This semiring captures the same information as the *play values* used in [VJ00]. By construction, the dependency graph of the min-max-system $\boldsymbol{FA}$ and the original parity game coincide, allowing us to directly apply the results of Section 5.4. We obtain from this the algorithm of [Lut08]. In particular, this algorithm does not rely on the reduction of parity games to mean payoff games. The algorithm therefore retains more information on the original parity game. This makes it possible to compare it to the well-known algorithm by Jurdzinski and Vöge [VJ00]. We refer the reader to [Lut08] for this comparison.

In [GS08] Gawlitza and Seidel study another reduction of parity games to linear min-max-systems. Their approach is based on the well-known reduction of parity games to mean payoff games, see e.g. [Jur98], yielding a min-max-system on the si-semiring $\langle \mathbb{Z} \cup \{\pm\infty\}, \max, +, -\infty, 0\rangle$. In contrast to the reduction proposed in this section, the connection between the resulting min-max-system and the parity game is lost when moving from parity games to mean payoff games as an intermediate step.

Parity Games: A parity game is played by two persons, in the following referred to as player 0 and player 1, on a so called *(parity game) arena*

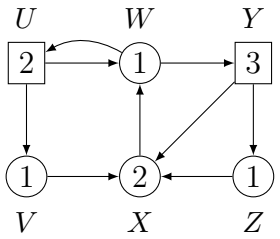


Figure 5.4: A parity game arena. The shape of the nodes indicates the owner, circular nodes belong to player 0, while boxed shaped nodes belong to player 1. Colors are written inside of the nodes. Above, resp. below the nodes their nodes are written.

$\mathcal{A} = (\{V_0, V_1\}, E, \chi)$ which is a directed graph, with nodes $V := V_0 \cup V_1$ and edge relation E, whose nodes are split between the two players, that is every node belongs either to player 0 or to player 1, where we use V_i to denote the nodes owned by player i; further, every node $v \in V$ of the graph is assigned a color $\chi(v)$ in $\{1, 2, \ldots, d\}$ where d is some fixed natural number. We assume that every node of the arena has at least one outgoing edge, i.e., $vE \neq \emptyset$ for all $v \in V$. A play $\pi : \mathbb{N} \to V$ of the two players starts from some initial node $\pi(0)$, and is a maximal, thus infinite, path through the arena where the owner of node $\pi(k)$ chooses $\pi(k+1)$ from $\pi(k)E$ for all $k \in \mathbb{N}$. The parity of the highest color seen infinitely often along π determines then the winner, i.e., player i wins π if $\limsup_{k\to\infty} \chi(\pi(k)) \equiv i \pmod 2$.

Example 5.5.1. Consider the parity game arena depicted in Figure 5.4. A possible play starting in U is the infinite word $U(WYX)^\omega$, i.e., the play ends up in the cycle consisting of $\{W, Y, X\}$. The colors appearing infinitely often along this play are $\{1, 2, 3\}$. Hence, this play is won by player 1. ◇

We say that player i wins a node $v \in V$ if he can react on every possible move of his opponent in such a way that he wins the resulting play. It is a well-known result shown in 1991 independently by Mostowski, respectively Emerson and Jutla that in order to win the maximal set of nodes, it is sufficient for both players to simply choose for every node v owned by them a single successor $\sigma(v)$ such that they move in every play hitting v to $\sigma(v)$:

Theorem 5.5.2 ([Mos91, EJ91])**.**
For any a parity game arena $\mathcal{A}$ there is a unique partition $\{W_0, W_1\}$ of V, and functions σ^*, τ^* with $\sigma^* : V_0 \to V$, and $\tau : V_1 \to V$ such that player 0, resp. player 1 wins every play π with $\pi(0) \in W_i$ by only moving along the edges $\{(v, \sigma^*(v)) \mid v \in V_i\}$, resp. $\{(v, \tau^*(v)) \mid v \in V_1\}$. ◇

The functions σ^* and τ^* are called deterministic memoryless strategies where

"deterministic" refers to fact that they fix for every $v \in V_i$ a unique determined successor, and "memoryless" emphasizes the fact that the move from $v \in V_0$ to $\sigma^*(v)$ (similarly for τ) does not depend on the history of the play. The important problem is then to determine the partition $\{W_0, W_1\}$, and the strategies σ^*, and τ^* for a given parity game. We only mention that deciding whether a node belongs to W_0 or W_1 is equivalent to the problem of deciding whether a given Kripke structure satisfies a given μ-calculus formula and refer the reader to [GTW02] for further details.

The next lemma basically states that we can always assume that $\mathcal{A}$ restricted to the nodes V_1 (short: $\mathcal{A}|_{V_1}$) is acyclic. In the following paragraph, this will allow us to associate with every parity game arena a clean linear min-max-system, and reduce the problem of finding $\{W_0, W_1\}$ to the problem of calculating the least fixed point of the associated min-max-system.

Lemma 5.5.3.
Let $\mathcal{A} = (\{V_0, V_1\}, E, \chi)$ be a parity game arena with $V := V_0 \cup V_1$. Then one can obtain from $\mathcal{A}$ in time polynomial in $|V|$ an arena $\mathcal{A}'$ and sets $U_0, U_1 \subseteq V$ such that (1) the nodes W_i won by player i in $\mathcal{A}$ are equal to $U_i \cup W_i'$ where W_i' is the set of nodes one by player i in $\mathcal{A}'$, and (2) $\mathcal{A}'$ restricted to the nodes owned by player 1 is acyclic. ◇

Proof. First, recall the concept of attractor: given a set $X \subseteq V$ of nodes the set $\mathrm{Attr}_i(X)$ consists of all nodes from which player i can force a play to a node in X. This set can be calculated in polynomial time by means of a simple fixed point iteration:

- Define the operator $A_i : 2^V \to 2^V$ for some $Y \subseteq V$ by

$$\begin{array}{rcl} A_i(Y) & := & Y \\ & \cup & \{v \in V_i \mid vE \cap Y\} \\ & \cup & \{v \in V_{1-i} \mid vE \subseteq Y\}. \end{array}$$

- Then $\mathrm{Attr}_i(X) := \mu_X A_i = \bigcup_{k \in \mathbb{N}} A_i^k(X)$.

Note that A_i is a monotone, ω-continuous operator on the powerset algebra. So, the fixed point $\mu_X A_i$ always exists and can be calculated in time polynomial in the size of the graph.

Consider now $\mathcal{A}|_{V_1}$, i.e., the restriction of $\mathcal{A}$ to the nodes V_1, and set $U_0 := \emptyset =: U_1$. Fix some order on the nodes of $\mathcal{A}|_{V_1}$ having odd color. For every such node v check if there is a (simple) cycle containing v in $\mathcal{A}|_{V_1}$ such that every node of this cycle is not greater than v w.r.t. the color. This can be done also in time $O(|E|)$ by means of a depth first search obviously. If so, player 1 wins all nodes located in $\mathrm{Attr}_1(\{v\})$. Hence, set $U_1 := U_1 \cup \mathrm{Attr}_1(\{v\})$, and remove $\mathrm{Attr}_1(\{v\})$ from $\mathcal{A}$, and call the resulting arena $\mathcal{A}$ again. We repeat these steps until there is no cycle in $\mathcal{A}|_{V_1}$ anymore whose highest color is odd. Then, player 1 has to force any play into V_0 as otherwise he will lose. So, calculate

$\mathrm{Attr}_1(V_0)$, and set $U_0 := V \setminus \mathrm{Attr}_1(V_0)$ as from these nodes player 1 is forced to stay in V_1, and, thus, ending up in a cycle whose highest color is even. Finally, remove also U_0 from $\mathcal{A}$, and all edges leading from $A_1^k(V_0)$ to some $A_1^l(V_0)$ with $l \geq k$, resulting in the arena $\mathcal{A}'$. Obviously, $\mathcal{A}'|_{V_1}$ is acyclic now. So it suffices to consider $\mathcal{A}'$, as all nodes in U_i are trivially won by player i. □

Min-Max-System: Let $\mathcal{A} = (\{V_0, V_1\}, E, \chi)$ be a parity game arena. We assume in the following that every node owned by player 1 has at least two successors – obviously, assigning nodes with exactly one successor to player 0 does not give him any advantage. We further make use of Lemma 5.5.3, and assume that $\mathcal{A}|_{V_1}$ is acyclic. With such a parity game arena $\mathcal{A}$ we associate a linear min-max-system $\boldsymbol{F}$ on the following si-semiring:

Definition 5.5.4.
Let $d \in \mathbb{N}$ be the maximal color of a parity game arena $\mathcal{A}$. The structure $\mathcal{P}_d := \langle \mathbb{Z}^d \cup \{\pm\infty\}, \sqsubseteq, +, -\infty, \infty, \mathbf{0}\rangle$ is then defined as follows:

- For $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}^d$ let $\delta(\boldsymbol{a}, \boldsymbol{b}) := \max\{i \in \{1, \ldots, d\} \mid \boldsymbol{a}_i \neq \boldsymbol{b}_i\}$ denote the greatest color in which $\boldsymbol{a}$ and $\boldsymbol{b}$ differ ([9]). Then

$$\begin{array}{rcll} \boldsymbol{a} \sqsubseteq \boldsymbol{b} & :\Leftrightarrow & \boldsymbol{a} = \boldsymbol{b} & \\ & \vee & \delta(\boldsymbol{a}, \boldsymbol{b}) \text{ odd} & \wedge \quad \boldsymbol{a}_{\delta(\boldsymbol{a},\boldsymbol{b})} > \boldsymbol{b}_{\delta(\boldsymbol{a},\boldsymbol{b})} \\ & \vee & \delta(\boldsymbol{a}, \boldsymbol{b}) \text{ even} & \wedge \quad \boldsymbol{a}_{\delta(\boldsymbol{a},\boldsymbol{b})} < \boldsymbol{b}_{\delta(\boldsymbol{a},\boldsymbol{b})}. \end{array}$$

  We then extend $\sqsubseteq$ to $\mathbb{Z}^d \cup \{\pm\infty\}$ such that $-\infty$ is the least element, and ∞ the greatest element.

- By $+$ we denote the canonical componentwise addition on $\mathbb{Z}^d$ and extend this to $\mathbb{Z}^d \cup \{\pm\infty\}$ by requiring that conditions (2a) and (2b) of Definition 5.3.1 are satisfied, i.e.,

$$\begin{array}{rcl} \boldsymbol{a} + -\infty = -\infty + \boldsymbol{a} = -\infty & \text{for all } \boldsymbol{a} \in \mathbb{Z}^d \cup \{\pm\infty\} \\ \boldsymbol{a} + \infty = \infty + \boldsymbol{a} = \infty & \text{for all } \boldsymbol{a} \in \mathbb{Z}^d \cup \{\infty\}. \end{array}$$

  With $\mathbf{0}$ we denote the d-dimensional vector which is equal to 0 in every component.

We embed a color $k \in \{1, \ldots, d\}$ into $\mathbb{Z}^d$ by identifying it with the vector

$$(\underbrace{0, \ldots, 0}_{k-1}, 1, \underbrace{0, \ldots, 0}_{d-k}). \qquad \diamond$$

[9] We assume that $\max \emptyset = 0$.

Remark 5.5.5.
Note that $\sqsubseteq$ is total and that for any $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{Z}^d$ we have

$$\boldsymbol{a} \sqsubset \boldsymbol{b} \Leftrightarrow \boldsymbol{a} + \boldsymbol{c} \sqsubset \boldsymbol{b} + \boldsymbol{c}.$$

$\mathcal{P}_d$ is thus a si-semiring. ◇

We now associate with any parity game arena $\mathcal{A}$ a min-max-system on $\mathcal{P}_d$:

Definition 5.5.6.
Let $\mathcal{A} = (\{V_0, V_1\}, E, \chi)$ be a parity game arena such that $\mathcal{A}|_{V_1}$ is acyclic, and every node in V_1 has at least two successors. Then from $\mathcal{A}$ we obtain the min-max-system $\boldsymbol{F}\mathcal{A} : \mathcal{P}_d^{\mathcal{X}} \to \mathcal{P}_d^{\mathcal{X}}$ with $\mathcal{X} := V_0 \cup V_1$, and

$$\begin{array}{lcll} \boldsymbol{F}\mathcal{A}_v & := & \bigsqcup\{\chi(v) + w \mid w \in vE\} \sqcup \{\mathbf{0}\} & \text{if } v \in V_0 \\ \boldsymbol{F}\mathcal{A}_v & := & \bigsqcap\{\chi(v) + w \mid w \in vE\} & \text{if } v \in V_1. \end{array}$$ ◇

We then have $\mathcal{X}_\sqcup = V_0$, and $\mathcal{X}_\sqcap = V_1$.

Example 5.5.7. We continue with Example 5.5.1. The arena $\mathcal{A}$ shown in Figure 5.4 does not contain any cycles consisting only of nodes owned by player 1. We therefore obtain from it the min-max-system $\boldsymbol{F}\mathcal{A}$ on $\mathcal{P}_3$ with

$$\begin{array}{lcl} \boldsymbol{F}\mathcal{A}_U & = & (0,1,0) + V \sqcap (0,1,0) + W \\ \boldsymbol{F}\mathcal{A}_V & = & (1,0,0) + X \sqcup (0,0,0) \\ \boldsymbol{F}\mathcal{A}_W & = & (1,0,0) + U \sqcup (1,0,0) + Y \sqcup (0,0,0) \\ \boldsymbol{F}\mathcal{A}_X & = & (0,1,0) + W \sqcup (0,0,0) \\ \boldsymbol{F}\mathcal{A}_Y & = & (0,0,1) + X \sqcap (0,0,1) + Z \\ \boldsymbol{F}\mathcal{A}_Z & = & (1,0,0) + X \sqcup (0,0,0). \end{array}$$

Recall that color 1 corresponds to the vector $(1,0,0)$, color 2 to $(0,1,0)$, and 3 to $(0,0,1)$. Further, the neutral element w.r.t. the multiplication $(+)$ of the semiring is the null vector $\mathbf{0} = (0,0,0)$. ◇

Obviously, the dependency graph $\mathcal{G}_{\boldsymbol{F}\mathcal{A}}$ and $\mathcal{A}$ are in one-to-one correspondence: there is an edge $v \xrightarrow{c} w$ in $\mathcal{G}_{\boldsymbol{F}\mathcal{A}}$ if and only if there is in an edge from v to w in $\mathcal{A}$, and its weight c is exactly the color of v in $\mathcal{A}$, i.e., $c = \chi(v)$. Thus, the strategy σ_0^*, resp. σ_1^* w.r.t. $\mathcal{A}$ corresponds to a $\sqcup$-strategy, resp. $\sqcap$-strategy w.r.t. $\boldsymbol{F}\mathcal{A}$. Further, $\boldsymbol{F}$ is in normal form as every cycle in $\mathcal{A}$ contains at least one node owned by player 0, i.e., a $\sqcup$-variable. From this it also follows that $\boldsymbol{F}\mathcal{A}$ is clean: simply choose the $\sqcup$-strategy σ mapping every $v \in V_0$ to 1 w.r.t. $\boldsymbol{F}\mathcal{A}$; then, $\boldsymbol{F}\mathcal{A}_\sigma \sqsubseteq \boldsymbol{F}\mathcal{A}$, and the dependency graph of $\boldsymbol{F}\mathcal{A}_\sigma$ is acyclic, hence $-\boldsymbol{\infty} \sqsubset \boldsymbol{F}\mathcal{A}_\sigma^n(-\boldsymbol{\infty})_v = (\mu\boldsymbol{F}\mathcal{A}_\sigma)_v \sqsubseteq (\mu\boldsymbol{F}\mathcal{A})_v$ for every variable v.

We next show that for every node $v \in V$ we have that v is won by player 0 ($v \in W_0$) if and only if $(\mu\boldsymbol{F}\mathcal{A})_v = \infty$ holds. For this, consider first the optimal

strategy σ^* player 0 can use in $\mathcal{A}$ to win every node of W_0. Obviously, we can apply this strategy also to $\boldsymbol{F}\mathcal{A}$, resp. $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$. As player 0 wins every play π starting in W_0 by means of σ^*, and σ^* is deterministic, that is every play is determined completely by player 1, every cycle visited has an even color as highest color. From the play π in $\mathcal{A}$ we obtain a play on $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$ by simply stopping the play π as soon as the first simple cycle is completed. As the highest color of this cycle is even, one easily checks by the definitions that the total weight of this cycle is greater than $\mathbf{0}$ ([10]) and, hence, by Definition 5.4.2 the value of this play in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$ is ∞. By Lemma 5.4.4 it also follows that $(\mu\boldsymbol{F}\mathcal{A})_v = \infty$ for all $v \in W_0$, as $(\mu\boldsymbol{F}\mathcal{A})_v$ is the value of the the play in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$ starting from v when both players play optimal. Consider thus any node $v \in W_1$. Here, player 1 can use his strategy τ^* to force any play in $\mathcal{A}$ starting from v into cycle whose highest color is odd. In $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$ now player 0, i.e., player $\sqcup$ can choose between ending up in a cycle or escaping to $\blacklozenge$. In both cases the value of the resulting play is less than ∞. Again by Lemma 5.4.4 it therefore follows that $(\mu\boldsymbol{F}\mathcal{A})_v \sqsubset \infty$ for all $v \in W_1$. Hence, we obtain the following result by virtue of Theorem 5.4.11:

Theorem 5.5.8.
Player 0 wins the node v in the parity game arena $\mathcal{A}$ if and only if $(\mu\boldsymbol{F}\mathcal{A})_v = \infty$ holds. One can calculate $\mu\boldsymbol{F}\mathcal{A}$ and, thus, the winning sets W_0 and W_1 using $O(1.724^{|V_0|})$ steps of locally optimal strategy iteration. ◇

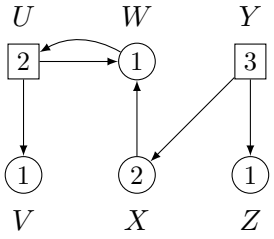
Example 5.5.9. When using locally optimal $\sqcup$-strategy iteration for solving the min-max-system of Example 5.5.7, the initial strategy σ_0 maps every $\sqcup$-variable, i.e., node belonging to player 0, to the constant monomial $(0,0,0)$. From this we obtain the first approximation $\nu\boldsymbol{F}\mathcal{A}_{\sigma_0}$ with $\left(\nu\boldsymbol{F}\mathcal{A}_{\sigma_0}\right)_U = (0,1,0)$, $\left(\nu\boldsymbol{F}\mathcal{A}_{\sigma_0}\right)_Y = (0,0,1)$, and $\left(\nu\boldsymbol{F}\mathcal{A}_{\sigma_0}\right)_S = (0,0,0)$ for all $S \in V_0$. Evaluating $\boldsymbol{F}\mathcal{A}$ at $\nu\boldsymbol{F}\mathcal{A}_{\sigma_0}$ yields:

$$\nu\boldsymbol{F}\mathcal{A}_{\sigma_0} = \begin{pmatrix} (0,1,0) \\ (0,0,0) \\ (0,0,0) \\ (0,0,0) \\ (0,0,1) \\ (0,0,0) \end{pmatrix} \sqsubseteq \begin{pmatrix} (0,1,0) \\ (0,0,0) \\ (1,1,0) \\ (0,1,0) \\ (0,0,1) \\ (0,0,0) \end{pmatrix} = \boldsymbol{F}\mathcal{A}(\nu\boldsymbol{F}\mathcal{A}_{\sigma_0}).$$

This means that we change σ_0 at the variables W and X. For X, there is only one monomial so the change is unique. In the case of W, the monomial $(1,0,0) + Y$ does not lead to an improvement, so we obtain the strategy σ_1 yielding the system $\boldsymbol{F}\mathcal{A}_{\sigma_1}$ with the arena $\mathcal{A}$ restricted by the strategy σ_1 shown on the right:

[10] The weight of a simple cycle in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$ is the vector $\boldsymbol{x} \in \mathbb{N}^d$ whose i^{th} component is the number of times how often color i appears along the cycle. By definition of $\sqsubseteq$ in the case of $\mathcal{P}_d$, no cycle in $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$ therefore has weight $\mathbf{0}$, and the weight of a cycle is greater than $\mathbf{0}$ if and only if the highest color appearing along the cycle is even.

$$
\begin{array}{lcl}
\left(\boldsymbol{F}\mathcal{A}_{\sigma_1}\right)_U &=& (0,1,0) + V \sqcap (0,1,0) + W \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_1}\right)_V &=& (0,0,0) \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_1}\right)_W &=& (1,0,0) + U \sqcup (0,0,0) \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_1}\right)_X &=& (0,1,0) + W \sqcup (0,0,0) \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_1}\right)_Y &=& (0,0,1) + X \sqcap (0,0,1) + Z \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_1}\right)_Z &=& (0,0,0).
\end{array}
$$

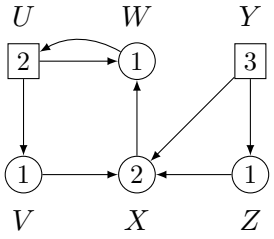


We obtain $\nu\boldsymbol{F}\mathcal{A}_{\sigma_1}$ by calculating $\boldsymbol{F}\mathcal{A}^6_{\sigma_1}(\boldsymbol{\infty})$ leading to

$$
\nu\boldsymbol{F}\mathcal{A}_{\sigma_1} = \begin{pmatrix} (0,1,0) \\ (0,0,0) \\ (1,1,0) \\ (1,2,0) \\ (0,0,1) \\ (0,0,0) \end{pmatrix} \sqsubseteq \begin{pmatrix} (0,1,0) \\ (2,2,0) \\ (1,1,0) \\ (1,2,0) \\ (0,0,1) \\ (2,2,0) \end{pmatrix} = \boldsymbol{F}\mathcal{A}(\nu\boldsymbol{F}\mathcal{A}_{\sigma_1}).
$$

Evaluating $\boldsymbol{F}\mathcal{A}$ at the new approximation, one checks that we now have to adapt the strategy for V and Z. In both cases there is only a single choice resulting in the strategy σ_2 with

$$
\begin{array}{lcl}
\left(\boldsymbol{F}\mathcal{A}_{\sigma_2}\right)_U &=& (0,1,0) + V \sqcap (0,1,0) + W \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_2}\right)_V &=& (1,0,0) + W \sqcup (0,0,0) \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_2}\right)_W &=& (1,0,0) + U \sqcup (0,0,0) \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_2}\right)_X &=& (0,1,0) + W \sqcup (0,0,0) \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_2}\right)_Y &=& (0,0,1) + X \sqcap (0,0,1) + Z \\
\left(\boldsymbol{F}\mathcal{A}_{\sigma_2}\right)_Z &=& (1,0,0) + X \sqcup (0,0,0).
\end{array}
$$



Finally, we obtain $\left(\nu\boldsymbol{F}\mathcal{A}_{\sigma_3}\right)_S = \infty$ for all variables S. Thus, player 0 wins all nodes by means of strategy σ_2, i.e., by disabling the edge from W to Y. ◇

Note that every step of the locally optimal strategy iteration consists of calculating $\boldsymbol{F}\mathcal{A}^n_{\sigma_i}(\boldsymbol{\infty})$ with σ_i some strategy along the strategy iteration, and $n = |V|$. Obviously, this takes at most $O(|E| \cdot |V|)$ many operations, i.e., time polynomial in the size of the arena $\mathcal{A}$.

5.6 Discussion and Related Work

The results of this chapter are based on several works by other authors: We started from the basic idea of $\sqcup$-strategy iteration as proposed by Gawlitza and Seidl in [GS08] for min-max-systems over the integers, and lifted their approach to the setting of si-semiring hoping to obtain a better understanding of when strategy iteration can be applied. We further extended their

approach to nondeterministic strategies. We gave a characterization of the strategies of a strategy iteration sequence and used this characterization to show that the least fixed point of any min-max-system on a si-semiring exists.

The more general class of nondeterministic strategies allowed us to introduce locally optimal $\sqcup$-strategy iteration. This idea was motivated by the work done by Schewe. In [Sch07] he describes how one can calculate an optimal update ([11]) of an *estimation* of the optimal play values. W.r.t. Section 5.4, an estimation basically corresponds to an underapproximation $\nu\boldsymbol{F}_{\sigma_i}$, while the optimal update corresponds to taking I_{σ_i} as successor strategy.

The algorithm of [Sch07] is based on the idea by Björklund, Sandberg and Vorobyov [BSV03, BSV04] to give one player of a mean payoff ([12]), resp. parity game the possibility of escaping an infinite cycle. W.r.t. Section 5.4 this option of escape corresponds to the node $\blacklozenge$ in the game played on $\mathcal{G}^{\blacklozenge}_{\boldsymbol{F}\mathcal{A}}$. Recall that the node $\blacklozenge$ was used to encode the constants appearing in a *clean* min-max-system. The introduction of an escape in a parity game therefore causes the underlying min-max-system $\boldsymbol{F}\mathcal{A}$ to become clean.

To summarize, the $\sqcup$-strategy iteration described in this chapter unifies and generalizes the strategy iteration approaches underlying the algorithms of [GS08] and [BSV04], and allows to directly apply the main idea of [Sch07]. In [Lut08] it is shown that the instantiation of our results to parity games (Section 5.5) also allows for a direct comparison with the algorithm by Jurdzinski and Vöge [VJ00].

[11] Schewe shows how to calculate the optimal update via Dijkstra's algorithm. It is left for future work to check if his approach can be applied to the locally optimal $\sqcup$-strategy iteration.

[12] Informally, a mean payoff game is a linear min-max-system on the integers. We refer the reader to [BSV04] for more details.

Chapter 6

Geometrical Properties of Newton's Method

6.1 Introduction

In this chapter, we discuss two questions that arise naturally when considering polynomial systems on the nonnegative reals.

The first question has already been motivated in Example 1.1.8. There, we considered the system

$$\begin{array}{rcrcl} X_1 & = & \boldsymbol{f}_1(X_1, X_2) & := & \frac{1}{2}X_1^2 + \frac{1}{4}X_2^2 + \frac{1}{4} \\ X_2 & = & \boldsymbol{f}_2(X_1, X_2) & := & \frac{1}{4}X_1 + \frac{1}{4}X_1X_2 + \frac{1}{4}X_2^2 + \frac{1}{4}. \end{array}$$

on the nonnegative reals. We visualized $\boldsymbol{f}$ by drawing the surfaces implicitly defined by $\boldsymbol{q}_1(X_1, X_2) := \boldsymbol{f}_1 - X_1 = 0$, resp. $\boldsymbol{q}_2(X_1, X_2) := \boldsymbol{f}_2 - X_2 = 0$ into the same coordinate system so that $\mu\boldsymbol{f}$ became the least nonnegative point of $\mathbb{R}^2$ common to both surfaces (see Figure 6.1). The shape of the two surfaces then suggested the following approach for approximating $\mu\boldsymbol{f}$: Given some point $\boldsymbol{x}$ located in the region enclosed by the coordinate axes and the two surfaces, we can obtain a better approximation of $\mu\boldsymbol{f}$ by moving from $\boldsymbol{x}$ to the surfaces parallel to the axes (points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$), and taking the tangents at these in the respective points as an approximation of actual surfaces. The intersection $\boldsymbol{y}$ of these tangents then should yield an improved approximation of $\mu\boldsymbol{f}$.

We called this approach the *tangent method*, and claimed that it converges at least as fast as Newton's method to the least nonnegative solution of a polynomial system on the nonnegative reals. We prove this claim in Section 6.3.

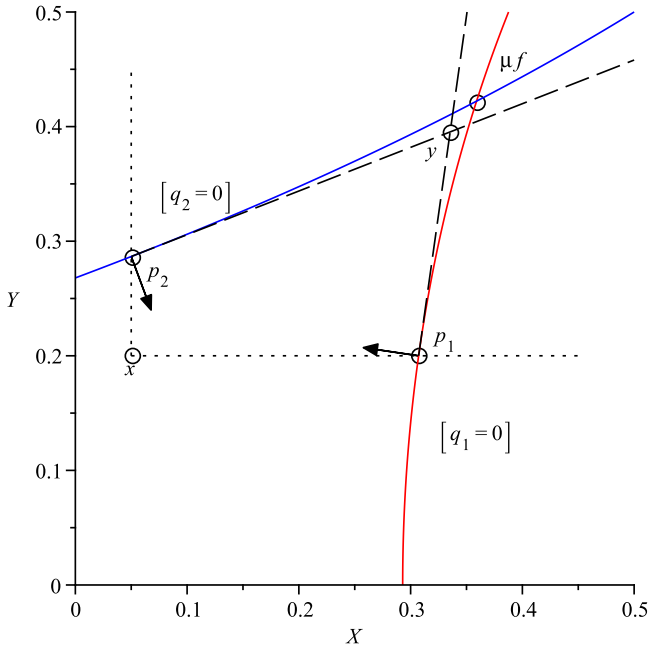


Figure 6.1: The tangent method. Given $\boldsymbol{x} \in R^{\boldsymbol{f}}$ we move to the points $\boldsymbol{p}_1$, and $\boldsymbol{p}_2$ located on the surfaces defined by $[\boldsymbol{q}_1 = 0]$, resp. $[\boldsymbol{q}_2 = 0]$. Then the intersection of the tangent planes in these points is taken as new approximation.

The second question can also be motivated by Figure 6.1. Again, when looking at the shape of the two surfaces, more precisely at their curvature, one gets the impression that there should be a second nonnegative intersection. This question is important in the context of Galton-Watson processes [WG75]. We give a description of Galton-Watson processes motivated by parallel programs:
Assume we have a finite set $\mathcal{X} = \{X_1, \ldots, X_n\}$ of task types. Every task takes exactly one time unit to execute, independent of its type. At termination the task then randomly generates a finite number of new tasks all running in parallel. More precisely, we associate with every type X_i a random variable (or random vector) Z_i where Z_i ranges over $\mathbb{N}_0^n$, and $\Pr[Z_i = (k_1, \ldots, k_n)]$ is the probability that a task of type X_i generates at termination k_l children of type X_l for $l \in [n]$. Initially, we have exactly one task of type X_1. Natural questions arising are whether such a parallel program eventually terminates, and what the expected time till termination is. Both questions are connected to the probability generating functions associated with the process:

$$\boldsymbol{f}_i(X_1, \ldots, X_n) := \sum_{(k_1,\ldots,k_n)\in\mathbb{N}_0^n} X_1^{k_1} \cdot X_2^{k_2} \cdot \ldots \cdot X_n^{k_n} \cdot \Pr[Z_i = (k_1, \ldots, k_n)],$$

where we use $\mathcal{X}$ also as the set of variables. This yields a system $\boldsymbol{f}$ of power series on the nonnegative reals. In particular, for a parallel program it is a sensible assumption that all random variables Z_i have a finite range, so that $\boldsymbol{f}$ becomes a polynomial system. We will assume in the following that $\boldsymbol{f}$ is a polynomial system.

In the setting of traditional Galton-Watson processes, one only considers a single type. Then $\boldsymbol{f}$ becomes a univariate polynomial $f(X)$ which is the probability generating function of the random variable describing the number child tasks generated. It is well-known that $f'(1)$ is then the expected number of tasks spawned by a single task at termination. The analysis of the Galton-Watson process is then split up into the critical case ($f'(1) = 1$) where the program terminates almost surely, although the expected time is unbounded; the subcritical case ($f'(1) < 1$) where the program terminates almost surely with finite expected running time; and the supercritical case ($f'(1) > 1$) where the program does not terminate almost surely, nor is its expected running time finite.

Of particular importance in the analysis of the subcritical (and also the supercritical) case (see [Lin76, Ner77]) is the existence of a second fixed point a of $f(X)$ greater than one ([1]). In Section 6.4, we study the more general question of the existence of a second fixed point of a polynomial equation systems on the nonnegative reals if the surfaces intersect in $\mu\boldsymbol{f}$.

6.2 Preliminaries

In the following, let $\boldsymbol{f}$ be a polynomial system on the nonnegative reals. Every polynomial of $\boldsymbol{f}_X$ is then a *positive polynomial*, i.e., all its coefficients are positive. In accordance with [EKL09a], we call such systems *systems of positive polynomials*, or short *SPP*. We assume that the variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ are used, i.e., $n := |\mathcal{X}|$, and write $\boldsymbol{X}$ for the column vector $(X_1, \ldots, X_n)^\top$. We also identify $\mathbb{R}^{\mathcal{X}}_{\geq 0}$ with $\mathbb{R}^n_{\geq 0}$. Instead of $\boldsymbol{f}_{X_i}$ we then simply write $\boldsymbol{f}_i$, and refer to it as the i-th component of $\boldsymbol{f}$. That is, $\boldsymbol{f}$ is interpreted as a (column) vector of polynomials in the canonical way. Then $\boldsymbol{f}$ is a map from $\mathbb{R}^n_{\geq 0}$ to $\mathbb{R}^n_{\geq 0}$. We may extend this to $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$. We call an SPP $\boldsymbol{f}$ *feasible* if $\mu\boldsymbol{f}$ exists in $\mathbb{R}^n_{\geq 0}$. It is well-known that any SPP $\boldsymbol{f}$ can be decomposed into strongly connected subsystems, i.e., the calculation of $\mu\boldsymbol{f}$ for a general SPP $\boldsymbol{f}$ can be reduced to the case that $\boldsymbol{f}$ is strongly-connected

[1] As $f'(1) < 1$ holds, the function $f(X) - X$ becomes negative on some interval $(1, 1+\varepsilon)$. By continuity, and $\lim_{t\to\infty} f(t) = \infty$, there has to be an $a > 1$ with $f(a) = a$.

(cf. [EY06]). We abbreviate $\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}$ by $\boldsymbol{q}$, and denote by $[\boldsymbol{q}_i = 0]$ the set $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{q}_i(\boldsymbol{x}) = 0\}$, i.e., $[\boldsymbol{q}_i = 0]$ is the surface associated with $\boldsymbol{f}_i$.

We introduce some additional notation:
We write $\mathbf{0}$ for the vector of $\mathbb{R}^n$ which is equal to zero in every component. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ we write $\boldsymbol{x} \leq \boldsymbol{y}$ if $\boldsymbol{x}$ is less than or equal to $\boldsymbol{y}$ in every component, i.e., $\forall i \in [n] : \boldsymbol{x}_i \leq \boldsymbol{y}_i$ (with $[n] := \{1, 2, \ldots, n\}$). Then $\boldsymbol{x} < \boldsymbol{y}$ holds, if $\boldsymbol{x} \neq \boldsymbol{y}$ and $\boldsymbol{x} \leq \boldsymbol{y}$. If $\boldsymbol{x}$ is less than $\boldsymbol{y}$ in every component, i.e., $\forall i \in [n] : \boldsymbol{x}_i < \boldsymbol{y}_i$, we write $\boldsymbol{x} \prec \boldsymbol{y}$. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ with $\boldsymbol{x} \leq \boldsymbol{y}$ we let $[\boldsymbol{x}, \boldsymbol{y}]$ denote the set $\{\boldsymbol{z} \in \mathbb{R}^n \mid \boldsymbol{x} \leq \boldsymbol{z} \leq \boldsymbol{y}\}$. By $\boldsymbol{x}_{-i}$ we denote the vector we obtain from $\boldsymbol{x} \in \mathbb{R}^n$ by removing the i-th component, i.e., $\boldsymbol{x}_{-i} \in \mathbb{R}^{n-1}$. Given some $y \in \mathbb{R}$ we then write $(\boldsymbol{x}_{-i}; y)$ for the vector we obtain from $\boldsymbol{x}$ by setting its i-th component to the value y.

We further use standard notation for derivatives in this chapter, as we are on the reals. We write $\partial_i g$ for the partial derivative w.r.t. the variable X_i of a function $g : \mathbb{R}^n \to R$ given in the variables $\mathcal{X}$. The evaluation of the partial derivative at some point $\boldsymbol{x} \in \mathbb{R}^n$ is then denoted by $\partial_X g|_{\boldsymbol{x}}$. The gradient of g is then the row vector ∇g:

$$\nabla g := (\partial_1 g, \partial_2 g, \ldots, \partial_n g).$$

Similarly, for $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$, we then denote the Jacobian of $\boldsymbol{g}$ by $J_{\boldsymbol{g}}$:

$$J_{\boldsymbol{g}} := \begin{pmatrix} \partial_1 \boldsymbol{g}_1 & \partial_2 \boldsymbol{g}_1 & \ldots & \partial_n \boldsymbol{g}_1 \\ \vdots & & & \\ \partial_1 \boldsymbol{g}_n & \partial_2 \boldsymbol{g}_n & \ldots & \partial_n \boldsymbol{g}_n \end{pmatrix},$$

i.e., the i-th row of the matrix $J_{\boldsymbol{g}}$ is the gradient of $\boldsymbol{g}_i$. Again, we write $\nabla g|_{\boldsymbol{x}}$, resp. $J_{\boldsymbol{g}}|_{\boldsymbol{x}}$ for the evaluation of the resp. objective at some point $\boldsymbol{x} \in \mathbb{R}^n$. W.r.t. Definition 3.1.7, we then have $D\boldsymbol{f}|_{\boldsymbol{x}}(\boldsymbol{X}) = J_{\boldsymbol{f}}|_{\boldsymbol{x}} \cdot \boldsymbol{X}$ using the identification of $\mathbb{R}^{\mathcal{X}}$ with $\mathbb{R}^n$ described above.

6.3 The Tangent Method

As already stated, we assume that we are given some point located within the region delimited by the coordinate axes and the surfaces $[\boldsymbol{q}_i(\boldsymbol{X}) = 0]$ for $i \in [n]$. We call this region $R^{\boldsymbol{f}}$ in the following. It was shown in [EKL09a] that this region can be characterized as follows:

Definition 6.3.1.
For a feasible SPP $\boldsymbol{f}$ we set $R^{\boldsymbol{f}} := \{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \mid \boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x}) \wedge \boldsymbol{x} \leq \mu \boldsymbol{f}\}$. ◇

We will see that all approximants obtained by Newton's method, or the tangent method, are indeed located in $R^{\boldsymbol{f}}$. As a side note, although Figure 6.1 might convey the impression that $R^{\boldsymbol{f}}$ is always convex, this is in general not the case.

Example 6.3.2. Consider the SSP $\boldsymbol{f}$ with

$$\boldsymbol{f}_i(\boldsymbol{X}) := \frac{1}{16}X_j^2 + \frac{1}{4}X_jX_k + \frac{1}{16}X_k^2 + \frac{5}{8}$$

for $\{i,j,k\} = [3]$. This SPP is clean, feasible and strongly-connected with $\mu\boldsymbol{f} = (1,1,1)^\top$.

Each of the equations $\boldsymbol{f}_i(\boldsymbol{X}) = \boldsymbol{X}_i$ then defines a hyperbolic paraboloid. Using standard techniques from differential geometry, one easily checks that $R^{\boldsymbol{f}}$ is not convex. ◇

We first show that for any $\boldsymbol{x} \in R^{\boldsymbol{f}}$ we can always move to the surfaces as already sketched.

Lemma 6.3.3.
Let $\boldsymbol{f}$ be a feasible SPP and $\boldsymbol{x} \in R^{\boldsymbol{f}}$. Set

$$\boldsymbol{\delta}^{\boldsymbol{x}}(t) := \boldsymbol{f}(\boldsymbol{x} + t \cdot \boldsymbol{e}^i) - \boldsymbol{x}.$$

Then $\mu\big(\boldsymbol{\delta}_i^{\boldsymbol{x}}\big)$ exists for every $i \in [n]$. By a slight abuse of notation, we write $\mu\boldsymbol{\delta}^{\boldsymbol{x}}$ for the vector with components $\mu\big(\boldsymbol{\delta}_i^{\boldsymbol{x}}\big)$. We then have $\mu\boldsymbol{\delta}^{\boldsymbol{x}} \leq \mu\boldsymbol{f} - \boldsymbol{x}$, and the ray $\boldsymbol{x} + \mathbb{R}_{\geq 0} \cdot \boldsymbol{e}^i$ intersects $[\boldsymbol{q}_i = 0]$ in the point $\boldsymbol{x} + \mu\boldsymbol{\delta}_i^{\boldsymbol{x}} \cdot \boldsymbol{e}^i$ for the first time. In particular,

$$\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i) > 0 \text{ and } \partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i} < 0$$

holds for all $t \in [0, \mu\boldsymbol{\delta}_i^{\boldsymbol{x}})$. ◇

Proof. Fix some $i \in [n]$ and set $d(t) := \boldsymbol{\delta}_i^{\boldsymbol{x}}(t)$. We have for all $t \in \mathbb{R}_{\geq 0}$

$$d(t) = \boldsymbol{f}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i) - \boldsymbol{x}_i \leq \boldsymbol{f}_i(\mu\boldsymbol{f} + t \cdot \boldsymbol{e}^i) - \boldsymbol{x}_i$$

as $\boldsymbol{x} \leq \mu\boldsymbol{f}$. By definition of R, we further have $\boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x})$, and so

$$0 \leq d(0) \leq \boldsymbol{f}_i(\mu\boldsymbol{f}) - \boldsymbol{x}_i = \mu\boldsymbol{f}_i - \boldsymbol{x}_i$$

follows. From this we obtain via induction for all $k \in \mathbb{N}$ and $i \in [n]$:

$$\begin{array}{rcl} d^{k+1}(0) & = & d(d^k(0)) \\ & \leq & \boldsymbol{f}_i(\boldsymbol{x} + d^k(0) \cdot \boldsymbol{e}^i) - \boldsymbol{x}_i \\ & \leq & \boldsymbol{f}_i(\boldsymbol{x} + (\mu\boldsymbol{f}_i - x_i) \cdot \boldsymbol{e}^i) - \boldsymbol{x}_i \\ & \leq & \boldsymbol{f}_i(\mu\boldsymbol{f}) - \boldsymbol{x}_i \\ & \leq & \mu\boldsymbol{f}_i - \boldsymbol{x}_i. \end{array}$$

By monotonicity of d, μd exists with $\mu d \leq \mu\boldsymbol{f}_i - \boldsymbol{x}_i$. Then the ray $\boldsymbol{x} + \mathbb{R}_{\geq 0} \cdot \boldsymbol{e}^i$ indeed intersects the surface defined by $\boldsymbol{f}_i(\boldsymbol{X}) = X_i$ in $\boldsymbol{x} + \mu d \cdot \boldsymbol{e}^i$.

If $0 < \mu d$, we have $t < d(t)$ for all $t \in [0, \mu d)$ as μd is the least nonnegative fixed point of $d(t)$. So,

$$0 < d(t) - t = \boldsymbol{f}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i) - (\boldsymbol{x}_i + t) = \boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)$$

follows for all $t \in [0, \mu d)$. Finally, we have

$$\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i) = \boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{q}_i|_{\boldsymbol{x}} \cdot (t \cdot \boldsymbol{e}^i) + r(t)$$

where r is the positive polynomial consisting of all monomials of $\boldsymbol{q}_i(\boldsymbol{x}+t\cdot\boldsymbol{e}^i)$ which depend at least quadratically on t. Again, if $0 < \mu d$, we have $\boldsymbol{q}_i(\boldsymbol{x}) > 0$, further for all $t \in \mathbb{R}_{\geq 0}$ it holds that $r(t) \geq 0$. So, as $\boldsymbol{q}_i(\boldsymbol{x} + \mu d \cdot \boldsymbol{e}^i) = 0$, it follows that

$$0 > \nabla \boldsymbol{q}_i|_{\boldsymbol{x}} \boldsymbol{e}^i = \partial_i \boldsymbol{q}_i|_{\boldsymbol{x}}.$$

As $\boldsymbol{x}$ was chosen arbitrarily from $R^{\boldsymbol{f}}$ except for the assumption that $\boldsymbol{q}_i(\boldsymbol{x}) > 0$, we conclude that for all $t \in [0, \mu d)$:

$$\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i} < 0. \qquad \square$$

Definition 6.3.4.
For $\boldsymbol{f}$ feasible and $\boldsymbol{x} \in R^{\boldsymbol{f}}$, let $\mu\boldsymbol{\delta}^{\boldsymbol{x}}$ be as defined in the preceding Lemma 6.3.3. We define the *height* of the i-th surface by $\boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x}) := \boldsymbol{x}_i + \mu\boldsymbol{\delta}_i^{\boldsymbol{x}}$. Similarly, $\boldsymbol{p}^i(\boldsymbol{x}) := \boldsymbol{x} + \boldsymbol{e}^i \cdot \mu\boldsymbol{\delta}_i^{\boldsymbol{x}}$ is the point on the i-th surface satisfying $\boldsymbol{p}^i_{-i}(\boldsymbol{x}) = \boldsymbol{x}_{-i}$. $\diamond$

Example 6.3.5. Consider the system $\boldsymbol{f}$ depicted in Figure 6.1:

$$\begin{array}{rcl} \boldsymbol{q}_1(\boldsymbol{X}) & := & \frac{1}{2}X_1^2 + \frac{1}{4}X_2^2 + \frac{1}{4} - X_1 \\ \boldsymbol{q}_2(\boldsymbol{X}) & := & \frac{1}{4}X_1 + \frac{1}{4}X_1X_2 + \frac{1}{4}X_2^2 + \frac{1}{4} - X_2. \end{array}$$

Here, we may solve $\boldsymbol{q}_i(\boldsymbol{X}) = 0$ directly for X_i. This yields

$$\begin{array}{rcl} \boldsymbol{h}_1^{\boldsymbol{f}}(\boldsymbol{X}) & = & 1 - \sqrt{\frac{1}{2} - \frac{1}{2} \cdot X_2^2} \\ \boldsymbol{h}_2^{\boldsymbol{f}}(\boldsymbol{X}) & = & 2 - \frac{1}{2}X - \frac{1}{2}\sqrt{X^2 - 12X + 12}. \end{array}$$

$\diamond$

As the components of $\boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})$ might be irrational, calculating $\boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})$ is not possible in general, and, thus, we have to consider the slightly more general case where we do not use the actual tangents, but rather approximate them.

Recall that for $\boldsymbol{x} \in [\boldsymbol{q}_i = 0]$ the tangent in $\boldsymbol{x}$ is given by the equation

$$\nabla \boldsymbol{q}_i|_{\boldsymbol{x}} \cdot (\boldsymbol{X} - \boldsymbol{x}) = 0.$$

If $\boldsymbol{x}$ is located beneath the surface ($\boldsymbol{x}_i \leq \boldsymbol{q}_i(\boldsymbol{x})$), then – as we will see – we can approximate the tangent in $\boldsymbol{p}_i(\boldsymbol{x})$ by

$$\boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{q}_i|_{\boldsymbol{x}} \cdot (\boldsymbol{X} - \boldsymbol{x}) = 0.$$

Intuitively, the addend $\boldsymbol{q}_i(\boldsymbol{x})$ moves the tangent in $\boldsymbol{x}$ at the surface $[\boldsymbol{q}_i = \boldsymbol{q}_i(\boldsymbol{x})]$ towards the actual surface $[\boldsymbol{q}_i = 0]$, as $\nabla \boldsymbol{q}_i|_{\boldsymbol{x}}$ points into the interior of $R^{\boldsymbol{f}}$. This motivates the following definition:

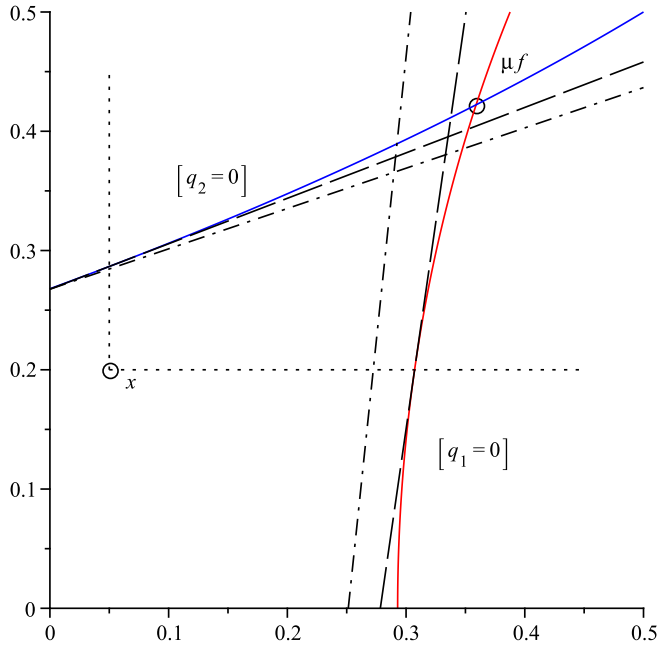


Figure 6.2: This figure shows the (approximated) tangent planes used for approximating the surfaces $[\boldsymbol{q}_i = 0]$ for $\boldsymbol{\alpha}^i = \boldsymbol{p}^i(\boldsymbol{x})$ (drawn by dashes), resp. for $\boldsymbol{\alpha}^i = \boldsymbol{x}$ (drawn by dots). The closer $\boldsymbol{\alpha}_i$ gets to $\boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$, the better $\boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{y} - \boldsymbol{\alpha}^i) = 0$ approximates the actual tangent.

Definition 6.3.6.
For $\boldsymbol{f}$ a feasible SPP, $\boldsymbol{x} \in R^{\boldsymbol{f}}$, and $\boldsymbol{\alpha} \in [\boldsymbol{x}, \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})]$, let $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ denote the following linear system (with $\boldsymbol{\alpha}^i := (\boldsymbol{x}_{-i}; \boldsymbol{\alpha}_i)$):

$$\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X}) := \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}^i) + \nabla \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} \cdot \boldsymbol{X} \qquad \diamond$$

Note that $\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}} - X_i = 0$ is exactly the (approximated) tangent in $\boldsymbol{\alpha}^i$ with the origin of the coordinate system moved into $\boldsymbol{\alpha}$. A fixed point of $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ is thus an intersection of all (approximated) tangents. See Figure 6.2 for an example.

Lemma 6.3.7.
Under the requirements of Definition 6.3.6 $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ is a feasible SPP with $\boldsymbol{\alpha} + \mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} \leq \mu \boldsymbol{f}$. $\diamond$

Proof. As the i-th component of $\boldsymbol{\alpha} - \boldsymbol{\alpha}^i$ is zero, $\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}$ is indeed a positive polynomial. We show that $\mu \boldsymbol{f} - \boldsymbol{\alpha}$ is a prefixed point. For this, recall that we have (Taylor expansion)

$$\boldsymbol{q}_i(\boldsymbol{X}) = \boldsymbol{q}_i((\boldsymbol{X} - \boldsymbol{\alpha}^i) + \boldsymbol{\alpha}^i) = \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}_i} \cdot (\boldsymbol{X} - \boldsymbol{\alpha}^i) + r(\boldsymbol{X} - \boldsymbol{\alpha}^i)$$

with $r(\boldsymbol{Y})$ a positive polynomial, as $\boldsymbol{q} = \boldsymbol{f} - \boldsymbol{X}$ only has negative terms of order one. With this at hand, and as $\boldsymbol{\alpha} \leq \mu\boldsymbol{f}$, we obtain:

$$\begin{aligned} \boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu\boldsymbol{f} - \boldsymbol{\alpha}) &= \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla\boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}^i) + \nabla\boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} \cdot (\mu\boldsymbol{f} - \boldsymbol{\alpha}) \\ &= \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla\boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\mu\boldsymbol{f} - \boldsymbol{\alpha}^i) + (\mu\boldsymbol{f}_i - \boldsymbol{\alpha}_i) \\ &\leq \boldsymbol{q}_i(\mu\boldsymbol{f}) + \mu\boldsymbol{f}_i - \boldsymbol{\alpha}_i \\ &= \mu\boldsymbol{f}_i - \boldsymbol{\alpha}_i. \end{aligned}$$

So, the Kleene sequence of $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ is bounded from above by $\mu\boldsymbol{f} - \boldsymbol{\alpha}$, i.e., $\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} \leq \mu\boldsymbol{f} - \boldsymbol{\alpha}$. □

Definition 6.3.8.
Under the requirements of Definition 6.3.6 we define the *tangent operator* by $\mathcal{T}(\boldsymbol{x};\boldsymbol{\alpha}) := \boldsymbol{\alpha} + \mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$. Further, set $\mathcal{N}(\boldsymbol{x}) := \mathcal{T}(\boldsymbol{x};\boldsymbol{x})$. We denote by $(\boldsymbol{\tau}^{(k)})_{k\in\mathbb{N}}$ any sequence obtained by using $\mathcal{T}(\cdot;\cdot)$ in every step starting from $\boldsymbol{\tau}^{(0)} := \boldsymbol{0}$. ◇

Our goal is to show that any sequence $(\boldsymbol{\tau}^{(k)})_{k\in\mathbb{N}}$ converges at least as fast as Newton's method. For this, we first analyze the properties of $\mathcal{T}(\cdot;\cdot)$ in more detail.

Note that for $\boldsymbol{\alpha} = \boldsymbol{x}$ we have $\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{x}}(\boldsymbol{X}) = \boldsymbol{q}_i(\boldsymbol{x}) + \nabla\boldsymbol{f}_i|_{\boldsymbol{x}} \cdot \boldsymbol{X}$. We can write this succinctly as $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}}(\boldsymbol{X}) = \boldsymbol{q}(\boldsymbol{x}) + J_{\boldsymbol{f}}|_{\boldsymbol{x}} \cdot \boldsymbol{X}$. As $J_{\boldsymbol{q}} = J_{\boldsymbol{f}} - \mathrm{Id}$, we obtain

$$\boldsymbol{X} = \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}}(\boldsymbol{X}) \Leftrightarrow \boldsymbol{q}(\boldsymbol{x}) + J_{\boldsymbol{q}}|_{\boldsymbol{x}} \cdot \boldsymbol{X} = 0.$$

In particular, for $\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}}$ it follows:

$$\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}} = \sum_{k=0}^{\infty} J_{\boldsymbol{f}}|_{\boldsymbol{x}}^k \cdot \boldsymbol{q}(\boldsymbol{x}) =: J_{\boldsymbol{f}}|_{\boldsymbol{x}}^* \cdot \boldsymbol{q}(\boldsymbol{x}).$$

We thus have $\boldsymbol{\nu}^{(k+1)} = \mathcal{N}(\boldsymbol{\nu}^{(k)})$ with $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ as defined in Definition 3.1.9. Note that by Lemma 6.3.7 we only know that this sum exists in $\mathbb{R}^n_{\geq 0}$; but we do not know whether all entries of $J_{\boldsymbol{f}}|_{\boldsymbol{x}}^*$ stay finite, i.e., whether $J_{\boldsymbol{f}}|_{\boldsymbol{x}}^*$ exists in $\mathbb{R}^{n\times n}_{\geq 0}$.

Lemma 6.3.9.
Let $\boldsymbol{f}$ be a clean, feasible, strongly-connected SPP. For $\boldsymbol{x} \leq \boldsymbol{\alpha} \prec \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})$ set

$$M := \begin{pmatrix} \nabla\boldsymbol{f}_1|_{\boldsymbol{\alpha}^1} \\ \vdots \\ \nabla\boldsymbol{f}_n|_{\boldsymbol{\alpha}^n} \end{pmatrix}.$$

Then M^* exists in $\mathbb{R}^{n\times n}_{\geq 0}$ with $M^* = (\mathrm{Id} - M)^{-1}$. In particular, $\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ is the unique solution of the linear equation system $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X}) = \boldsymbol{X}$. ◇

Proof. Recall that $\boldsymbol{\alpha}^i := (\boldsymbol{x}_{-i}; \boldsymbol{\alpha}_i)$, and, hence, $\boldsymbol{x} \leq \boldsymbol{\alpha}^i \leq \boldsymbol{\alpha} \prec \mu \boldsymbol{f}$ for all $i \in [n]$. By virtue of the results obtained by Etessami and Yannakakis in [EY06], we know that $J_{\boldsymbol{q}}|_{\boldsymbol{\alpha}}$ is regular with

$$J_{\boldsymbol{q}}|_{\boldsymbol{\alpha}}^{-1} = -\sum_{k \in \mathbb{N}} J_{\boldsymbol{f}}|_{\boldsymbol{\alpha}}^{k}.$$

This means the inverse of $-J_{\boldsymbol{q}}|_{\boldsymbol{\alpha}}$ is given by the Kleene star of $J_{\boldsymbol{f}}|_{\boldsymbol{\alpha}}$. Note that this matches our definition of the Newton sequence on the semiring of nonnegative reals. The point here is that on the semiring the matrix might have infinite entries, while this result says that in the case of a feasible, clean, strongly-connected SPP this matrix even exists in $\mathbb{R}^n_{\geq 0}$.

As $\boldsymbol{\alpha}^i \leq \boldsymbol{\alpha}$, it follows that $\nabla \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} \leq \nabla \boldsymbol{f}_i|_{\boldsymbol{\alpha}}$ for all $i \in [n]$, i.e., $M \leq J_{\boldsymbol{f}}|_{\boldsymbol{\alpha}}$. We therefore obtain that M^* exists in $\mathbb{R}^n_{\geq 0}$ as $M^* \leq J_{\boldsymbol{f}}|_{\boldsymbol{\alpha}}^*$. □

Note that the requirement that $\boldsymbol{\alpha} \prec \mu \boldsymbol{f}$ is not severe, as in the case that $\boldsymbol{\alpha}_i = \mu \boldsymbol{f}_i$ for some $i \in [n]$, we can simply substitute $\mu \boldsymbol{f}_i$ for X_i in $\boldsymbol{f}$, and consider the reduced system, and its strongly-connected components.

We now show that $\mathcal{T}(\boldsymbol{x}; \boldsymbol{\alpha})$ indeed improves with $\boldsymbol{\alpha} \to \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})$ as suggested by Figure 6.2.

Lemma 6.3.10.
For $\boldsymbol{f}$ feasible, $\boldsymbol{x} \in R^{\boldsymbol{f}}$, and $\boldsymbol{\alpha}, \boldsymbol{\beta}$ with $\boldsymbol{x} \leq \boldsymbol{\alpha} \leq \boldsymbol{\beta} \leq \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})$ we have $\mathcal{T}(\boldsymbol{x}; \boldsymbol{\alpha}) \leq \mathcal{T}(\boldsymbol{x}; \boldsymbol{\beta})$. ◇

Proof. Set $\boldsymbol{d} := \boldsymbol{\alpha} - \boldsymbol{x} \geq \boldsymbol{0}$. We split the proof into two parts. In the first part we introduce an SPP $\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$ and show that $\mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}} = \mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} + \boldsymbol{d}$. We then show in the second part that $\mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}} \leq \mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\beta}}$. As $\mathcal{T}(\boldsymbol{x}; \boldsymbol{\alpha}) = \boldsymbol{\alpha} + \mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ by definition, the claim then follows.

(a) Define $\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$ as follows for $i \in [n]$:

$$\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X}) := \begin{cases} \frac{\boldsymbol{q}_i(\boldsymbol{\alpha}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i}} + (\boldsymbol{\alpha}_i - \boldsymbol{x}_i) + \sum_{k \neq i} \frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i}} \cdot X_k & \text{if } \partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} < 0 \\ \boldsymbol{\alpha}_i - \boldsymbol{x}_i & \text{if } \partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} = 0. \end{cases}$$

The reader can easily check that $\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$ is also an SPP. For $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} < 0$, we have the following relation between $\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}$ and $\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}$:

$$\begin{aligned} & \boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X}) - X_i \\ = \; & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}^i) + \nabla \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} \cdot \boldsymbol{X} - X_i \\ = \; & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot \boldsymbol{X} \\ = \; & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{X} + \boldsymbol{\alpha} - \boldsymbol{\alpha}^i) \\ = \; & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{X} + \boldsymbol{d} + \boldsymbol{x} - \boldsymbol{\alpha}^i) \\ = \; & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{x} - \boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{X} + \boldsymbol{d}) \\ = \; & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) - \partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha}_i - \boldsymbol{x}_i) + \textstyle\sum_{k \neq i} \partial_k \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} \cdot (X_k - \boldsymbol{d}_k) + \partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (X_i - \boldsymbol{d}_i) \\ = \; & -\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot \big(\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X} + \boldsymbol{d}) - (X_i + \boldsymbol{d}_i)\big). \end{aligned} \tag{$*$}$$

We want to show that $\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} + \boldsymbol{d} = \mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$. By equation $(*)$ we have for all $i \in [n]$ with $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} < 0$ that $\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} + \boldsymbol{d}) = (\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_i + \boldsymbol{d}_i$.

Assume therefore that there is an $i \in [n]$ with $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} = 0$. By definition, $\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X})$ is the constant polynomial $\boldsymbol{d}_i = \boldsymbol{\alpha}_i - \boldsymbol{x}_i$, and, thus, $(\mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}})_i = \boldsymbol{d}_i$. We will show in a moment that also $(\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_i = 0$ holds for $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} = 0$. So we also have $\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} + \boldsymbol{d}) = (\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_i + \boldsymbol{d}_i$ for all $i \in [n]$ with $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} = 0$.

We conclude that $\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} + \boldsymbol{d}$ is a fixed point of $\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$. In particular, $\mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$ exists, and it follows that $\mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}} = \mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} + \boldsymbol{d}$ indeed holds.

We turn to the proof of $(\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_i = 0$ for $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} = 0$ or, equivalently, $\partial_i \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} = 1$. By Lemma 6.3.3 we know that $\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} < 0$ if $\boldsymbol{\alpha}^i < \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$. So, $\boldsymbol{\alpha}_i = \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x})$ has to hold, from which $\boldsymbol{q}_i(\boldsymbol{\alpha}^i) = 0$ follows. We therefore obtain:

$$\begin{array}{ll} & \boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{X}) - X_i \\ = & \boldsymbol{q}_i(\boldsymbol{\alpha}^i) + \nabla \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}^i) + \nabla \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} \cdot \boldsymbol{X} - X_i \\ = & \sum_{k \neq i} \partial_k \boldsymbol{f}_i|_{\boldsymbol{\alpha}^i} \cdot (\boldsymbol{\alpha}_k - \boldsymbol{x}_k + X_k) \\ = & \sum_{k \neq i} \partial_k \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} \cdot (X_k + \boldsymbol{d}_k). \end{array} \qquad (**)$$

Let $\widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}$ be the system we obtain by setting $\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}} := 0$. Obviously, $\widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}$ is an SPP with $\widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}(\boldsymbol{x}) \leq \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ and, hence, $\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}} \leq \mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$. Further, as $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$ and $\widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}$ agree for all $k \in [n] \setminus \{i\}$, we have $\big(\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}\big)_k = \boldsymbol{f}_k^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}})$. Assume that $0 = \big(\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}\big)_i < \boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}})$. From $(**)$ it follows that

$$0 < \sum_{k \neq i} \partial_k \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} \cdot \big((\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}})_k + \boldsymbol{d}_k\big).$$

We therefore find a $j \in [n] \setminus \{i\}$ with

$$0 < \partial_j \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} \cdot \big((\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}})_j + \boldsymbol{d}_j\big).$$

On the other hand, by $(**)$ it also follows that

$$0 = \sum_{k \neq i} \partial_k \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} \cdot \big((\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_k + \boldsymbol{d}_k\big), \text{ i.e., } 0 = \partial_j \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} \cdot \big((\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_j + \boldsymbol{d}_j\big).$$

Hence, we have $\partial_j \boldsymbol{f}_i|_{\boldsymbol{\alpha}_i} > 0$ and $\boldsymbol{d}_j = 0$, which yields $\big(\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}\big)_j = 0 < \big(\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}\big)_j$. This contradicts the fact that $\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}} \leq \mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$. Therefore $\mu \widetilde{\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}}$ has to be a fixed point of $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$, in particular the least. So we obtain the result that $(\mu \boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_i = 0$.

(b) It remains to show that $\mathcal{T}(\boldsymbol{x};\boldsymbol{\alpha}) \leq \mathcal{T}(\boldsymbol{x};\boldsymbol{\beta})$. For this we show that $\mu \boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$ increases monotonically with $\boldsymbol{\alpha}$. Again, we argue componentwise, and fix some $i \in [n]$. If $\boldsymbol{x}_i = \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$, there is nothing to show. So, assume $\boldsymbol{x}_i < \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$. By Lemma 6.3.3 we know that $\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}} < 0$ holds. We show that the coefficients of (with $\boldsymbol{\alpha}_i = \boldsymbol{x}_i + t$)

$$\frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x} + t \cdot \boldsymbol{e}^i}} + t + \sum_{k \neq i} \frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{x} + t \cdot \boldsymbol{e}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x} + t \cdot \boldsymbol{e}^i}} \cdot X_k$$

increase with t while $t \in [0, \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x}) - \boldsymbol{x}_i)$. We set $t_0 := \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{x}) - \boldsymbol{x}_i$.

Recall that $\partial_i \boldsymbol{q}_i|_{\boldsymbol{x} + t\boldsymbol{e}^i}$ is negative for $t \in [0, t_0)$, and increases with t. Thus, it remains to show that

$$\frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x} + t \cdot \boldsymbol{e}^i}} + t$$

also increases with t.

As $\boldsymbol{q}(\boldsymbol{X}) = \boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}$, we find a positive polynomial $r(t)$ such that

$$\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i) = \boldsymbol{q}_i(\boldsymbol{x}) + t \cdot \partial_i \boldsymbol{q}_i|_{\boldsymbol{x}} + r(t).$$

From this we obtain

$$\begin{array}{rcl} \partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i} & = & \lim_{h\to 0} \frac{\boldsymbol{q}_i(\boldsymbol{x}+(t+h)\cdot\boldsymbol{e}^i) - \boldsymbol{q}_i(\boldsymbol{x}+t\cdot\boldsymbol{e}^i)}{h} \\ & = & \partial_i \boldsymbol{q}_i|_{\boldsymbol{x}} + r'(t). \end{array}$$

With this at hand, we may write for all $t \in [0, t_0)$:

$$\begin{array}{rcl} & & \frac{d}{dt}\left(\frac{\boldsymbol{q}_i(\boldsymbol{x}+t\cdot\boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + t\right) \\ & = & \frac{\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}} + r'(t)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + \frac{\boldsymbol{q}_i(\boldsymbol{x}+t\cdot\boldsymbol{e}^i)}{(-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i})^2} \cdot r''(t) + 1 \\ & = & \frac{\boldsymbol{q}_i(\boldsymbol{x}+t\cdot\boldsymbol{e}^i)}{(-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i})^2} \cdot r''(t) \\ & \geq & 0. \end{array}$$

So, all coefficients increase monotonically with t.

It remains to consider the limit case if $\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}$ vanishes for $t = t_0$. By Lemma 6.3.7 we have $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu\boldsymbol{f} - \boldsymbol{\alpha}) \leq \mu\boldsymbol{f} - \boldsymbol{\alpha}$. Set $\boldsymbol{D} := \mu\boldsymbol{f} - \boldsymbol{x} \geq \boldsymbol{0}$ so that $\mu\boldsymbol{f} - \boldsymbol{\alpha} = \boldsymbol{D} - \boldsymbol{d}$. From $(*)$ we therefore obtain:

$$\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{D}) - \boldsymbol{D}_i = \frac{\boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}} - \boldsymbol{\alpha}) - \left((\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}})_i - \boldsymbol{\alpha}_i\right)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{\alpha}^i}} \leq 0.$$

So, as long as $\boldsymbol{\alpha}_i = \boldsymbol{x} + t \cdot \boldsymbol{e}^i < \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$, it holds that $\boldsymbol{D}$ is a prefixed point of $\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}$ and, thus, $\mu\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}} \leq \boldsymbol{D}$. Consider again the inequation $\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{D}) \leq \boldsymbol{D}_i$ for $\boldsymbol{\alpha}_i = \boldsymbol{x}_i + t < \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$, i.e.,

$$\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\boldsymbol{D}) = \frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + t + \sum_{k\neq i} \frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} \cdot \boldsymbol{D}_k \leq \boldsymbol{D}_i.$$

As $\boldsymbol{D}$ is constant, this implies in turn that for $t \in [0, t_0)$

$$\sum_{k\neq i} \frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} \cdot \boldsymbol{D}_k = 0.$$

Otherwise this term would go to ∞ for $t \nearrow t_0$, i.e., $\boldsymbol{\alpha}^i \nearrow \boldsymbol{h}_i^{\boldsymbol{f}}(\boldsymbol{x})$. This means that either $\frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} = 0$ if $\boldsymbol{D}_k > 0$, or $\frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} > 0$ if $\boldsymbol{D}_k = 0$. But in the latter case we also have $(\mu\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}})_k = 0$ as $\mu\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}} \leq \boldsymbol{D}$. So we obtain:

$$(\mu\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}})_i = \boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}}(\mu\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}}) = \frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + t + \sum_{k\neq i} \frac{\partial_k \boldsymbol{f}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} \cdot (\mu\boldsymbol{g}^{\boldsymbol{x};\boldsymbol{\alpha}})_k = \frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + t.$$

It therefore suffices to consider the constant term

$$\frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + t.$$

Set $g(t) := \boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)$. Then $g'(t) = \partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}$. In particular, t_0 is a root of both polynomials. Hence, t_0 has to be root of $g(t)$ of multiplicity at least 2. From this we obtain that $\lim_{t \nearrow t_0} \frac{g(t)}{g'(t)} = 0$. So for $t \nearrow t_0$

$$\frac{\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{e}^i)}{-\partial_i \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{e}^i}} + t \nearrow t_0.$$

Note that for $t = t_0$ we have by definition that $\boldsymbol{g}_i^{\boldsymbol{x};\boldsymbol{\alpha}} = t_0$. □

We come to our last result needed for showing that for any sequence $(\boldsymbol{\tau}^{(k)})_{k\in\mathbb{N}}$ we have that $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\tau}^{(k)}$ holds for all $k \in \mathbb{N}$.

Lemma 6.3.11.
For $\boldsymbol{f}$ feasible and $\boldsymbol{x}, \boldsymbol{y} \in R^{\boldsymbol{f}}$ with $\boldsymbol{x} \leq \boldsymbol{y}$ it holds that $\mathcal{N}(\boldsymbol{x}) \leq \mathcal{N}(\boldsymbol{y})$. ◇

Proof. Set $\boldsymbol{d} := \boldsymbol{y} - \boldsymbol{x} \geq \boldsymbol{0}$. We show that $\boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}$ is a prefixed point of $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}}$. For this, fix some $i \in [n]$:

$$\begin{array}{rcl} \boldsymbol{f}_i^{\boldsymbol{x};\boldsymbol{x}}(\boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}) & = & \boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{f}_i|_{\boldsymbol{x}} \cdot (\boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}) \\ & \leq & \boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{f}_i|_{\boldsymbol{y}} \cdot (\boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}) \\ & = & \boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{f}_i|_{\boldsymbol{y}} \cdot \boldsymbol{d} - \boldsymbol{q}_i(\boldsymbol{y}) + \boldsymbol{f}_i^{\boldsymbol{y};\boldsymbol{y}}(\mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}) \\ & = & \boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{q}_i|_{\boldsymbol{y}} \cdot \boldsymbol{d} - \boldsymbol{q}_i(\boldsymbol{y}) + \boldsymbol{d}_i + \mu\boldsymbol{f}_i^{\boldsymbol{y};\boldsymbol{y}} \end{array}$$

So it suffices to show

$$\boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{q}_i|_{\boldsymbol{x}+\boldsymbol{d}} \cdot \boldsymbol{d} - \boldsymbol{q}_i(\boldsymbol{x} + \boldsymbol{d}) \geq 0.$$

Again, we find a positive polynomial $r(t)$ such that

$$\boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{d}) = \boldsymbol{q}_i(\boldsymbol{x}) + t \cdot \nabla \boldsymbol{q}_i|_{\boldsymbol{x}}\boldsymbol{d} + r(t) \text{ and } \nabla \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{d}} = \nabla \boldsymbol{q}_i|_{\boldsymbol{x}}\boldsymbol{d} + r'(t).$$

In particular, we can write $r(t)$ as $t^2 \cdot p(t)$ with $p(t)$ a positive polynomial, too. It follows for all $t \in \mathbb{R}_{\geq 0}$:

$$\begin{array}{rcl} & & \boldsymbol{q}_i(\boldsymbol{x}) + \nabla \boldsymbol{q}_i|_{\boldsymbol{x}+t\cdot\boldsymbol{d}} \cdot (t \cdot \boldsymbol{d}) - \boldsymbol{q}_i(\boldsymbol{x} + t \cdot \boldsymbol{d}) \\ & = & t \cdot r'(t) - r(t) \\ & = & r(t) + t^3 \cdot p'(t) \\ & \geq & 0. \end{array}$$

So, $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}}(\boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}) \leq \boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}$ and $\mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}} \leq \boldsymbol{d} + \mu\boldsymbol{f}^{\boldsymbol{y};\boldsymbol{y}}$ follow. As $\mathcal{N}(\boldsymbol{x}) := \boldsymbol{x} + \mu\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{x}}$, the lemma is shown. □

Theorem 6.3.12.
Let $\boldsymbol{f}$ be feasible SPP. Then for any sequence $(\boldsymbol{\tau}^{(k)})_{k\in\mathbb{N}}$ we have

$$\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\tau}^{(k)} \leq \mu\boldsymbol{f} \text{ for all } k \in \mathbb{N}$$

with $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ the Newton sequence as defined in Definition 3.1.9. ◇

Proof. We proceed by induction on k. By definition we have $\boldsymbol{\nu}^{(0)} = \boldsymbol{\tau}^{(0)} = \boldsymbol{0}$. For $k \geq 0$, and $\boldsymbol{\alpha} \in [\boldsymbol{\tau}^{(k)}, \boldsymbol{h}^{\boldsymbol{f}}(\boldsymbol{\tau}^{(k)})]$ such that $\boldsymbol{\tau}^{(k+1)} = \mathcal{T}(\boldsymbol{\tau}^{(k)}; \boldsymbol{\alpha})$:

$$\boldsymbol{\nu}^{(k+1)} = \mathcal{N}(\boldsymbol{\nu}^{(k)}) \leq \mathcal{N}(\boldsymbol{\tau}^{(k)}) \leq \mathcal{T}(\boldsymbol{\tau}^{(k)}; \boldsymbol{\alpha}) = \boldsymbol{\tau}^{(k+1)}.$$ □

We close this section by noting that the described tangent method should naturally extend itself to the setting of *min-SPPs* [EGKS08]: Similarly to the extension of polynomial systems to min-max-systems on totally ordered semirings, we may consider systems $\boldsymbol{F}$ where every component $\boldsymbol{F}_i$ (with $i \in [n]$) is the minimum of a finite number of polynomials. Such a min-SPP $\boldsymbol{F}$ is again a monotone operator, and it is feasible if $\mu\boldsymbol{F}$ exists in $\mathbb{R}^n_{\geq 0}$.

In order to approximate $\mu\boldsymbol{F}$, for every polynomial $p(\boldsymbol{X})$ appearing in $\boldsymbol{F}_i$, we may obtain an approximation of the surface defined by $p(\boldsymbol{X}) = X_i$. We then define $\boldsymbol{F}_i^{\boldsymbol{x};\alpha}$ to be the minimum of all these approximated tangent planes. In particular, this yields a linear min-SPP, to which the results of [EGKS08] should be applicable. A formal proof is left for future work. The following example sketches the principle idea of this extension:

Example 6.3.13. Consider the min-SPP $\boldsymbol{F}$ given by

$$\begin{array}{lcl} X_1 & = & \boldsymbol{F}_1(\boldsymbol{X}) := \min(\frac{1}{2}X^2 + \frac{1}{4}Y^2 + \frac{1}{4}, 2Y^2 + \frac{1}{4}), \\ X_2 & = & \boldsymbol{F}_2(\boldsymbol{X}) := \min(\frac{1}{4}X + \frac{1}{4}Y^2 + \frac{1}{4}XY, 4X^2 + 2XY + \frac{1}{8}). \end{array}$$

From this min-SPP we obtain the quadrics defined by

$$\begin{array}{lcl} q_1 & = & \frac{1}{2}X^2 + \frac{1}{4}Y^2 + \frac{1}{4} - X, \\ q_2 & = & \frac{1}{4}X + \frac{1}{4}Y^2 + \frac{1}{4}XY - Y, \\ q_3 & = & 2Y^2 + \frac{1}{4} - X, \\ q_4 & = & 4X^2 + 2XY + \frac{1}{8} - Y. \end{array}$$

and depicted in Fig 6.3(a).

Given some point $\boldsymbol{x} \in R^{\boldsymbol{F}} := \{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \mid \boldsymbol{x} \leq \boldsymbol{F}(\boldsymbol{x}) \wedge \boldsymbol{x} \leq \mu\boldsymbol{F}\}$, for every polynomial p appearing in $\boldsymbol{F}_i$ we try to move to the surface $[p - X_i = 0]$ by moving along the ray $\boldsymbol{x} + \mathbb{R}_{\geq 0} \cdot \boldsymbol{e}^i$. As we are now not guaranteed that $\mu\boldsymbol{F}$ is located on $[p - X_i = 0]$, it might happen that $\partial_i(p - X_i)|_{\boldsymbol{x}} > 0$. In this case, Lemma 6.3.3 implies that we may remove the polynomial p from $\boldsymbol{F}_i$ for the further approximation of $\mu\boldsymbol{F}$. On the other hand, if $\partial_i p|_{\boldsymbol{x}} < 1$, then we find an intersection of the ray with the surface, resp. we can approximate this intersection, yielding an approximated tangent of the surface $[p - X_i = 0]$.

Using these approximated tangents, we obtain a linear min-SPP along the lines of the definition of $\boldsymbol{f}^{\boldsymbol{x};\boldsymbol{\alpha}}$. This linear min-SPP basically inscribes into $R^{\boldsymbol{F}}$ a polyhedron approximating $[\boldsymbol{x}, \infty) \cap R^{\boldsymbol{F}}$. This is shown in Figure 6.3(b) using the actual tangents, resp. in Figure 6.3(c) where the approximated tangents of Newton's method are used.

Figures 6.3(b) and (c) suggest that we should try to find in the respective polyhedron the point maximizing the 1-norm. It was shown in [EGKS08] that the least nonnegative fixed-point of a clean linear min-SPP $\boldsymbol{F}_{\mathrm{lin}}$ is indeed characterized by

$$\mu\boldsymbol{F}_{\mathrm{lin}} = \mathrm{argmax}\{\sum_{i=1}^{n} \boldsymbol{x}_i \mid \boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \wedge \boldsymbol{x} \leq \boldsymbol{F}_{\mathrm{lin}}(\boldsymbol{x})\}.$$

One may thus obtain $\mu\boldsymbol{F}_{\mathrm{lin}}$ by solving a linear program.

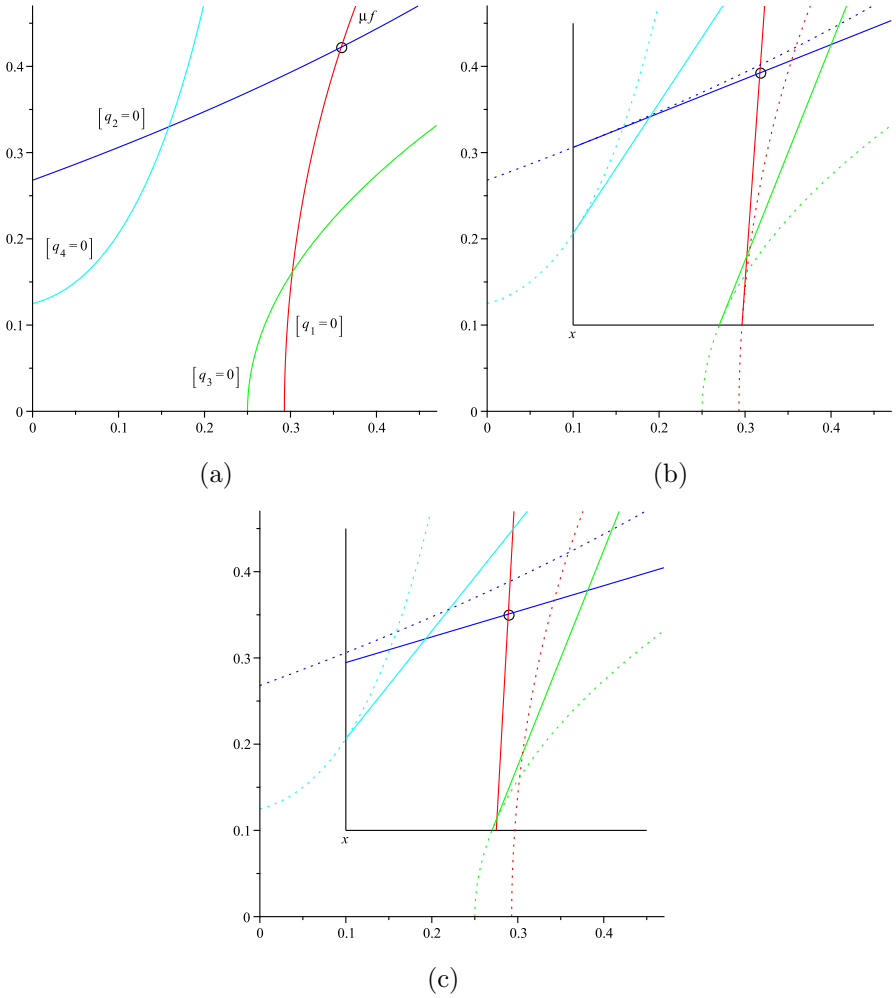


Figure 6.3: The quadrics q_1, q_2, q_3, q_4 from Ex. 6.3.13 and the region $R^{\boldsymbol{F}}$ enclosed by them.

Note that $R^{\boldsymbol{F}}$ is the intersection of all $R^{\boldsymbol{f}}$ with $\boldsymbol{f}$ a feasible SPP obtained from $\boldsymbol{F}$ by selecting for every $i \in [n]$ a polynomial of $\boldsymbol{F}_i$. Hence, the results on $\mathcal{T}(\cdot;\cdot)$ carry over to the setting of min-SPPs. ◇

6.4 Existence of a Second Fixed Point

For the following, let $\boldsymbol{f}$ be a clean, feasible SPP. As previously stated, we denote by $\boldsymbol{q}$ the system $\boldsymbol{f} - \boldsymbol{X}$. In this section, we study the existence of a second fixed point in $\mathbb{R}^n_{\geq 0}$ if the surfaces $[\boldsymbol{q}_i = 0]$ not only touch, but intersect in $\mu\boldsymbol{f}$, i.e., we assume that $J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}$ is regular.

In order to simplify notation, we assume that every polynomial of $\boldsymbol{f}$ is of degree at most two. This is no restriction of generality, as we can always break down a monomial of degree at least three into a system of monomials of degree at most two by introducing auxiliary variables, for instance $Y = X^3$ becomes $Y = XZ \wedge Z = X^2$.

As $\boldsymbol{f}$ is quadratic, we find a $n \times n$ matrix $A^{(i)}$, a n dimensional row-vector $\boldsymbol{b}^{(i)}$ and some constant c_i such that

$$\boldsymbol{f}_i(\boldsymbol{X}) = \boldsymbol{X}^\top A^{(i)} \boldsymbol{X} + \boldsymbol{b}^{(i)} \boldsymbol{X} + c_i.$$

Note that every entry of A_i, resp. $\boldsymbol{b}_i$, resp. c_i itself is nonnegative. Without restriction we may assume that $A^{(i)}$ is symmetric, i.e., $A^{(i)} = \left(A^{(i)}\right)^\top$.

By means of the matrices $A^{(i)}$ we define the bilinear form

$$B(\boldsymbol{X}, \boldsymbol{Y}) := \begin{pmatrix} \boldsymbol{X}^\top A^{(1)} \boldsymbol{Y} \\ \vdots \\ \boldsymbol{X}^\top A^{(n)} \boldsymbol{Y} \end{pmatrix}.$$

Note that $B(\boldsymbol{X}, \boldsymbol{Y}) = B(\boldsymbol{Y}, \boldsymbol{X})$, as we assume that $A^{(i)}$ is symmetric.

Similarly, we can summarize the row-vectors $\boldsymbol{b}^{(i)}$, resp. the constants c_i by means of a matrix, resp. vector

$$L := \begin{pmatrix} \boldsymbol{b}^{(1)} \\ \vdots \\ \boldsymbol{b}^{(n)} \end{pmatrix}, \text{ resp. } \boldsymbol{c} := \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}.$$

We then have

$$\boldsymbol{f}(\boldsymbol{X}) = B(\boldsymbol{X}, \boldsymbol{X}) + L\boldsymbol{X} + \boldsymbol{c}.$$

The Jacobian $J_{\boldsymbol{f}}|_{\boldsymbol{x}}$ of $\boldsymbol{f}$ in some point $\boldsymbol{x}$ can then be written as

$$J_{\boldsymbol{f}}|_{\boldsymbol{x}} := 2B(\boldsymbol{x}, \cdot) + L \text{ with } J_{\boldsymbol{f}}|_{\boldsymbol{x}} \boldsymbol{y} = 2B(\boldsymbol{x}, \boldsymbol{y}) + L\boldsymbol{y}.$$

The following examples show that if $\boldsymbol{f}$ is not strongly-connected, then there exists no second nonnegative fixed point in general.

Example 6.4.1. For $n = 2$ with $\boldsymbol{X} = (X, Y)^\top$ consider the following system:

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{3}X^2 + \frac{2}{3} \\ \frac{1}{4}X^2 + \frac{1}{2}XY + \frac{1}{4} \end{pmatrix}.$$

The Jacobian of $\boldsymbol{f}$ is

$$J_{\boldsymbol{f}} = \begin{pmatrix} \frac{2}{3}X & 0 \\ \frac{1}{2}X + \frac{1}{2}Y & \frac{1}{2}X \end{pmatrix}.$$

Obviously, we have $(J_{\boldsymbol{f}}^k)_{1,2} = 0$ for all $k \in \mathbb{N}$, and so $\boldsymbol{f}$ is not strongly-connected.

It is easily checked that for a fixed point of $\boldsymbol{f}$ we either have to have $X = 1$ or $X = 2$. Solving $\boldsymbol{f}_2(\boldsymbol{X}) = Y$ for Y yields

$$Y(X) = \frac{X^2 + 1}{4 - 2X}.$$

Hence, $\mu\boldsymbol{f} = (1, 1)^\top$ is the only nonnegative fixed point of $\boldsymbol{f}$ in $\mathbb{R}^2$, as $Y(X) \to \infty$ for $X \to 2$. Note that

$$J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}} - \mathrm{Id} = \begin{pmatrix} -\frac{1}{3} & 0 \\ 1 & -\frac{1}{2} \end{pmatrix}$$

has rank 2, i.e., $(J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}} - \mathrm{Id})^{-1}$ exists.

As a second example consider

$$\boldsymbol{g}(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{2}X + \frac{1}{2} \\ \frac{1}{4}X^2 + \frac{1}{2}XY + \frac{1}{4} \end{pmatrix}.$$

Again, $\boldsymbol{g}$ is not strongly-connected. The only fixed point of this system is $(1, 1)$, and $J_{\boldsymbol{g}}|_{(1,1)} - \mathrm{Id}$ is invertible. ◇

We therefore assume in the following that $\boldsymbol{f}$ is strongly-connected. Recall that we require that the Jacobian of $\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}$ evaluated at $\mu\boldsymbol{f}$, i.e., $J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}} - \mathrm{Id}$ with Id the identity matrix, is invertible.

Lemma 6.4.2.
There exists a vector $\boldsymbol{\epsilon} \succ \boldsymbol{0}$ such that for all $\boldsymbol{x} \in [\boldsymbol{0}, \mu\boldsymbol{f} + \boldsymbol{\epsilon})$ we have that $(\mathrm{Id} - J_{\boldsymbol{f}}|_{\boldsymbol{x}})^{-1}$ exists, the spectral radius $\rho(J_{\boldsymbol{f}}|_{\boldsymbol{x}})$ is less than 1, and

$$(\mathrm{Id} - J_{\boldsymbol{f}}|_{\boldsymbol{x}})^{-1} = \sum_{k=0}^{\infty} J_{\boldsymbol{f}}^k|_{\boldsymbol{x}} =: J_{\boldsymbol{f}}|_{\boldsymbol{x}}^*. \qquad \diamond$$

Proof. It was shown by Etessami and Yanakakis in [EY06] that for a strongly-connected system $\boldsymbol{f}$ with $\mu\boldsymbol{f} > \boldsymbol{0}$ this holds for all $\boldsymbol{x} < \mu\boldsymbol{f}$. In particular, it was shown that the spectral radius of $J_{\boldsymbol{f}}|_{\boldsymbol{x}}$ is less than 1 for all $\boldsymbol{x} < \mu\boldsymbol{f}$. Let $\rho(\boldsymbol{x})$ denote the spectral radius of $J_{\boldsymbol{f}}|_{\boldsymbol{x}}$.

As $\boldsymbol{f}$ is assumed to be strongly-connected and $\mu\boldsymbol{f} \succ \boldsymbol{0}$, we have $J_{\boldsymbol{f}}^k|_{\mu\boldsymbol{f}} \succ \boldsymbol{0}$ for some $k > 0$. So, by the Perron-Frobenius theorem, the spectral radius ρ of $J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}}$ is an Eigenvalue of $J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}}$ having algebraic and geometric multiplicity one. As the Eigenvalues are continuous w.r.t. to the coefficients of $J_{\boldsymbol{f}}|_{\boldsymbol{X}}$, and, thus, continuous in $\boldsymbol{X}$, there exists an open ball B centered in $\mu\boldsymbol{f}$ such that $\rho(\cdot)$ is continuous when restricted to B.

Let $\boldsymbol{v} \neq \boldsymbol{0}$ be a (right) Eigenvector of $J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}}$ to the Eigenvalue $\rho(\mu\boldsymbol{f})$. Then, as $(\mathrm{Id} - J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}})^{-1}$ is assumed to exist, and

$$(\mathrm{Id} - J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}})\boldsymbol{v} = (1 - \rho(\mu\boldsymbol{f}))\boldsymbol{v}$$

we have $\rho(\mu\boldsymbol{f}) \neq 1$. As $\rho(\cdot)$ is continuous on B, by reducing the radius of B we may assume that $\rho(\cdot) \neq 1$ on B. Note that B intersects $[\mathbf{0}, \mu\boldsymbol{f})$. But on $[\mathbf{0}, \mu\boldsymbol{f})$ we know that $\rho(\cdot) < 1$. So, $\rho(\cdot) < 1$ on B follows, and, hence the Neumann series exists on B with

$$(\mathrm{Id} - J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}})^{-1} = J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}. \qquad \square$$

Consider now the quadrics

$$[\boldsymbol{q}_i = 0] := \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{q}_i(\boldsymbol{x}) = 0\}$$

whose least nonnegative intersection is $\mu\boldsymbol{f}$. For $\boldsymbol{q}$ we obtain

$$J_{\boldsymbol{q}}|_{\boldsymbol{x}} = J_{\boldsymbol{f}}|_{\boldsymbol{x}} - \mathrm{Id} = 2B(\boldsymbol{x}, \cdot) + L - \mathrm{Id}.$$

As already mentioned, the i-th row of $J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}$ is basically the normal to the tangent plane at $[\boldsymbol{q}_i = 0]$ in $\mu\boldsymbol{f}$. Our assumption of the existence of the inverse of $J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}$ thus means that the normals at the n quadrics in $\mu\boldsymbol{f}$ are linearly independent.

Assume for the moment that $\boldsymbol{\nu}$ is a fixed point of $\boldsymbol{f}$ with $\boldsymbol{\nu} > \mu\boldsymbol{f}$. Let $\boldsymbol{r} := \boldsymbol{\nu} - \mu\boldsymbol{f} > \mathbf{0}$ be the direction in which we must head from $\mu\boldsymbol{f}$ to reach $\boldsymbol{\nu}$. Consider $\boldsymbol{q}$ along the ray $\mu\boldsymbol{f} + t \cdot \boldsymbol{r}$ for $t \in [0, 1]$. Then we have

$$\begin{aligned} \boldsymbol{q}(\mu\boldsymbol{f} + t\boldsymbol{r}) &= \boldsymbol{q}(\mu\boldsymbol{f}) + t \cdot J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}\boldsymbol{r} + t^2 \cdot B(\boldsymbol{r}, \boldsymbol{r}) \\ &= t \cdot \big(J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}\boldsymbol{r} + t \cdot B(\boldsymbol{r}, \boldsymbol{r})\big). \end{aligned}$$

As $B(\boldsymbol{r}, \boldsymbol{r}) \geq \mathbf{0}$, and $\boldsymbol{q}(\mu\boldsymbol{f} + \boldsymbol{r} = \mathbf{0})$, we have

$$\boldsymbol{d} := J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}\boldsymbol{r} < \mathbf{0}.$$

The strict inequality has to hold as $\boldsymbol{r} > \mathbf{0}$, and $J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}$ is invertible. From this it follows that $B(\boldsymbol{r}, \boldsymbol{r}) > \mathbf{0}$.

As $-J_{\boldsymbol{q}}|^{-1}_{\mu\boldsymbol{f}} = J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} \geq \mathbf{0}$, we can rewrite the last equation as

$$\begin{aligned} & & J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}\boldsymbol{r} &= \boldsymbol{d} & (< \mathbf{0}) \\ \Leftrightarrow\ & (-J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}})^{-1} J_{\boldsymbol{q}}|_{\mu\boldsymbol{f}}\boldsymbol{r} &= J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{d} & (< \mathbf{0}) \\ \Leftrightarrow\ & & \boldsymbol{r} &= J_{\boldsymbol{f}}|^*_{\boldsymbol{f}}(-\boldsymbol{d}) & (> \mathbf{0}). \end{aligned}$$

From this it follows that when looking for a second nonnegative fixed point we only need to consider directions $\boldsymbol{r} := J_{\boldsymbol{f}}|^*_{\boldsymbol{f}}\boldsymbol{d}$ with $\boldsymbol{d} > \mathbf{0}$.

We therefore move the origin of the coordinate system into $\mu\boldsymbol{f}$, and change the basis to the one given by the normals of the quadrics in $\mu\boldsymbol{f}$, i.e.,

$$\widetilde{\boldsymbol{q}}(\boldsymbol{X}) := \boldsymbol{q}(\mu\boldsymbol{f} + J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X}).$$

One can easily check that

$$\boldsymbol{q}(\boldsymbol{x}+\boldsymbol{y}) = \boldsymbol{q}(\boldsymbol{x}) + J_{\boldsymbol{q}}|_{\boldsymbol{x}}\boldsymbol{y} + B(\boldsymbol{y},\boldsymbol{y}),$$

so that we obtain

$$\widetilde{\boldsymbol{q}}(\boldsymbol{X}) = -\boldsymbol{X} + B(J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X}, J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X})$$

where we have used that $J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} = -J_{\boldsymbol{q}}|^{-1}_{\mu\boldsymbol{f}}$. Finally, we set

$$\widetilde{\boldsymbol{f}}(\boldsymbol{X}) := B(J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X}, J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X}).$$

Note that $\widetilde{\boldsymbol{f}}$ does not need to be strongly-connected anymore as shown in the following example.

Example 6.4.3. For $\boldsymbol{X} = (X,Y)^\top$ consider the system

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{pmatrix} \frac{1}{4}X^2 + \frac{1}{2}Y + \frac{1}{4} \\ \frac{1}{2}X + \frac{1}{2} \end{pmatrix}.$$

The least nonnegative fixed point of $\boldsymbol{f}$ is $\mu\boldsymbol{f} = (1,1)^\top$. Further, we have that $J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}} - \mathrm{Id}$ is invertible with

$$-(J_{\boldsymbol{f}}|^{-1}_{\mu\boldsymbol{f}} - \mathrm{Id}) = J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix},$$

i.e., $\boldsymbol{f}$ is strongly-connected. After applying the transformation $\mu\boldsymbol{f} + J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X}$ we obtain the new system

$$\widetilde{\boldsymbol{f}}(\boldsymbol{X}) = \begin{pmatrix} (2X+Y)^2 \\ 0 \end{pmatrix}.$$

Clearly, this system is not strongly-connected anymore. Further, we can simplify it to the univariate system

$$X = 4X^2$$

having the two solutions $X = 0$, and $X = \frac{1}{4}$. This yields the two fixed points $(0,0)^\top$, and $(\frac{1}{4},0)^\top$ of $\widetilde{\boldsymbol{f}}$. Moving back to the original coordinate system yields the second nonnegative fixed point $\boldsymbol{\nu}$ of $\boldsymbol{f}$:

$$\boldsymbol{\nu} = \mu\boldsymbol{f} + J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} \begin{pmatrix} \frac{1}{4} \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ \frac{3}{2} \end{pmatrix}.$$

◇

Recall that $B_i(\boldsymbol{X},\boldsymbol{X}) = \boldsymbol{X}^\top A^{(i)}\boldsymbol{X}$ with $A^{(i)} \geq 0$ and $A^{(i)} = (A^{(i)})^\top$. From this we obtain for $\widetilde{\boldsymbol{f}}_i$:

$$\widetilde{\boldsymbol{f}}_i(\boldsymbol{X}) = \boldsymbol{X}^\top \left(J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\right)^\top A^{(i)} J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\boldsymbol{X}.$$

Set

$$\widetilde{A}^{(i)} := \left(J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}\right)^\top A^{(i)} J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}}.$$

We claim that either $\widetilde{A}^{(i)} = \mathbf{0}$, or $\widetilde{A}^{(i)} \succ \mathbf{0}$. If $A^{(i)} = \mathbf{0}$, then of course we also have $\widetilde{A}^{(i)} = \mathbf{0}$. Assume that $A^{(i)} > \mathbf{0}$. By our assumptions that $\mu\boldsymbol{f} \succ \mathbf{0}$ and that $\boldsymbol{f}$ is strongly-connected, we have $J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} \succ \mathbf{0}$. One easily checks that $\widetilde{A}^{(i)} \succ \mathbf{0}$ ([2]) follows. We therefore always may simplify $\widetilde{\boldsymbol{f}}$ by setting all variables X_i with $\widetilde{\boldsymbol{f}}_i = 0$ to zero. The resulting system is then strongly-connected w.r.t. to the remaining variables. Unless otherwise noted we assume in the following that $\widetilde{A}^{(i)} \succ \mathbf{0}$ for all $i \in [n]$.

We now have to find a fixed point of $\widetilde{\boldsymbol{f}}$ which is greater than $\mathbf{0}$, and we know that we only have to look for such a fixed point along rays $\mathbb{R}_{\geq 0}\boldsymbol{d} = \{t \cdot \boldsymbol{d} \mid t \in \mathbb{R}_{\geq 0}\}$ with $\boldsymbol{d} > \mathbf{0}$. The intersection of such ray with the quadric $[\widetilde{\boldsymbol{q}}_i = 0]$ can be determined from

$$\widetilde{\boldsymbol{q}}(t \cdot \boldsymbol{d}) = t^2 \cdot \widetilde{\boldsymbol{f}}(\boldsymbol{d}) - t \cdot \boldsymbol{d},$$

that is for $t > 0$ the ray can only have some point common with $[\widetilde{\boldsymbol{q}}_i = 0]$ if either $\widetilde{\boldsymbol{f}}(\boldsymbol{d})_i = 0 \wedge \boldsymbol{d}_i = 0$, i.e., the ray $\mathbb{R}_{\geq 0}\boldsymbol{d}$ is contained in $[\widetilde{\boldsymbol{q}}_i = 0]$, or $\widetilde{\boldsymbol{f}}(\boldsymbol{d}, \boldsymbol{d})_i > 0 \wedge \boldsymbol{d}_i > 0$ and the ray intersects $[\widetilde{\boldsymbol{q}}_i = 0]$ for

$$t = \lambda_i(\boldsymbol{d}) := \frac{\boldsymbol{d}_i}{\widetilde{\boldsymbol{f}}(\boldsymbol{d})_i}.$$

Note that from our assumptions (i.e., $\widetilde{A}^{(i)} \succ \mathbf{0}$ and $\boldsymbol{d} > \mathbf{0}$) it follows that $\widetilde{\boldsymbol{f}}_i(\boldsymbol{d}) > 0$. Thus, if there is a second nonnegative fixed point $\boldsymbol{\nu}$ of $\boldsymbol{f}$ (with $\boldsymbol{\nu} \geq \mu\boldsymbol{f}$), then there has to be a direction $\boldsymbol{d}^* > 0$ such that

$$\forall i \in [n] : \lambda_i(\boldsymbol{d}^*) = \lambda_j(\boldsymbol{d}^*).$$

Definition 6.4.4.
The *standard simplex of dimension k* is defined by

$$\Delta_k := \Big\{\boldsymbol{d} \in [0,1]^n \mid \sum_{i \in [k+1]} \boldsymbol{d}_i = 1\Big\}$$

Given $k+1$ points $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k+1)}$ in $\mathbb{R}^{k+1}$ in general position, they define the k-dimensional simplex

$$\Delta(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k+1)}) := \Big\{ \sum_{i \in [k+1]} \boldsymbol{d}_i \cdot \boldsymbol{v}^{(i)} \mid \boldsymbol{d} \in \Delta_k \Big\}.$$

A *simplicial subdivision* of a k-dimensional simplex S is a partition of S into smaller k-dimensional simplices, called cells, such that any two cells are either disjoint or their intersection itself is a simplex of dimension $\leq k$. ◇

[2]For this to see, assume that $A^{(i)}$ has exactly one positive entry, w.l.o.g. $A^{(i)}_{1,1} > 0$. Then $(A^{(i)} J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}})_{1,j} > 0$ for all $j \in [n]$, and finally $\widetilde{A}^{(i)}_{j,k} > 0$ for all $j, k \in [n]$.

Obviously, it suffices to consider $\boldsymbol{d} \in \Delta_{n-1}$ as directions in which a second nonnegative fixed point might be found. It is therefore sufficient to show that there is a point $\boldsymbol{d}^* \in S_{n-1}$ such that

$$\forall i, j \in [n] : \lambda_i(\boldsymbol{d}^*) = \lambda_j(\boldsymbol{d}^*).$$

For this, we will use *Sperner's lemma*.

Lemma 6.4.5 (Sperner's lemma)**.**
Let $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k+1)} \in \mathbb{R}^{k+1}$ be some points in general position, and V the vertices of some subdivision S of $\Delta(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k+1)})$. Further assume that every vertex $\boldsymbol{v} \in V$ is painted in some color from $[k+1]$ with the restriction that (i) the corners $\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k+1)}\}$ are all colored differently, and (ii) vertices $\boldsymbol{v}$ located in the subsimplex $\Delta(\boldsymbol{v}^{(i_1)}, \ldots, \boldsymbol{v}^{(i_l)})$ (for $\{i_1, \ldots, i_l\} \subseteq [k+1]$) are painted only in the colors from $\{i_1, \ldots, i_l\}$. Then there is a cell of S whose vertices have all different colors. ◇

We use Sperner's lemma as follows: as all functions λ_i are continuous on Δ_{n-1}, and Δ_{n-1} is compact in $\mathbb{R}^n$, we find for any given $\epsilon > 0$ a $\delta_\epsilon > 0$ such that for every two $\boldsymbol{d}, \boldsymbol{d}' \in \Delta_{n-1}$ with $\|\boldsymbol{d} - \boldsymbol{d}'\|_2 < \delta_\epsilon$ we have $|\lambda_i(\boldsymbol{d}) - \lambda_i(\boldsymbol{d}')| < \epsilon$. Further, we can find for every such δ_ϵ a subdivision of Δ_{n-1} such that every cell fits into an open ball of diameter δ_ϵ, e.g. by repeatedly splitting edges into half. Every vertex $\boldsymbol{d}$ of such a subdivision we assign a color $c(\boldsymbol{d}) \in [n]$ such that $\lambda_{c(\boldsymbol{d})}(\boldsymbol{d}) \geq \lambda_i(\boldsymbol{d})$ for all $i \in [n]$. That is the ray in the direction $\boldsymbol{d}$ hits the quadric $[\widetilde{\boldsymbol{q}}_c = 0]$ the latest. One easily checks that this coloring satisfies the requirement of Sperner's lemma. Sperner's lemma now tells us that there is a cell of such a subdivision whose vertices all have different color, i.e., for every vertex of such a cell another quadric is hit the latest. As all functions λ_i only vary by ϵ on such a cell, we then can show that by letting ϵ go to 0, the vertices of such a cell converge to a point $\boldsymbol{d}^*$ with $\lambda_i(\boldsymbol{d}^*) = \lambda_j(\boldsymbol{d}^*)$ for all $i, j \in [n]$.

We now give a formal proof.

Theorem 6.4.6.
Let $\phi_1, \ldots, \phi_n : \mathbb{R}^n \to \mathbb{R}$ be n continuous functions with $\phi(\boldsymbol{d}) > 0$ for all $\boldsymbol{d} \in \Delta_{n-1}$. Set $\mu_i(\boldsymbol{d}) := \frac{\boldsymbol{d}_i}{\phi_i(\boldsymbol{d})}$. Then there is some $\boldsymbol{d}^* \in \Delta_{n-1}$ such that $\mu_i(\boldsymbol{d}^*) = \mu_j(\boldsymbol{d}^*)$ for all $i, j \in [n]$. In particular, we have $\boldsymbol{d}^* \succ \boldsymbol{0}$. ◇

Proof. If $n = 1$, we have $d^* = 1$. Thus assume that $n > 1$. As μ_i is the composition of continuous functions, and $\phi_i(\boldsymbol{d}) > 0$ for all $\boldsymbol{d} \in \Delta_{n-1}$, μ_i is continuous on Δ_{n-1}. In particular, Δ_{n-1} is compact, so μ_i is uniformly continuous on Δ_{n-1}. We therefore find for every $\epsilon > 0$ a $\delta_\epsilon^{(i)}$ such that for all $\boldsymbol{d}, \boldsymbol{d}' \in \Delta_{n-1}$ with $\|\boldsymbol{d} - \boldsymbol{d}'\|_2 < \delta_\epsilon^{(i)}$ we have $\left|\mu_i(\boldsymbol{d}) - \mu_i(\boldsymbol{d}')\right| < \epsilon$. Set $\delta_\epsilon := \min\{\delta_\epsilon^{(1)}, \ldots, \delta_\epsilon^{(n)}\}$.

Choose now some subdivision of Δ_{n-1} such that every cell fits into an open ball of diameter δ_ϵ, and let $V \subseteq \Delta_{n-1}$ denote the set of vertices of the chosen subdivision. We assign every $\boldsymbol{d} \in V$ the color $c(\boldsymbol{d})$ defined by

$$c(\boldsymbol{d}) := \min\{k \in [n] \mid \forall i \in [n] : \mu_k(\boldsymbol{d}) \geq \mu_i(\boldsymbol{d})\}.$$

Let $\boldsymbol{d}^{(i)}$ denote the corner of Δ_{n-1} with $\boldsymbol{d}^{(i)}_i = 1$. In particular $\Delta_{n-1} = \Delta(\boldsymbol{d}^{(1)}, \ldots, \boldsymbol{d}^{(n)})$. We then have $\mu_i(\boldsymbol{d}^{(i)}) > 0$, and for all $j \neq i$ we have $\mu_j(\boldsymbol{d}^{(i)}) = 0$. Thus, all corners of the subdivision are colored differently. Similarly, we have $\mu_j(\boldsymbol{d}) = 0$ for all $\boldsymbol{d} \in \Delta(\boldsymbol{d}^{(i_1)}, \ldots, \boldsymbol{d}^{(i_l)})$, if $j \notin \{i_1, \ldots, i_l\}$ (with $\{i_1, \ldots, i_l\} \subseteq [n]$). By Sperner's lemma we therefore find a cell of the subdivision whose vertices are all colored differently. Let $\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}\}$ be the vertices of such a cell. We may assume that $c(\boldsymbol{v}^{(i)}) = i$.

As we have chosen the subdivision in such a way that the cell $\Delta(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)})$ fits into an open ball of diameter δ_ϵ, we have for all $\boldsymbol{v}, \boldsymbol{v}' \in \Delta(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)})$ that $|\mu_i(\boldsymbol{v}) - \mu_i(\boldsymbol{v}')| < \epsilon$, in particular

$$|\mu_i(\boldsymbol{v}^{(i)}) - \mu_i(\boldsymbol{v}^{(j)})| < \epsilon \text{ for all } i, j \in [n].$$

From this it follows that

$$\mu_j(\boldsymbol{v}^{(i)}) \geq \mu_j(\boldsymbol{v}^{(j)}) - \epsilon \text{ for all } i, j \in [n].$$

Consider now two vertices $\boldsymbol{v}^{(i)}$ and $\boldsymbol{v}^{(j)}$ (with $i \neq j$). We then have

$$\mu_i(\boldsymbol{v}^{(i)}) \geq \mu_j(\boldsymbol{v}^{(i)}), \text{ resp. } \mu_j(\boldsymbol{v}^{(j)}) \geq \mu_i(\boldsymbol{v}^{(j)})$$

by Sperner's lemma. Combining this with the previous inequality, we obtain

$$\mu_i(\boldsymbol{v}^{(i)}) \geq \mu_j(\boldsymbol{v}^{(i)}) \geq \mu_j(\boldsymbol{v}^{(j)}) - \epsilon.$$

As the same holds true for i and j interchanged, we get

$$\mu_i(\boldsymbol{v}^{(i)}) \geq \mu_j(\boldsymbol{v}^{(i)}) \geq \mu_i(\boldsymbol{v}^{(i)}) - 2\epsilon.$$

By continuity of the μ_i, we therefore get that in the limit, i.e., $\epsilon \to 0$, we find a $\boldsymbol{d}^* \in \Delta_{n-1}$ with $\mu_i(\boldsymbol{d}^*) = \mu_j(\boldsymbol{d}^*)$ for all $i, j \in [n]$.

As $\boldsymbol{d}^* \neq \boldsymbol{0}$, we have $\mu_i(\boldsymbol{d}^*) > 0$ for at least one $i \in [n]$ and, thus, for all $i \in [n]$ implying that $\boldsymbol{d}^* \succ \boldsymbol{0}$. □

For the existence of a second nonnegative fixed point in the case that $\boldsymbol{f}$ is strongly-connected (but not necessarily $\widetilde{\boldsymbol{f}}$), we proceed as already sketched in Example 2.

Theorem 6.4.7.
Every clean, feasible, strongly-connected system $\boldsymbol{f}(\boldsymbol{X})$ with $J_{\boldsymbol{f}}|_{\mu\boldsymbol{f}} - \mathrm{Id}$ invertible, and $\boldsymbol{f}$ quadratic in at least one component, has a second nonnegative fixed point $\boldsymbol{\nu}$ with $\boldsymbol{\nu} \succ \mu\boldsymbol{f}$. ◇

Proof. We first move from $\boldsymbol{f}$ to $\widetilde{\boldsymbol{f}}$. Let $\{X_{i_1}, \ldots, X_{i_k}\}$ be the set of variables with $\widetilde{\boldsymbol{f}}_{i_l} \neq 0$. Set the remaining variables to 0. As we assume that $\boldsymbol{f}$ is quadratic in at least one component, we have $k \geq 1$.

Apply now Theorem 6.4.6 to the functions $\phi_l := \widetilde{\boldsymbol{f}}_{i_l}$ for $l \in [k]$. This yields a direction $\boldsymbol{d}^* \in \Delta_{k-1}$. We lift $\boldsymbol{d}^*$ to $\mathbb{R}^n$ by means of the vector $\boldsymbol{r}$ with $\boldsymbol{r}_{i_l} := \boldsymbol{d}^*_l$ for $l \in [k]$, and $\boldsymbol{r}_j := 0$ for $j \notin \{i_1, \ldots, i_k\}$. Then

$$\widetilde{\boldsymbol{f}}(\lambda_{i_1}(\boldsymbol{d}^*) \cdot \boldsymbol{r}) = \lambda_{i_1}(\boldsymbol{d}^*)\boldsymbol{r}_j \text{ for all } j \in [n].$$

Finally, set

$$\boldsymbol{\nu} := \mu\boldsymbol{f} + J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} \cdot (\lambda_i(\boldsymbol{d}^*)\boldsymbol{r}).$$

By construction we have $\boldsymbol{f}(\boldsymbol{\nu}) = \boldsymbol{\nu}$. As $\boldsymbol{r} > \boldsymbol{0}$, and $J_{\boldsymbol{f}}|^*_{\mu\boldsymbol{f}} \succ \boldsymbol{0}$, it follows $\boldsymbol{\nu} \succ \mu\boldsymbol{f}$. □

Chapter 7

Conclusions

We discuss the main results obtained in the preceding chapters and possible future work.

7.1 Contribution

Our main results in Chapter 3 were the generalization of Newton's method to ω-continuous semirings and the characterization of the Newton approximants by means of derivation trees: we introduced the notion of dimension of a tree and obtained the result that the k-th Newton approximants exactly corresponds to the yields of all trees of dimension at most k. We showed that the generalized Newton's method allows for a better approximation of the least fixed point of a system of power-series than the classical Kleene fixed point iteration. In particular, we showed that Newton's method always converges to the least fixed point after a number of steps linear in the number of variables if addition is idempotent and multiplication is commutative. Note that even in this more restrictive case the Kleene sequence is not guaranteed to reach the least fixed point. We further lifted this result to the more general setting of commutative Kleene algebras, thereby identifying the method described by Hopkins and Kozen in [HK99] as Newton's method and improving their exponential upper bound on the number of iterations.

We also studied the connection between Newton's method and languages of finite-index, the latter a long-standing concept of formal language theory dating back to 1967 [Ynt67]. Here we obtained the result that for context-free languages the i-th Newton approximant corresponds to the index-$i+1$ ap-

proximation implying that Newton's method converges on idempotent semirings if the considered system of power series represents a context-free grammar of finite index. As on general (commutative) ω-continuous semirings Newton's method is not guaranteed to reach the least fixed point in a finite number of steps, e.g. consider the real semiring, we therefore obtained a first classification of when Newton's method can be used to calculate the least fixed point.

In Chapter 4 we studied in more detail the proof concept underlying our result on the convergence of Newton's method on commutative and idempotent semirings. Using the connection between approximants and derivation trees, we identified three classes of semirings which allow for an even faster calculation of the least fixed point, namely star-distributive, lossy and 1-bounded semirings. In the case of star-distributive semirings we showed that we can obtain from any polynomial system a linear system having the same least fixed point, thus reducing the problem of solving a nonlinear system to the problem of solving a linear one. Note that our result on Newton's method on commutative and idempotent semirings basically says that one can obtain the least fixed point of nonlinear system by solving a linear number of linear systems. In this sense, star-distributive semirings allow for an even faster calculation of the least fixed point. We applied our result on star-distributive semirings to the problem of the throughput of a procedural program and obtained a speed up of the algorithm presented in [CCFR07].
In the case of lossy semirings we showed that for every strongly connected system the same construction as in the case of star-distributive semirings yields a linear system preserving the least fixed point. We also showed how this then allows to calculate the least fixed point of arbitrary nonlinear systems. We therefore obtained a more general restatement of the result by Courcelle [Cou91] that the downward-closure of a context-free language is regular.
Finally, we considered 1-bounded semirings in Chapter 4 and showed that on these already the Kleene sequence reaches the least fixed point after a number of iterations given by the number of variables considered.
In Chapter 5 we studied totally ordered semirings motivated by our results on star-distributive semirings. Assuming that addition is idempotent, addition becomes the maximum w.r.t. the natural order on such semirings, and it is natural to study min-max-systems, i.e., polynomial systems not only using maximum as addition, but also minimum as a second kind of addition. From our result on star-distributive semirings, we immediately obtained the result that on totally ordered star-distributive semirings the least fixed point of a min-max-system exists and can be represented by regular expressions.

In the second part of Chapter 5 we considered strategy iteration, a well-known approach for solving min-max-systems. Motivated by existing applications of strategy iteration to infinite two-person games (parity games, mean-payoff games) [VJ00, BSV04] and interval analysis [GS08] we introduced si-semirings, a class of semirings general enough to encompass all these special cases, and showed that the algorithm by Gawlitza and Seidl [GS08] can be generalized to this class. We further extended their approach to nondeterministic strategies and showed that required approximations can be calculated by means of a linear number of Kleene fixed point iterations. We then explicitly studied linear min-max-systems and showed that the use of nondeterministic strategies allows for an improved bound on the number of strategies considered. We then instantiated our algorithm explicitly to parity games in order to exemplify the results obtain in this chapter.

In Chapter 6 we studied polynomial systems on the nonnegative reals in more detail. In the first part, we showed that the visually motivated method of approximation by means of tangents indeed also converges to the least fixed point and that it does so at least as fast as Newton's method. In the second part of Chapter 6 we turned to the questions of the existence of a second fixed point, a question arising from the study of multi-type Galton-Watson processes.

7.2 Open Problems

We discuss several open problems we deem worthwhile to study in more detail in future work.

We have seen that Newton's method converges in a finite number of steps on commutative and idempotent ω-continuous semirings. Idempotence in particular can be described by adding the axiom $1 + 1 = 1$. A natural extension is to consider semirings where the equation $k = k + 1$ holds for some $k \in \mathbb{N}$ (cf. [Ési08]) assuming the canonical embedding of $\mathbb{N}$ into the semiring. It should not be too hard to show that Newton's method also converges on commutative ω-continuous semirings with this generalized form of idempotence, but it remains future work to check the proofs in detail.

As already stated in Section 3.4.3 the question arises if there is a purely equational proof of the fact that Newton's method converges in a linear number of steps to the least fixed point on commutative and idempotent semirings. It was shown by Aceto, Ésik and Ingólfsdóttir in [AEI01] that the

convergence can be shown in this way, but the question on the number of iterations remains open.

We also consider the setting where only subdistributivity holds, but not distributivity. This setting was motivated by static analyses which are not distributive. We showed that both the Kleene and the Newton sequence converge to a safe overapproximation of the MOP, i.e., the sum of all values of all terminating runs where the limit of the Kleene sequence is always less than or equal to the limit of the Newton sequence. It remains to analyze when Kleene and Newton coincide in the subdistributive setting.

Another open question is the generalization of the results given regarding semirings if one replaces multiplication by a family of operations of varying arity. For example in [Knu77] the following problem is considered:

Example 7.2.1. Let $\mathcal{F} = \{g_1, \ldots, g_r\}$ be a family of functions from $(\mathbb{R}_{\geq 0} \cup \{\infty\})^{k_i} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ with $k_i \in \mathbb{N}$ for $i \in \{1, \ldots, r\}$. Every function $g \in \mathcal{F}$ (of arity k) is assumed to satisfy:

$$g(x_1, \ldots, x_k) \geq \max\{x_1, \ldots, x_n\} \text{ for all } x_1, \ldots, x_k \in \mathbb{R}_{\geq 0} \cup \{\infty\}.$$

Assume now we are given a context-free grammar G with variables $\mathcal{X}$ and constants $\mathcal{F} \cup \{(,)\}$ such that every production rule is of the form

$$X \to g(X_1, \ldots, X_k)$$

with $g \in \mathcal{F}$, k the arity of g, and $X_1, \ldots, X_k \in \mathcal{X}$.

Given an axiom X_0, every word of the represented language $L_{X_0}(G)$ naturally defines a value in $\mathbb{R}_{\geq 0} \cup \{\infty\}$ by means of evaluation. Knuth is then interested in the infimum of all the values represented by $L(G)$.

The same argument as in the case of 1-bounded semirings immediately shows that it suffices to consider only these words which possess a derivation tree of height at most n: For every tree of height at least $n+1$ we can again find a pump tree representing a unary function $f(x)$ satisfying $f(x) \geq x$. See also the proof of Lemma 5.3.12 where we have used a similar argument for trees whose nodes where labeled not only by monomials, but by polynomials w.r.t. $\sqcup$ as semiring addition. ◇

In particular, in the setting of strategy iteration this generalization might be worthwhile to study as this would allow to encompass results of [EGKS08]. There strategy iteration was applied to min-max-systems where multiplication was replaced by the familiy of polynomials with positive coefficients. Regarding our results on strategy iteration it also remains to study the role of commutativity, and if locally optimal strategy iteration also allows to prove a better upper bound on the number iterations in the case of nonlinear min-max-systems.

Appendix A

Missing Proofs of Chapter 3

A.1 Proofs of Section 3.2

To avoid typographical clutter in the following proofs, we use the following notation. Given some class of objects (e.g. derivation trees t) and a predicate $P(t)$, we write

$$\sum_t \mathsf{Y}(t) : P(t)$$

instead of

$$\sum_{t \text{ such that } P(t) \text{ holds}} \mathsf{Y}(t)\,.$$

Proposition 3.2.5.
$\left(\boldsymbol{\kappa}^{(i)}\right)_X = \mathsf{Y}(\mathcal{H}^i_X)$, i.e., the X-component of the i-th Kleene approximant $\boldsymbol{\kappa}^{(i)}$ is equal to the yield of $\mathcal{H}^i_X$. ◇

Proof. By induction on i. The base case $i = 0$ is easy. Induction step ($i \geq 0$):

$$\begin{aligned}
&\left(\boldsymbol{\kappa}^{(i+1)}\right)_X \\
&\quad = \boldsymbol{f}_X(\boldsymbol{\kappa}^{(i)}) \\
&\quad = \sum_{j \in J} m_{X,j}(\boldsymbol{\kappa}^{(i)}) \\
&\quad = \sum_{j \in J} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots X_k a_{k+1} \\ y = a_1 \boldsymbol{\kappa}^{(i)}_{X_1} \cdots \boldsymbol{\kappa}^{(i)}_{X_k} a_{k+1} \end{array} \right.
\end{aligned}$$

by induction:

$$
\begin{aligned}
&= \sum_{j\in J} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots X_k a_{k+1} \\ y = a_1 \,\mathsf{Y}(\mathcal{H}^i_{X_1}) \cdots \mathsf{Y}(\mathcal{H}^i_{X_k}) a_{k+1} \end{array} \right. \\
&= \sum_{\substack{j\in J \\ t_1,\ldots,t_k}} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots X_k a_{k+1} \\ t_1,\ldots,t_k \text{ trees with } h(t_r) \le i, \lambda_v(t_r) = X_r \quad (1 \le r \le k) \\ y = a_1 \,\mathsf{Y}(t_1) \cdots \mathsf{Y}(t_k) a_{k+1} \end{array} \right. \\
&= \sum_{j\in J,t} \mathsf{Y}(t) : t \text{ is a tree with } h(t) \le i+1,\ \lambda(t) = (X,j) \\
&= \mathsf{Y}(\mathcal{H}^i_X) \qquad \square
\end{aligned}
$$

The following definition of *fine dimension* is analogous to Definition 3.2.9, but adds a second component, which measures the length of the path from the root to the lowest node with the same dimension as the root:

Definition A.1.1 (fine dimension)**.**
The *fine dimension* $dl(t) = (d(t), l(t))$ of a tree t is inductively defined as follows:

(1) If t has no children, then $dl(t) = (0, 0)$.

(2) If t has exactly one child t_1, then $dl(t) = (d(t_1), l(t_1) + 1)$.

(3) If t has at least two children, let t_1, t_2 be two distinct children of t such that $d(t_1) \ge d(t_2)$ and $d(t_2) \ge d(t')$ for every child $t' \ne t_1$. Let $d_1 = d(t_1)$ and $d_2 = d(t_2)$. Then

$$
dl(t) = \left\{ \begin{array}{ll} (d_1 + 1, 0) & \text{if } d_1 = d_2 \\ (d_1, l(t_1) + 1) & \text{if } d_1 > d_2. \end{array} \right. \qquad \diamond
$$

Remark A.1.2.
Notice that, by Definition 3.2.12, a tree t is proper if and only if $l(t) = 0$. So we have:

$$
\mathsf{Y}(P^i_X) = \sum_t \mathsf{Y}(t) : t \text{ tree with } \lambda_v(t) = X,\ dl(t) = (i, 0). \qquad \diamond
$$

Now we can prove the remaining lemmata from Section 3.2.

Lemma 3.2.14.
For every variable $X \in \mathcal{X}$ and every $i \ge 0$: $\boldsymbol{\tau}^{(i)}_X = \mathsf{Y}(\mathcal{D}^i_X)$. $\diamond$

Proof. By induction on i. Induction base ($i = 0$):

$$\begin{aligned}\boldsymbol{\tau}_X^{(0)} = \boldsymbol{f}_X(\mathbf{0}) &= \sum_t \mathsf{Y}(t) : \lambda_v(t) = X, h(t) = 0 \\ &= \sum_t \mathsf{Y}(t) : \lambda_v(t) = X, d(t) = 0 \\ &= \mathsf{Y}(\mathcal{D}_X^0)\end{aligned}$$

Induction step ($i + 1 > 0$):
We need to show that $D\boldsymbol{f}|^*_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)})$ equals exactly the yield of all trees of dimension $i+1$, i.e., that for all $X \in \mathcal{X}$

$$\left(D\boldsymbol{f}|^*_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)})\right)_X = \sum_t \mathsf{Y}(t) : \lambda_v(t) = X,\ d(t) = i+1\ .$$

We prove the following stronger claim by induction on p:

$$\left(D\boldsymbol{f}|^p_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)})\right)_X = \sum_t \mathsf{Y}(t) : \lambda_v(t) = X,\ dl(t) = (i+1, p)$$

The claim holds for $p = 0$ by Remark A.1.2. For the induction step, let $p \geq 0$. Then we have for all $X \in \mathcal{X}$:

$$\begin{aligned}&\left(D\boldsymbol{f}|^{p+1}_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)})\right)_X \\ &\quad= \left(D\boldsymbol{f}|_{\boldsymbol{\tau}^{(i)}} \circ\ D\boldsymbol{f}|^p_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)})\right)_X \\ &\quad= D\boldsymbol{f}_X|_{\boldsymbol{\tau}^{(i)}} \circ\ D\boldsymbol{f}|^p_{\boldsymbol{\tau}^{(i)}}(\boldsymbol{\delta}^{(i)})\end{aligned}$$

Define the vector $\widetilde{\boldsymbol{Y}}$ by $\widetilde{\boldsymbol{Y}}_{X_0} = \sum_t \mathsf{Y}(t) : \lambda_v(t) = X_0, dl(t) = (i+1, p)$. Then, by induction hypothesis (on p), above expression is equal to

$$\begin{aligned}&= D\boldsymbol{f}_X|_{\boldsymbol{\tau}^{(i)}}(\widetilde{\boldsymbol{Y}}) \\ &= \sum_{j \in J} Dm_{X,j}|_{\boldsymbol{\tau}^{(i)}}(\widetilde{\boldsymbol{Y}}) : m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ &= \sum_{j \in J, r} y : \begin{cases} m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ 1 \leq r \leq k \\ y = a_1 \boldsymbol{\tau}^{(i)}_{X_1} \cdots a_r \widetilde{\boldsymbol{Y}}_{X_r} a_{r+1} \boldsymbol{\tau}^{(i)}_{X_{r+1}} \cdots a_k \boldsymbol{\tau}^{(i)}_{X_k} a_{k+1} \end{cases} \\ \text{(by induction on } i) &= \sum_{\substack{j \in J, r, \\ t_1, \ldots, t_k}} y : \begin{cases} m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ 1 \leq r \leq k \\ t_1, \ldots, t_k \text{ trees with } \lambda_v(t_s) = X_s \quad (1 \leq s \leq k) \\ \qquad dl(t_r) = (i+1, p), \\ \qquad d(t_s) \leq i \quad (1 \leq s \leq k,\ s \neq r) \\ y = a_1\, \mathsf{Y}(t_1) \cdots a_r\, \mathsf{Y}(t_r) \cdots a_k\, \mathsf{Y}(t_k) a_{k+1} \end{cases} \\ &= \sum_{j \in J, t} \mathsf{Y}(t) : t \text{ tree with } \lambda(t) = (X, j),\ dl(t) = (i+1, p+1) \\ &= \sum_t \mathsf{Y}(t) : t \text{ tree with } \lambda_v(t) = X,\ dl(t) = (i+1, p+1) \qquad \square\end{aligned}$$

Lemma 3.2.15.
The sequence $(\boldsymbol{\tau}^{(i)})_{i\in\mathbb{N}}$ is a Newton sequence as defined in Definition 3.1.9, i.e., the $\boldsymbol{\delta}^{(i)}$ of Definition 3.2.12 satisfy $\boldsymbol{f}(\boldsymbol{\tau}^{(i)}) = \boldsymbol{\tau}^{(i)} + \boldsymbol{\delta}^{(i)}$. ◇

Proof.

$$\begin{aligned}
\boldsymbol{f}_X(\boldsymbol{\tau}^{(i)}) &= \sum_{j\in J} m_{X,j}(\boldsymbol{\tau}^{(i)}) \\
&= \sum_{j\in J} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ y = a_1 \boldsymbol{\tau}^{(i)}_{X_1} \cdots a_k \boldsymbol{\tau}^{(i)}_{X_k} a_{k+1} \end{array} \right.
\end{aligned}$$

(by Lemma 3.2.14)

$$\begin{aligned}
&= \sum_{\substack{j\in J \\ t_1,\ldots,t_k}} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ t_1,\ldots,t_k \text{ trees with } \lambda_v(t_r) = X_r,\ d(t_r) \le i, \quad (1 \le r \le k) \\ y = a_1\, \mathsf{Y}(t_1) \cdots a_k\, \mathsf{Y}(t_k) a_{k+1} \end{array} \right. \\
&= \sum_{\substack{j\in J \\ t_1,\ldots,t_k}} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ t_1,\ldots,t_k \text{ trees with } \lambda_v(t_r) = X_r,\ d(t_r) \le i, \quad (1 \le r \le k) \\ \qquad \text{such that at most one of the } t_r \text{ with } d(t_r) = i \\ y = a_1\, \mathsf{Y}(t_1) \cdots a_k\, \mathsf{Y}(t_k) a_{k+1} \end{array} \right. \\
&\quad + \sum_{\substack{j\in J \\ t_1,\ldots,t_k}} y : \left\{ \begin{array}{l} m_{X,j} = a_1 X_1 \cdots a_k X_k a_{k+1} \\ t_1,\ldots,t_k \text{ trees with } \lambda_v(t_r) = X_r,\ d(t_r) \le i, \quad (1 \le r \le k) \\ \qquad \text{such that at least two of the } t_r \text{ with } d(t_r) = i \\ y = a_1\, \mathsf{Y}(t_1) \cdots a_k\, \mathsf{Y}(t_k) a_{k+1} \end{array} \right. \\
&= \sum_t \mathsf{Y}(t) : t \text{ tree with } \lambda_v(t) = X,\ d(t) \le i \\
&\quad + \sum_t \mathsf{Y}(t) : t \text{ tree with } \lambda_v(t) = X,\ dl(t) = (i+1, 0)
\end{aligned}$$

(by Lemma 3.2.14 resp. Remark A.1.2)

$$\begin{aligned}
&= \boldsymbol{\tau}^{(i)}_X + \mathsf{Y}(P^{i+1}_X) \\
&= \boldsymbol{\tau}^{(i)}_X + \boldsymbol{\delta}^{(i)}_X
\end{aligned}$$

□

A.2 Proofs of Section 3.3.1

Theorem 3.3.4.
Let $G = (\mathcal{X}, \Sigma, P, X_0)$ be a context-free grammar in CNF and let $(\boldsymbol{\nu}^{(i)})_{i\in\mathbb{N}}$ be the Newton sequence associated with G. Then $\left(\boldsymbol{\nu}^{(i)}\right)_{X_0} = L_{i+1}(G)$ for every $i \ge 0$. ◇

The proof of Theorem 3.3.4 follows from Theorem 3.2.11 and the following two lemmata.

Lemma A.2.1.
Let $G = (\mathcal{X}, \Sigma, P, X_0)$ be a context-free grammar in CNF. Let $w \in \Sigma^*$ be derivable from X by an index-i derivation. Then there is an X-tree t having yield $Y(t) = w$ and dimension $d(t) < i$. ◇

Proof. Let D be a derivation of w. One can associate a derivation tree t to D in the obvious way. We show by induction on i and on the height of t that $d(t) \geq i$ implies $ind(D) > i$, where $ind(D)$ denotes the index of D. The base case $i = 0$ is trivial, because any derivation has index at least 1. The other base case $i = 1$ implies that t has two children, hence $ind(D) \geq 2$. Let $i > 1$ and $d(t) \geq i$. Then t has two children t_1, t_2. By definition of dimension, either $d(t_1) \geq i - 1$ and $d(t_2) \geq i - 1$ or $d(t_1) \geq i$.

- In the first case, the very first step of D already produces two variables $\lambda_v(t_1)$ and $\lambda_v(t_2)$. Since $d(t_1) \geq 1$ and $d(t_2) \geq 1$, neither of those two variables can be derived to a terminal word immediately. So the most "economical" way to continue the derivation is to finish the derivation of $\lambda_v(t_1)$ or $\lambda_v(t_2)$ before touching the other variable. But, by induction on i, any subderivations of D that "flatten" t_1 and t_2 have indices at least i. Hence $ind(D) > i$.
- In the second case, any subderivation of D that "flattens" t_1 has, by induction on the height, index greater than i. So, D itself cannot have a smaller index. □

Lemma A.2.2.
Let $G = (\mathcal{X}, \Sigma, P, X_0)$ be a context-free grammar. Let m be the largest number of nonterminals in the right-hand sides of P. Let t be an X-tree having yield $\mathsf{Y}(t) = w$ and dimension $d(t) = i$. Then there is a derivation of w from X with index at most $i \cdot (m - 1) + 1$. ◇

Proof. The sought derivation D can be constructed by "flattening" the derivation tree t according to a certain strategy. The first step of D is $\lambda_v(t) \Rightarrow \lambda_m(t)$. After that, the strategy is to completely flatten each subtree of t in the order of increasing dimension. We prove by induction on i and on the height of t that this yields $ind(D) \leq i \cdot (m - 1) + 1$. The base case $i = 0$ is clear. Let $i > 0$ and $t_1, \ldots, t_k$ $(k \leq m)$ be the subtrees of t ordered by increasing dimension. During the flattening of t_j, at most $m - 1$ nonterminals, namely $\lambda_v(t_{j+1}), \ldots, \lambda_v(t_k)$, stick around. The trees $t_1, \ldots, t_{k-1}$ have dimension at most $i-1$. By induction on i, they can be flattened to derivations with index at most $(i-1) \cdot (m-1)+1$. So, during the flattening of $t_1, \ldots, t_{k-1}$ the index of D grows to at most $(i-1) \cdot (m-1)+1+(m-1) = i \cdot (m-1)+1$. The tree t_k has dimension at most i. By induction on the height, t_k can be flattened to a derivation with index at most $i \cdot (m - 1) + 1$. During the flattening of t_k, no other nonterminals stick around. So, the index of D does not grow over $i \cdot (m - 1) + 1$. □

A.3 Redko's Theorem and Commutative Kleene Algebras

There is a number of inequivalent definitions of Kleene algebras. This includes *C-algebras* and *Kleene algebras in the sense of Kozen* the latter of which we simply refer to as *Kleene algebras*.

Both definitions require an algebraic structure $(K, +, \cdot, {}^*, 0, 1)$ that is an idempotent semiring under $+, \cdot, 0, 1$. In addition, different sets of axioms are required.

A C-algebra [Con71] must satisfy the following axioms:

$$\begin{array}{ll} C11 & (a+b)^* = (a^*b)^*a^* \\ C12 & (ab)^* = 1 + a(ba)^*b \\ C13 & (a^*)^* = a^* \\ C14.n & a^* = a^{n*}a^{<n} \quad (n > 0). \end{array}$$

A Kleene algebra [Koz91] on the other hand must satisfy the following axioms:

$$\begin{array}{lllll} K1 & 1 + aa^* \leq a^* & & \\ K2 & 1 + a^*a \leq a^* & & \\ K3 & a + bc \leq c & \rightarrow & b^*a \leq c \\ K4 & a + cb \leq c & \rightarrow & ab^* \leq c, \end{array}$$

where $\leq$ refers to the natural partial order on K.

It was shown in [Koz91] that the axioms of Kleene algebra are *complete* for the algebra of regular languages. That means, if an equation $\alpha = \beta$ between regular expressions holds under the canonical interpretation over the regular languages, then it holds in any Kleene algebra. It is easy to see that equations $C11$ – $C14$ hold under the canonical interpretation. Therefore any Kleene algebra is a C-algebra.

The axioms of C-algebra are not complete, i.e., they are too weak to derive some equation valid under the canonical interpretation [Con71]. However, if two more axioms (C^+1 and C^+2, see below) describing commutativity are added, the resulting system of axioms (defining *commutative C-algebras*) becomes complete for the algebra of commutative regular languages. In other words, if the Parikh images of languages $L(\alpha)$ and $L(\beta)$ are equal, then $\alpha = \beta$ can be proved using only the axioms of commutative C-algebras. The additional axioms are:

$$\begin{array}{ll} C^+1 & ab = ba \\ C^+2 & a^*b^* = (ab)^*(a^* + b^*). \end{array}$$

The completeness of commutative C-algebras is called *Redko's theorem*. Conway's monograph [Con71] contains a proof of this theorem.

We want to show that the system of axioms of Kleene algebra plus the commutativity axiom $ab = ba$ (defining *commutative Kleene algebras*) is complete for commutative regular languages as well. Appealing to Redko's theorem, we only have to show that equation C^+2 is a theorem of commutative Kleene algebra.

We use the identity $a^*b^* = (a+b)^*$ which is a theorem of commutative Kleene algebra [HK99]. Since $(a+b)^* \geq (ab)^*(a^*+b^*)$ holds in any Kleene algebra, we only need to show $(a+b)^* \leq (ab)^*(a^*+b^*)$. With $K3$ it suffices to show

$$1 + (a+b)(ab)^*(a^*+b^*) \leq (ab)^*(a^*+b^*).$$

We show this inequality for each term of the sum at the left hand side. For 1 it obviously holds. We also have $a(ab)^*a^* = (ab)^*aa^* \leq (ab)^*a^*$ using commutativity and $K1$. Similarly, $a(ab)^*b^* = (ab)^*ab^* = (ab)^*a + (ab)^*abb^* \leq (ab)^*a + (ab)^*b^* \leq (ab)^*(a^*+b^*)$. Here we used that $b^* = 1 + bb^*$ is a theorem of Kleene algebra. The other inequalities follow symmetrically.

Appendix B

Missing Proofs of Chapter 4

B.1 Proofs of Section 4.2

Theorem 4.2.3.
Let $\boldsymbol{f}$ be a system of polynomials over an io-semiring. For every variable X of $\boldsymbol{f}$ we have $\mathsf{Y}(\mathcal{B}_X) = (\mu \boldsymbol{f}_{\mathcal{B}})_X$, i.e., the yield of the X-bamboos is equal to the X-component of the least solution of the bamboo system. ◇

Proof. By definition, we have

$$\boldsymbol{f}_{\mathcal{B}}(\boldsymbol{X}) = \boldsymbol{f}(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}(\boldsymbol{X}).$$

Its Kleene sequence, thus, becomes

$$\begin{array}{rcl}
\boldsymbol{f}_{\mathcal{B}}(\boldsymbol{0}) & = & \boldsymbol{f}(\boldsymbol{0}) \\
\boldsymbol{f}^2_{\mathcal{B}}(\boldsymbol{0}) & = & \boldsymbol{f}(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big) \\
\boldsymbol{f}^3_{\mathcal{B}}(\boldsymbol{0}) & = & \boldsymbol{f}(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big) + \underbrace{D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}\big(D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big)\big)}_{=:D\boldsymbol{f}|^2_{\boldsymbol{f}^n(\boldsymbol{0})}\left(\boldsymbol{f}(\boldsymbol{0})\right)} \\
& \vdots & \\
\boldsymbol{f}^k_{\mathcal{B}}(\boldsymbol{0}) & = & \boldsymbol{f}(\boldsymbol{0}) + D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big) + \ldots + D\boldsymbol{f}|^{k-1}_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big).
\end{array}$$

As $\mu\boldsymbol{f}_{\mathcal{B}} = \sup \boldsymbol{f}^k_{\mathcal{B}}(\boldsymbol{0})$, we get

$$\mu\boldsymbol{f}_{\mathcal{B}} = \sum_{k\in\mathbb{N}} D\boldsymbol{f}|^k_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big),$$

where $D\boldsymbol{f}|^k_{\boldsymbol{f}^n(\boldsymbol{0})}$ denotes the k-fold application of $D\boldsymbol{f}|_{\boldsymbol{f}^n(\boldsymbol{0})}$. In particular, we have $D\boldsymbol{f}|^0_{\boldsymbol{f}^n(\boldsymbol{0})}\big(\boldsymbol{f}(\boldsymbol{0})\big) = \boldsymbol{f}(\boldsymbol{0})$.

Let $\mathcal{B}_X^{(h)}$ be the set of X-bamboos (w.r.t. $\boldsymbol{f}$) of height *at most* h. Similarly, $\mathcal{B}_X$ denotes the set of all bamboos of height at most h.

$\sqsupseteq$-direction: We first prove that

$$\mathsf{Y}(\mathcal{B}_X) \sqsupseteq \big(\mu \boldsymbol{f}_{\mathcal{B}}\big)_X.$$

It suffices to show (for $h \geq 0$) that

$$\mathsf{Y}(\mathcal{B}_X^{(h+n-1)}) \sqsupseteq \Big(D\boldsymbol{f}|^h_{\boldsymbol{f}^n(\mathbf{0})}\big(\boldsymbol{f}^n(\mathbf{0})\big)\Big)_X$$

holds for all $X \in \mathcal{X}$ as this immediately implies

$$\begin{array}{rcl} \mathsf{Y}(\mathcal{B}_X) & = & \sum_{h\in\mathbb{N}} \mathsf{Y}(\mathcal{B}_X^{(h+n-1)}) \\ & \sqsupseteq & \sum_{h\in\mathbb{N}} \Big(D\boldsymbol{f}|^h_{\boldsymbol{f}^n(\mathbf{0})}\big(\boldsymbol{f}^n(\mathbf{0})\big)\Big)_X \\ & \sqsupseteq & \sum_{h\in\mathbb{N}} \Big(D\boldsymbol{f}|^h_{\boldsymbol{f}^n(\mathbf{0})}\big(\boldsymbol{f}(\mathbf{0})\big)\Big)_X \\ & = & \big(\mu \boldsymbol{f}_{\mathcal{B}}\big)_X. \end{array}$$

We proceed by induction on h: For $h = 0$, we have $\mathcal{B}_X^{(n-1)} = \mathcal{H}_X^{(n-1)}$, and $\mathsf{Y}(\mathcal{H}_X^{(n-1)}) = \boldsymbol{f}^n(\mathbf{0})_X$ (cf. prop. 3.2.5). Thus,

$$\mathsf{Y}(\mathcal{B}_X^{(n-1)}) = \mathsf{Y}(\mathcal{H}_X^{(n-1)}) = \boldsymbol{f}^n(\mathbf{0})_X = \Big(D\boldsymbol{f}|^0_{\boldsymbol{f}^n(\mathbf{0})}\big(\boldsymbol{f}^n(\mathbf{0})\big)\Big)_X$$

follows immediately.

Consider therefore $\Big(D\boldsymbol{f}|^{h+1}_{\boldsymbol{f}^n(\mathbf{0})}\big(\boldsymbol{f}^n(\mathbf{0})\big)\Big)_X$ for $h \geq 0$, and let $\mathsf{Y}(\mathcal{B}^{(h)})$ denote the vector defined by $\mathsf{Y}(\mathcal{B}^{(h)})_X := \mathsf{Y}(\mathcal{B}_X^{(h)})$. We then have by induction on h that

$$\Big(D\boldsymbol{f}|^{h+1}_{\boldsymbol{f}^n(\mathbf{0})}\big(\boldsymbol{f}^n(\mathbf{0})\big)\Big)_X \sqsubseteq \Big(D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big)\Big)_X$$

Assume that $\boldsymbol{f}_X = \sum_{i=1}^k m_i$ where $m_1, \ldots, m_k$ are monomials. As addition is idempotent, we may assume these monomials are pairwise different. By definition, we have

$$\begin{array}{rcl} \Big(D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big)\Big)_X & = & D\boldsymbol{f}_X|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big) \\ & = & \sum_{i=1}^k Dm_i|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big). \end{array}$$

If all monomials $m_1, \ldots, m_k$ have degree 0, then $D\boldsymbol{f}_X|_{\boldsymbol{v}}(\boldsymbol{a}) = 0$ for all $\boldsymbol{v}, \boldsymbol{a} \in V$. But this also implies that $\mathcal{B}^{(h)} = \mathcal{H}_X^{(0)}$ also holds for all $h \geq 1$, hence, we may assume that there is at least one monomial of degree at least one. Let $m \in \{m_1, \ldots, m_k\}$ be such a monomial with $m = a_1 X_1 a_2 \ldots X_l a_{l+1}$ (for some $l \geq 1$). We then have

$$Dm|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big) = \sum_{Y\in\mathcal{X}} D_Y m|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big)$$

by definition. Consider a $Y \in \{X_1, \ldots, X_l\}$, i.e. a variable appearing in m (for the remaining variables, the differential is again 0), and let $\mathrm{pos}_Y(m) = \{i \mid X_i = Y\}$ be the

set of "positions of Y in m". We then may write

$$
\begin{aligned}
& D_Y m|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big) \\
= & \sum_{p\in \mathrm{pos}_Y(m)} \left(\prod_{q=1}^{p-1} a_q \cdot \boldsymbol{f}^n(\mathbf{0})_{X_q}\right) \cdot a_p \cdot \mathsf{Y}(\mathcal{B}^{(h+n-1)})_{X_p} \cdot \left(\prod_{q=p+1}^{l} a_q \cdot \boldsymbol{f}^n(\mathbf{0})_{X_q}\right) \cdot a_{l+1} \\
= & \sum_{p\in \mathrm{pos}_Y(m)} \left(\prod_{q=1}^{p-1} a_q \cdot \boldsymbol{f}^n(\mathbf{0})_{X_q}\right) \cdot a_p \cdot \mathsf{Y}(\mathcal{B}^{(h+n-1)}_{X_p}) \cdot \left(\prod_{q=p+1}^{l} a_q \cdot \boldsymbol{f}^n(\mathbf{0})_{X_q}\right) \cdot a_{l+1} \\
= & \sum_{p\in \mathrm{pos}_Y(m)} \left(\prod_{q=1}^{p-1} a_q \cdot \mathsf{Y}(\mathcal{H}^{(n-1)})_{X_q}\right) \cdot a_p \cdot \mathsf{Y}(\mathcal{B}^{(h+n-1)}_{X_p}) \cdot \left(\prod_{q=p+1}^{l} a_q \cdot \mathsf{Y}(\mathcal{H}^{(n-1)})_{X_q}\right) \cdot a_{l+1} \\
= & \sum_{p\in \mathrm{pos}_Y(m)} \left(\prod_{q=1}^{p-1} a_q \cdot \mathsf{Y}(\mathcal{H}^{(n-1)}_{X_q})\right) \cdot a_p \cdot \mathsf{Y}(\mathcal{B}^{(h+n-1)}_{X_p}) \cdot \left(\prod_{q=p+1}^{l} a_q \cdot \mathsf{Y}(\mathcal{H}^{(n-1)}_{X_q})\right) \cdot a_{l+1} \\
= & \sum_{p\in \mathrm{pos}_Y(m)} \left(\prod_{q=1}^{p-1} a_q \cdot \sum_{t\in\mathcal{H}^{(n-1)}_{X_q}} \mathsf{Y}(t)\right) \cdot a_p \cdot \left(\sum_{t\in\mathcal{B}^{(h+n-1)}_{X_p}} \mathsf{Y}(t)\right) \cdot \left(\prod_{q=p+1}^{l} a_q \cdot \sum_{t\in\mathcal{H}^{(n-1)}_{X_q}} \mathsf{Y}(t)\right) \cdot a_{l+1}
\end{aligned}
$$

But this last sum is simply the yield of all X-bamboos $t \in \mathcal{B}^{(h+n)}_X$ with $\lambda_2(t) = m$ having height at least 1, and at most $h+n$. As for $t \in \mathcal{B}^{(h)}_X$ we have $\lambda_2(t) \in \{m_1, \ldots, m_k\}$, we get by idempotent addition

$$\Big(D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mathsf{Y}(\mathcal{B}^{(h+n-1)})\big)\Big)_X \sqsubseteq \mathsf{Y}(\mathcal{B}^{(h+n)}_X).$$

$\sqsubseteq$-direction: We now turn to the proof of

$$\mathsf{Y}(\mathcal{B}_X) \sqsubseteq \big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_X.$$

As addition is idempotent, it suffices to show that

$$\mathsf{Y}(t) \sqsubseteq \big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_X.$$

We proceed by induction on the number of nodes in t. If t has just one node then $\mathsf{Y}(t) \sqsubseteq (\boldsymbol{f}(\mathbf{0}))_X \sqsubseteq \big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_X$. For the induction step, t has children. So assume w.l.o.g. that $\lambda_2(t) = a_1 X_1 \cdots X_s a_{s+1}$ for some $s \geq 1$. Denote the children of t by $t_1, \ldots, t_s$. Furthermore we assume w.l.o.g. that the backbone of t goes through t_1. Hence, t_1 is itself a bamboo having less nodes than t. By induction we have $\mathsf{Y}(t_1) \sqsubseteq \big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_{X_1}$. As t is a bamboo, the other children $t_2, \ldots, t_s$ have a height of at most $n-1$. It is easy to see (cf. [EKL07a]) that this implies

$$\mathsf{Y}(t_r) \sqsubseteq (\boldsymbol{f}^n(\mathbf{0}))_{X_r} \text{ for all } 2 \leq r \leq s\,. \tag{B.1}$$

Now we have:

$$
\begin{aligned}
\mathsf{Y}(t) &= a_1\,\mathsf{Y}(t_1)\cdots\mathsf{Y}(t_s)a_{s+1} && \text{(def. of yield } \mathsf{Y}\text{)} \\
&\sqsubseteq a_1\big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_{X_1} a_2\,\mathsf{Y}(t_2)\cdots\mathsf{Y}(t_s)a_{s+1} && \text{(by induction)} \\
&\sqsubseteq a_1\big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_{X_1} a_2(\boldsymbol{f}^n(\mathbf{0}))_{X_2}\cdots(\boldsymbol{f}^n(\mathbf{0}))_{X_s}a_{s+1} && \text{(Equation (B.1))} \\
&\sqsubseteq D_{X_1}(a_1X_1\cdots X_s a_{s+1})|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mu\boldsymbol{f}_{\mathcal{B}}\big) && \text{(def. of differentials)} \\
&\sqsubseteq D_{X_1}\boldsymbol{f}_X|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mu\boldsymbol{f}_{\mathcal{B}}\big) && (t \in \mathcal{B}_X) \\
&\sqsubseteq D\boldsymbol{f}_X|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mu\boldsymbol{f}_{\mathcal{B}}\big) && \text{(def. of differentials)} \\
&= \big(D\boldsymbol{f}|_{\boldsymbol{f}^n(\mathbf{0})}\big(\mu\boldsymbol{f}_{\mathcal{B}}\big)\big)_X && \text{(def. of differentials)} \\
&\sqsubseteq \big(\mu\boldsymbol{f}_{\mathcal{B}}\big)_X && \square
\end{aligned}
$$

B.2 Proofs of Section 4.3

Proposition 4.3.3.
In any star-distributive semiring, the following equations hold:

(1) $a^*b^* = a^* + b^*$.

(2) $(ab^*)^* = a^* + ab^*$. ◇

Proof.

(1) The equation $a^*b^* = (a+b)^*$ holds in any commutative idempotent semiring. By star-distributivity, $(a+b)^* = a^* + b^*$.

(2) In any commutative io-semiring, we have $(ab^*)^* = 1 + aa^*b^*$ (see e.g. [HK99]). By (1), we have $1 + aa^*b^* = 1 + aa^* + ab^* = a^* + ab^*$. □

Theorem 4.3.4.
For any polynomial system $\boldsymbol{f}$ over a star-distributive semiring $\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$ holds. ◇

Proof. We will need the following notation: If t' is a subtree of a derivation tree t, we write $t = \widehat{t} \cdot t'$ where $\widehat{t}$ is the partial derivation tree obtained from t by removing t'. If, in addition, $t' = \widehat{t'} \cdot t''$, and t' and t'' have the same variable-label. we say the decomposition $t = \widehat{t} \cdot \widehat{t'} \cdot t''$ is *pumpable*, because $\widehat{t} \cdot (\widehat{t'})^i \cdot t''$ is a valid tree for all $i \geq 0$. We define $\widehat{t} \cdot (\widehat{t'})^* \cdot t'' = \{\widehat{t} \cdot (\widehat{t'})^i \cdot t'' \mid i \geq 0\}$. Notice that, due to commutativity of product, it holds $\mathsf{Y}(\widehat{t} \cdot (\widehat{t'})^* \cdot t'') = \mathsf{Y}(\widehat{t}) \cdot \mathsf{Y}(\widehat{t'})^* \cdot \mathsf{Y}(t'')$. We call this yield the *pumping yield* of the decomposition $t = \widehat{t} \cdot \widehat{t'} \cdot t''$.

The proof is by derivation tree analysis. So it suffices to discharge the precondition of Corollary 4.2.4. More precisely we need to show that, for any X-tree t, we have $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$. If t does not have a pumpable decomposition, then t has a height of at most $n-1$, hence $t \in \mathcal{B}_X$ and so $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$. It remains to show: if t has a pumpable decomposition $t = \widehat{t} \cdot \widehat{t_1} \cdot t_1'$, then $\mathsf{Y}(t) \sqsubseteq \mathcal{B}_X$. In fact, we show $\mathsf{Y}(\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1') \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$, which is stronger because $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1')$.

Denote by $\#(t)$ the number of nodes in a tree t. We assign to a pumpable decomposition $t = \widehat{t} \cdot \widehat{t_1} \cdot t_1'$ a *size* by setting $\mathit{size}(t = \widehat{t} \cdot \widehat{t_1} \cdot t_1') = (\#(t), \#(\widehat{t_1} \cdot t_1'))$. We define a total order on sizes by setting $(i,j) \lhd (i',j')$ if either $i < i'$ or $i = i'$ and $j < j'$. We use this order to prove by induction that for any size (i,j), if there is a pumpable decomposition $t = \widehat{t} \cdot \widehat{t_1} \cdot t_1'$ of size (i,j), then $\mathsf{Y}(\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1') \sqsubseteq \mathsf{Y}(\mathcal{B}_X)$.

The induction base is trivial because trees t with $\#(t) = 1$ do not have a decomposition. For the induction step, let t be an X-tree and let $t = \widehat{t} \cdot \widehat{t_1} \cdot t_1'$ be pumpable. Choose a path p in t from the root to a leaf through t_1'. If p is a valid stem of an X-bamboo, then all trees in $\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1'$ are X-bamboos, so $\mathsf{Y}(\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1') \sqsubseteq \mathcal{B}_X$. Hence, assume that p is not a valid stem, i.e., there is some subtree of t, disjoint from p, with height at least n. So this tree has a subtree $t_2 = \widehat{t_2} \cdot t_2'$ such that t_2 and t_2' have the same variable-label. We distinguish two cases.

(a) t_2 *is not* a subtree of $\widehat{t_1}$. Then $\widehat{t_1}$ and $\widehat{t_2}$ are disjunct and so there exists a $\widetilde{y}$ such that $\mathsf{Y}(t) = \widetilde{y} \cdot \mathsf{Y}(\widehat{t_1}) \cdot \mathsf{Y}(\widehat{t_2})$. Then:

$$\begin{aligned}
\mathsf{Y}(\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1') &= \widetilde{y} \cdot \mathsf{Y}(\widehat{t_1})^* \cdot \mathsf{Y}(\widehat{t_2}) \\
&\sqsubseteq \widetilde{y} \cdot \mathsf{Y}(\widehat{t_1})^* \cdot \mathsf{Y}(\widehat{t_2})^* && \text{(def. of Kleene } {}^*\text{)} \\
&= \widetilde{y} \cdot \mathsf{Y}(\widehat{t_1})^* + \widetilde{y} \cdot \mathsf{Y}(\widehat{t_2})^* && \text{(Prop. 4.3.3 (1))}
\end{aligned}$$

The expression $\widetilde{y} \cdot \mathsf{Y}(\widehat{t_1})^*$ equals the pumping yield of a decomposition of an X-tree which is obtained from t by removing the substructure $\widehat{t_2}$. Similarly, the expression $\widetilde{y} \cdot \mathsf{Y}(\widehat{t_2})^*$ is equal to the pumping yield of a decomposition of an X-tree which is obtained from t by removing the substructure $\widehat{t_1}$. By induction on the size, both of those pumping yields are $\sqsubseteq \mathsf{Y}(\mathcal{B}_X)$.

(b) t_2 *is* a subtree of $\widehat{t_1}$. Then we can write $\widehat{t_1} = \widehat{\widehat{t_1}} \cdot \widehat{t_2} \cdot t_2'$. We have:

$$\begin{aligned}
\mathsf{Y}(\widehat{t} \cdot (\widehat{t_1})^* \cdot t_1') &= \mathsf{Y}(\widehat{t}) \cdot \mathsf{Y}(\widehat{\widehat{t_1}} \cdot \widehat{t_2} \cdot t_2')^* \cdot \mathsf{Y}(t_1') \\
&= \mathsf{Y}(\widehat{t}) \cdot \Big(\mathsf{Y}(\widehat{\widehat{t_1}}) \cdot \mathsf{Y}(\widehat{t_2}) \cdot \mathsf{Y}(t_2')\Big)^* \cdot \mathsf{Y}(t_1') \\
&\sqsubseteq \mathsf{Y}(\widehat{t}) \cdot \Big(\mathsf{Y}(\widehat{\widehat{t_1}}) \cdot \mathsf{Y}(\widehat{t_2})^* \cdot \mathsf{Y}(t_2')\Big)^* \cdot \mathsf{Y}(t_1') \\
&= \left\{ \begin{array}{l} \mathsf{Y}(\widehat{t}) \cdot \mathsf{Y}(\widehat{\widehat{t_1}}) \cdot \mathsf{Y}(\widehat{t_2})^* \cdot \mathsf{Y}(t_2') \cdot \mathsf{Y}(t_1') + \\ \mathsf{Y}(\widehat{t}) \cdot \Big(\mathsf{Y}(\widehat{\widehat{t_1}}) \cdot \mathsf{Y}(t_2')\Big)^* \cdot \mathsf{Y}(t_1') \end{array} \right\} && \text{(Prop. 4.3.3 (2))} \\
&= \left\{ \begin{array}{l} \mathsf{Y}(\widehat{t} \cdot (\widehat{\widehat{t_1}} \cdot \widehat{t_2}^* \cdot t_2') \cdot t_1') + \\ \mathsf{Y}(\widehat{t} \cdot (\widehat{\widehat{t_1}} \cdot t_2')^* \cdot t_1') \end{array} \right\}
\end{aligned}$$

The first expression in this sum equals $\mathsf{Y}((\widehat{t} \cdot \widehat{\widehat{t_1}} \cdot t_1') \cdot \widehat{t_2}^* \cdot t_2')$. This is the pumping yield of the decomposition $t = (\widehat{t} \cdot \widehat{\widehat{t_1}} \cdot t_1') \cdot \widehat{t_2} \cdot t_2'$. Since $t_2 = \widehat{t_2} \cdot t_2'$ is a proper subtree of $\widehat{t_1} \cdot t_1'$, it has fewer nodes than $\widehat{t_1} \cdot t_1'$. So this decomposition is smaller (in the second component), i.e., by induction, the first expression in the above sum is $\sqsubseteq \mathsf{Y}(\mathcal{B}_X)$.

The second expression in the above sum equals the pumping yield of the decomposition of an X-tree, which is obtained from t by removing the substructure $\widehat{t_2}$. By induction, this pumping yield is $\sqsubseteq \mathsf{Y}(\mathcal{B}_X)$. □

B.3 Proofs of Section 4.4

Lemma 4.4.3.
For every clean $\boldsymbol{g} \in \mathcal{S}[\mathcal{X}]^{\mathcal{X}}$ we can construct in linear time a system $\boldsymbol{f} \in \mathcal{S}[\mathcal{X}']^{\mathcal{X}'}$ in quadratic normal form such that $\mathcal{X} \subseteq \mathcal{X}'$ and $\mu\boldsymbol{g}_X = \mu\boldsymbol{f}_X$ for all $X \in \mathcal{X}$. ◇

Proof. We first transform $\boldsymbol{g}$ into Chomsky normal-form, which gives us a system $\boldsymbol{g}'$ over the same semiring. As the transformation into Chomsky normal-form introduces new variables, $\boldsymbol{g}'$ is given in a super set $\mathcal{X}'$ of $\mathcal{X}$ with $\mu\boldsymbol{g}'_X = \mu\boldsymbol{g}_X$ for all $X \in \mathcal{X}$. Next, as $\boldsymbol{g}$

is clean, we can ensure that $\boldsymbol{g}'$ is clean, too. We therefore may set $\boldsymbol{g}'' := \boldsymbol{g}' + \mathbf{1}$ without changing the least solution. Hence, every polynomial of $\boldsymbol{g}''_X$ has the form

$$c^{(X)} + \sum_{Y,Z \in \mathcal{X}'} a^{(X)}_{Y,Z} \cdot Y \cdot Z \text{ with } 1 \sqsubseteq c \text{ and } a_{Y,Z} \in \{0,1\}.$$

Finally, as $\mathbf{1} \sqsubseteq \mu\boldsymbol{g}''$ we have

$$\begin{aligned}
\mu\boldsymbol{g}''_X &= \boldsymbol{g}''_X(\mu\boldsymbol{g}''_X) \\
&= c^{(X)} + \sum_{Y,Z \in \mathcal{X}'} a^{(X)}_{Y,Z} \cdot \mu\boldsymbol{g}''_Y \cdot \mu\boldsymbol{g}''_Z \\
&= c^{(X)} + \sum_{Y,Z \in \mathcal{X}'} a^{(X)}_{Y,Z} \cdot (1 + \mu\boldsymbol{g}''_Y) \cdot (1 + \mu\boldsymbol{g}''_Z) \\
&= c^{(X)} + 1 + \sum_{Y,Z \in \mathcal{X}'} a^{(X)}_{Y,Z} \cdot \mu\boldsymbol{g}''_Y \cdot \mu\boldsymbol{g}''_Z + \sum_{Y \in \mathcal{X}'} \left(\sum_{Z \in \mathcal{X}'} a^{(X)}_{Y,Z} + a^{(X)}_{Z,Y} \right) \cdot \mu\boldsymbol{g}''_Y \\
&= \boldsymbol{g}''_X(\mu\boldsymbol{g}''_X) + 1 + \sum_{Y \in \mathcal{X}'} \left(\sum_{Z \in \mathcal{X}'} a^{(X)}_{Y,Z} + a^{(X)}_{Z,Y} \right) \cdot \mu\boldsymbol{g}''_Y.
\end{aligned}$$

We now define $\boldsymbol{f}$ by setting for all $X \in \mathcal{X}'$

$$\boldsymbol{f}_X := \boldsymbol{g}''_X + 1 + \sum_{Y \in \mathcal{X}'} \left(\sum_{Z \in \mathcal{X}'} a^{(X)}_{Y,Z} + a^{(X)}_{Z,Y} \right) \cdot Y.$$

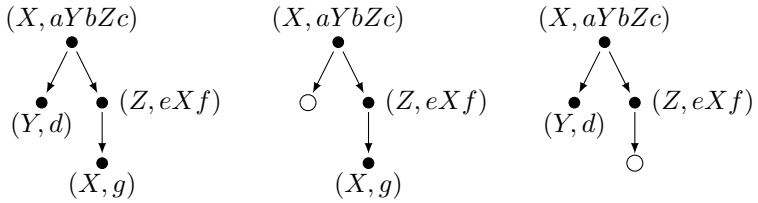
We then have $\boldsymbol{g}'' \sqsubseteq \boldsymbol{f}$, and, thus, $\mu\boldsymbol{g}'' \sqsubseteq \mu\boldsymbol{f}$, but also $\boldsymbol{f}(\mu\boldsymbol{g}'') = \mu\boldsymbol{g}''$, i.e. $\mu\boldsymbol{g}'' = \mu\boldsymbol{f}$. □

For the proof of Theorem 4.4.4, we first define partial derivation trees. Intuitively, they are the result of removing exactly one subtree from a derivation tree, leaving a "dangling pointer".

Definition B.3.1.
Let $\boldsymbol{f} \in \mathcal{S}[\mathcal{X}]^{\mathcal{X}}$. Let t be some X-tree for $X \in \mathcal{X}$. Further, let $Y \in \mathcal{X}$ be some variable such that t has at least one leaf s with $\lambda_v(s) = Y$. By erasing exactly one such leaf s from t, we obtain an *XY-tree*. We write $\mathcal{T}_{X,Y}$ for the set of all XY-trees.

The set $\mathcal{B}_{X,Y}$ is defined similarly. A tree $t' \in \mathcal{B}_{X,Y}$ results from a tree $t \in \mathcal{B}_X$ by removing exactly one such leaf s having the following properties: (i) $\lambda_v(s) = Y$ and (ii) the path from t' to s has maximal length. ◇

Example B.3.2. Consider the X-tree depicted on the left. By deleting the leaf labeled by (Y, d), we obtain the XY-tree depicted in the middle, where we represent the missing leaf/subtree by ○. Similarly, we obtain the XX-tree shown on the right by deleting the leaf labeled by (X, g).



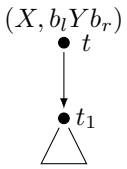
Note that we can replace $\circ$ in the XY-tree by any Y-tree in order to obtain a valid X-tree, again. In other words, the yield of an XY-tree is a linear monomial in Y.

We now can state our main theorem.

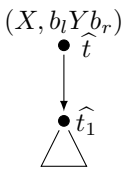
Theorem 4.4.4.
For a finite set of variables $\mathcal{X}$ ($n := |\mathcal{X}|$), let $\mathcal{S}$ be a lossy semiring, and $\boldsymbol{f} \in \mathcal{S}[\mathcal{X}]^{\mathcal{X}}$ a clean and strongly-connected system of polynomials in normal-form. We then have $\mu\boldsymbol{f} = \mu\boldsymbol{f}_{\mathcal{B}}$. ◇

Proof. We again show that we can transform any X-tree t w.r.t. $\boldsymbol{f}$ into a tree $\hat{t}$ contained in $\mathcal{B}_X$ with $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(\hat{t})$. We proceed by induction on the number N of nodes of t. If $N = 1$, then t has height 0. By definition, we have $t \in \mathcal{B}_X$, so we are done.

Therefore assume $N > 1$. As $\boldsymbol{f}$ is in normal form, we either have $\lambda_m(t) = b_l Y b_r$ or $\lambda_m(t) = YZ$ for some $Y, Z \in \mathcal{X}$, and $b_l, b_r \in S \setminus \{0\}$. If t is labeled by $\lambda_m(t) = b_l Y b_r$,

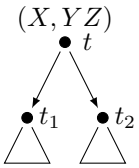


then t has exactly one child t_1, which immediately can be replaced by some tree $\widehat{t_1}$ in $\mathcal{B}_Y$ with $\mathsf{Y}(t_1) = \mathsf{Y}(\widehat{t_1})$ because of induction. This gives us the tree $\widehat{t}$

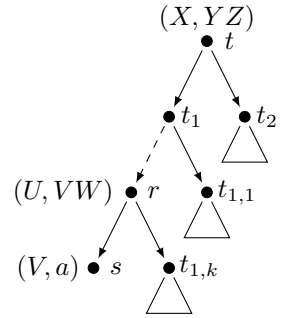


and $\mathsf{Y}(\widehat{t}) = b_l\, \mathsf{Y}(\widehat{t_1}) b_r = b_l\, \mathsf{Y}(t_1) b_r = \mathsf{Y}(t)$.

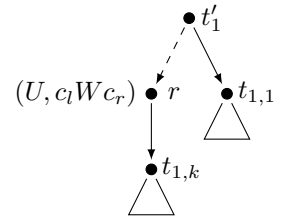
Hence, assume that $\lambda_m(t) = YZ$, i.e. t has two children t_1, t_2.



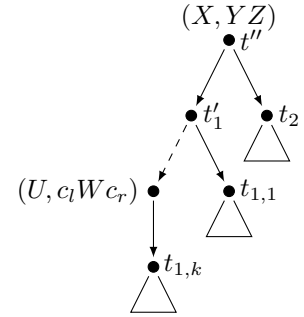
Descending into t_1 by always taking the left most child, we end up at the left most leaf s of t. We denote by $t_{1,1}$ to $t_{1,k}$ the "right" children of the nodes located on the path from t_1 to s for some $k \in \mathbb{N}$. Let r then be the father of s with $\lambda_v(s) = V$, and $\lambda_m(s) = a \in S$. We assume that $\lambda_m(r) = VW$ for some $W \in \mathcal{X}$



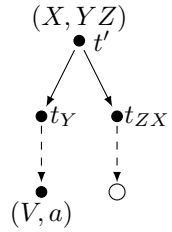
As $\boldsymbol{f}$ is in normal form, and VW is a monomial of $\boldsymbol{f}_U$, there exists also a monomial $c_l W c_r$ appearing in $\boldsymbol{f}_U$ for some $c_l, c_r \in S \setminus \{0\}$. We first remove from t_1 the leaf s, and relabel the node r by setting $\lambda_m(r) := c_l W c_r$. This gives us the tree t_1' with $\mathsf{Y}(t_1) \sqsubseteq a \cdot \mathsf{Y}(t_1')$, as $1 \sqsubseteq c_l, c_r$:



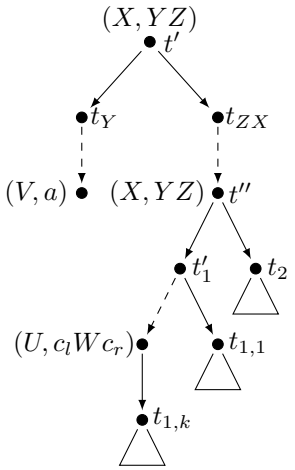
As YZ is a monomial of $\boldsymbol{f}_X$ we can construct from the trees t_1' and t_2 the tree t'':



Now, as $\boldsymbol{f}$ is strongly-connected and in normal form, we find an Y-tree t_Y of height at most $n-1$ which has (V,a) as its single leaf, such that $a \sqsubseteq \mathsf{Y}(t_Y)$; similarly, we find a ZX-tree t_{ZX} of height at most $n-1$ having $\bigcirc$ as its single leaf; the "yield" of t_{ZX} is some monomial $d_l X d_r$ for some $d_l, d_r \in S \setminus \{0\}$. Using these, we construct the following tree t' with $\lambda_v(t') = X$, and $\lambda_m(t') = YZ$. As left child of t', we take the Y-tree t_Y, whereas we take t_{ZX} as the right child, giving us:



We complete this partial derivation tree to a derivation tree by replacing $\bigcirc$ with the tree t'':



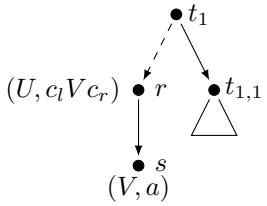
We now have

$$\begin{aligned}\mathsf{Y}(t') = \mathsf{Y}(t_Y) \cdot \mathsf{Y}(t_{ZX}) &= \mathsf{Y}(t_Y) \cdot d_l \cdot \mathsf{Y}(t'') \cdot d_r \\ &\sqsupseteq a \cdot d_l \cdot \mathsf{Y}(t'') \cdot d_r \\ &\sqsupseteq a \cdot \mathsf{Y}(t_1') \cdot \mathsf{Y}(t_2) \\ &\sqsupseteq \mathsf{Y}(t_1) \cdot \mathsf{Y}(t_2) \\ &= \mathsf{Y}(t).\end{aligned}$$

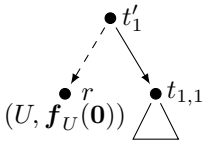
By construction of t', the left child is a Y-tree of height at most $n-1$, while every node from t_{ZX} to t'' has exactly one child. Hence, only the subtree t'' might not have the required form. But as t'' has one node less than t, we find by induction on the number of nodes a tree $\widehat{t''} \in \mathcal{B}_X$ with $\mathsf{Y}(t'') \sqsubseteq \mathsf{Y}(\widehat{t''})$. Replacing in t' the subtree t'' by this tree $\widehat{t''}$,

we then obtain the tree $\hat{t}$ with $\hat{t} \in \mathcal{B}_X$ and $\mathsf{Y}(\hat{t}) \sqsupseteq \mathsf{Y}(t') \sqsupseteq \mathsf{Y}(t)$. This ends the case that $\lambda_m(r) = VW$.

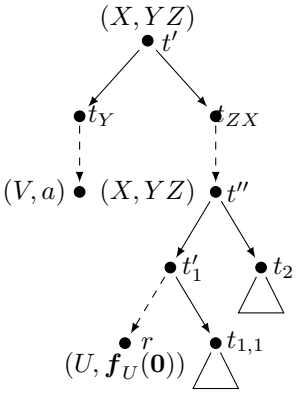
Assume therefore that $\lambda_m(r) = c_l V c_r$ for some $c_l, c_r \in S \setminus \{0\}$, i.e.



We proceed similarly to the previous case, but we define t_1' as follows: again, we remove the leaf s from t_1, but as r has s as its only child, we now relabel r by $\lambda_m(r) := \boldsymbol{f}_U(\mathbf{0})$. As $\boldsymbol{f}$ is clean, we have $\boldsymbol{f}_U(\mathbf{0}) \sqsupseteq 1$. This gives us:



and



Again, we can find a $\widehat{t''} \in \mathcal{B}_X$ with $\mathsf{Y}(t'') \sqsubseteq \mathsf{Y}(\widehat{t''})$ as t'' has one node less than t, and the induction is complete. □

B.4 Proofs of Section 4.5

Theorem 4.5.2.
$\mu\boldsymbol{f} = \boldsymbol{f}^n(\mathbf{0})$ holds for polynomial systems over 1-bounded semirings. ◇

Proof. We reuse the notation from the proof of Theorem 4.3.4: If t_2 is a subtree of a derivation tree t, we write $t = t_1 \cdot t_2$ where t_1 is the partial derivation tree obtained from t by removing t_2.

Recall that, by Proposition 3.2.5, $(\boldsymbol{f}^n(\mathbf{0}))_X = \mathsf{Y}(\mathcal{H}_X^{(n-1)})$, where $\mathcal{H}_X^{(n-1)}$ contains all X-trees of height at most $n-1$. We proceed by derivation tree analysis, i.e., by discharging the precondition of Proposition 4.1.2. So it suffices to show that for any X-tree t there is a tree t' of height at most $n-1$ with $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(t')$. We proceed by induction on the number of nodes in t. For the induction base, t has just one node, so $t \in \mathcal{H}_X^{(0)}$. For the induction step w.l.o.g. let t be an X-tree with a height of at least n. Then there is a decomposition $t = t_1 \cdot t_2 \cdot t_3$ with $\lambda_1(t_2) = \lambda_1(t_3)$. We have $\mathsf{Y}(t) = y_1y_2y_3y_4y_5$ where $\mathsf{Y}(t_1) = y_1y_5$, $\mathsf{Y}(t_2) = y_2y_4$ and $\mathsf{Y}(t_3) = y_3$. Let $t' = t_1 \cdot t_3$. Notice that t' is a valid X-tree as $\lambda_1(t_2) = \lambda_1(t_3)$. We have $\mathsf{Y}(t') = y_1y_3y_5$ and, since $y_1y_2y_3y_4y_5 = \mathsf{Y}(t)$, we have $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(t')$. As t' has fewer nodes than t, there is, by induction hypothesis, an X-tree t'' of height at most $n-1$ such that $\mathsf{Y}(t') \sqsubseteq \mathsf{Y}(t'')$. So we get $\mathsf{Y}(t) \sqsubseteq \mathsf{Y}(t') \sqsubseteq \mathsf{Y}(t'')$. □

Appendix C

Missing Proofs of Chapter 5

C.1 Proofs of Section 5.2

Proposition 5.2.2.
Let $\langle S, \sqcup, \cdot, \bot, 1\rangle$ be a totally ordered cio-semiring. Define $a \sqcap b$ as stated above. Let $a, b, c \in S$ and $(a_i)_{i\in\mathbb{N}}$ an ω-chain. We then have:

(1) $\sqcup$ and $\sqcap$ distribute:

$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c), \text{ and } a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c).$$

(2) $\cdot$ distributes over $\sqcap$: $a \cdot (b \sqcap c) = (a \cdot b) \sqcap (a \cdot c)$.

(3) $\sqcap$ is ω-continuous (w.r.t. $\sqsubseteq$):

$$c \sqcap \left(\bigsqcup_{i\in\mathbb{N}} a_i\right) = \bigsqcup_{i\in\mathbb{N}} (c \sqcap a_i).$$

(4) $\sqcap$ is monotone: $b \sqsubseteq c \Rightarrow a \sqcap b \sqsubseteq a \sqcap c$. ◇

Proof. (1) Choose any $a, b, c \in S$. We may assume that $b \sqsubseteq c$, as $\sqsubseteq$ is total; otherwise swap b and c. Then either $a \sqsubseteq b$ or $b \sqsubseteq a \sqsubseteq c$ or $c \sqsubseteq a$ has to hold.

Assume $a \sqsubseteq b \sqsubseteq c$:

$$\begin{array}{lcccl} a \sqcap (b \sqcup c) & = & a \sqcap c & = & a \\ (a \sqcap b) \sqcup (a \sqcap c) & = & a \sqcup a & = & a \end{array}$$

$$\begin{array}{lcccl} a \sqcup (b \sqcap c) & = & a \sqcup b & = & b \\ (a \sqcup b) \sqcap (a \sqcup c) & = & b \sqcap c & = & b \end{array}$$

Assume $b \sqsubseteq a \sqsubseteq c$.

$$\begin{array}{lcll} a \sqcap (b \sqcup c) & = & a \sqcap c & = & a \\ (a \sqcap b) \sqcup (a \sqcap c) & = & b \sqcup a & = & a \end{array}$$

$$\begin{array}{lcll} a \sqcup (b \sqcap c) & = & a \sqcup b & = & a \\ (a \sqcup b) \sqcap (a \sqcup c) & = & a \sqcap c & = & a \end{array}$$

Assume $b \sqsubseteq c \sqsubseteq a$:

$$\begin{array}{lcll} a \sqcap (b \sqcup c) & = & a \sqcap c & = & c \\ (a \sqcap b) \sqcup (a \sqcap c) & = & b \sqcup c & = & c \end{array}$$

$$\begin{array}{lcll} a \sqcup (b \sqcap c) & = & a \sqcup b & = & a \\ (a \sqcup b) \sqcap (a \sqcup c) & = & a \sqcap a & = & a \end{array}$$

(2) We may assume that $b \sqsubseteq c$. Then $a(b \sqcup c) = ac$ and $a(b \sqcap c) = ab$. By distributivity we also have $a(b \sqcup c) = ab \sqcup ac$. So $ac = ab \sqcup ac \sqsupseteq ab$ and, thus, $ab \sqcap ac = ab$ follows.

(3) Assume c is an upper bound of $(a_i)_{i \in \mathbb{N}}$. We then obviously have

$$c \sqcap \bigsqcup_{i \in \mathbb{N}} a_i = \bigsqcup_{i \in \mathbb{N}} a_i \text{ and } \bigsqcup_{i \in \mathbb{N}} (c \sqcap a_i) = \bigsqcup_{i \in \mathbb{N}}.$$

Therefore, assume there is some $k \in \mathbb{N}$ with $c \sqsubseteq a_k$. We then have $c \sqsubseteq a_k \sqsubseteq \bigsqcup_{i \in \mathbb{N}} a_i$, i.e., $c \sqcap \bigsqcup_{i \in \mathbb{N}} a_i = c$. On the other hand, c is always an upper bound on $(c \sqcap a_i)_{i \in \mathbb{N}}$. Thus, c has to be the maximum of $(c \sqcap a_i)_{i \in \mathbb{N}}$.

(4) As $\sqcap$ is ω-continuous in both arguments, it is also monotone. □

C.2 Proofs of Section 5.3

Proposition 5.3.3.
Every si-semiring $\langle S, \cdot, \sqsubseteq, 1, \bot, \top \rangle$ has the following properties:

(1) $\cdot$ distributes both over $\sqcup$ and $\sqcap$:

$$a \cdot (b \sqcup c) = (a \cdot b) \sqcup (a \cdot c) \text{ and } a \cdot (b \sqcap c) = (a \cdot b) \sqcap (a \cdot c).$$

(2) $\sqcup$ and $\sqcap$ distribute:

$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c) \text{ and } a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c).$$

(3) $(1 \sqsubset a \wedge a \cdot x \sqsubseteq x) \Rightarrow x \in \{\bot, \top\}$

(4) $x \sqsubset a \cdot x \Rightarrow (x \in S \setminus \{\bot, \top\} \wedge 1 \sqsubset a)$. ◇

Proof. (1) W.l.o.g. we may assume that $b \sqsubseteq c$. We then have

$$a \cdot (b \sqcup c) = a \cdot c \text{ and } a \cdot (b \sqcap c) = a \cdot b.$$

By monotonicity of $\cdot$ we also have

$$b \sqsubseteq c \Rightarrow a \cdot b \sqsubseteq a \cdot c,$$

which implies that

$$(a \cdot b) \sqcup (a \cdot c) = a \cdot c \text{ and } (a \cdot b) \sqcap (a \cdot c) = a \cdot b.$$

(2) See the proof of Proposition 5.2.2(2).

(3) Assume $x \in S \setminus \{\bot, \top\}$. As we require that strict inequations are preserved when multiplied by x we have:

$$1 \sqsubset a \Rightarrow x \sqsubset a \cdot x.$$

This yields the contradiction $x \sqsubset a \cdot x \sqsubseteq x$.

(4) If $x = \bot$, then we have $x = \bot = a \cdot x$ which contradicts our assumption that $a \cdot x \sqsupset x$. Similarly, if $x = \top$, then $x = \top \sqsupseteq a \cdot x$. Again, this contradicts our assumption that $x \sqsubset a \cdot x$. So, $x \notin \{\bot, \top\}$.

Finally, assume that $a \sqsubseteq 1$. Then $a \cdot x \sqsubseteq x$ and, hence, the contradiction $x \sqsubset a \cdot x \sqsubseteq x$ arises. □

Bibliography

[ABJ98] P. A. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy FIFO channels. In *CAV'98*, LNCS 1427, pages 305–318. Springer, 1998.

[AEI01] L. Aceto, Z. Ésik, and A. Ingólfsdóttir. A fully equational proof of parikhs theorem. *RAIRO, Theoretical Informatics and Applications*, 36:200–2, 2001.

[Ati67] M.F. Atiyah. K-theory (lectures by M.F. Atiyah. notes taken by D.W. Anderson, fall, 1964), 1967.

[BEKL09] T. Brázdil, J. Esparza, S. Kiefer, and M. Luttenberger. Space-efficient scheduling of stochastically generated tasks. Technical report, Technische Universität München, Institut für Informatik, April 2009.

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model checking. In *Proc. CONCUR'97*, LNCS 1243, pages 135–150, 1997.

[Bos01] S. Bosch. *Algebra*. Springer, 2001.

[BSV02] H. Björklund, S. Sandberg, and S. Vorobyov. Optimization on completely unimodal hypercubes. Technical Report 2002–18, Department of Information Technology, Uppsala University, 2002.

[BSV03] H. Björklund, S. Sandberg, and S. Vorobyov. A discrete subexponential algorithm for parity games. In *STACS'03*, LNCS 2607, pages 663–674. Springer, 2003.

[BSV04] H. Björklund, S. Sandberg, and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. In *MFCS'04*, LNCS 3153, pages 673–685. Springer, 2004.

[CC76] P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In *Second Int. Symp. on Programming*, pages 106–130, 1976.

[CC91] P. Cousot and R. Cousot. Comparison of the galois connection and widening/narrowing approaches to abstract interpretation. In *JTASPEFL 91*, pages 74:107–110, 1991.

[CCFR07] D. Caucal, J. Czyzowicz, W. Fraczak, and W. Rytter. Efficient computation of throughput values of context-free languages. In *CIAA'07*, LNCS 4783, pages 203–213. Springer, 2007.

[Con71] J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.

[Cou91] B. Courcelle. On constructing obstruction sets of words. *EATCS Bulletin*, 44:178–185, 1991.

[EGKS08] J. Esparza, T. Gawlitza, S. Kiefer, and H. Seidl. Approximative methods for monotone systems of min-max-polynomial equations. In Luca Aceto et al., editor, *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP), part I*, volume 5125 of *Lecture Notes in Computer Science*, pages 698–710. Springer, 2008.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS'91*. IEEE Computer Society Press, 1991.

[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. Springer, 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.

[EKL08a] J. Esparza, S. Kiefer, and M. Luttenberger. Convergence thresholds of Newton's method for monotone polynomial equations. In Pascal Weil and Susanne Albers, editors, *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 289–300, Bordeaux, France, 2008. Available at http://arxiv.org/abs/0802.2856.

[EKL08b] J. Esparza, S. Kiefer, and M. Luttenberger. Derivation tree analysis for accelerated fixed-point computation. In Masami Ito and

Masafumi Toyama, editors, *Proceedings of the 12th International Conference on Developments in Language Theory (DLT)*, volume 5257 of *Lecture Notes in Computer Science*, pages 301–313, Kyoto, Japan, 2008. Springer.

[EKL09a] J. Esparza, S. Kiefer, and M. Luttenberger. Computing the least fixed point of positive polynomial systems. Technical report, Technische Universität München, Institut für Informatik, April 2009.

[EKL09b] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian program analysis. Technical report, Technische Universität München, Institut für Informatik, April 2009.

[EKM04] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. In *LICS 2004*. IEEE Computer Society, 2004.

[Ési08] Z. Ésik. Iteration semirings. In *Developments in Language Theory*, pages 1–20, 2008.

[EY05] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. In *STACS*, pages 340–352, 2005.

[EY06] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations, 2006. Draft journal submission, `http://homepages.inf.ed.ac.uk/kousha/bib\_index.html`.

[FH97] H. Fernau and M. Holzer. Conditional context-free languages of finite index. In *New Trends in Formal Languages*, pages 10–26, 1997.

[Fri09] O. Friedmann. A super-polynomial lower bound for the parity game strategy improvement algorithm as we know it. *CoRR*, abs/0901.2731, 2009.

[Gru71] J. Gruska. A few remarks on the index of context-free grammars and languages. *Information and Control*, 19:216–223, 1971.

[GS68] S. Ginsburg and E. Spanier. Derivation-bounded languages. *Journal of Computer and System Sciences*, 2:228–250, 1968.

[GS07] T. Gawlitza and H. Seidl. Precise fixpoint computation through strategy iteration. In *European Symposium on Programming (ESOP)*, LNCS 4421, pages 300–315. Springer, 2007.

[GS08] T. Gawlitza and H. Seidl. Precise interval analysis vs. parity games. In *FM*, pages 342–357, 2008.

[GTW02] E. Grädel, W. Thomas, and Th. Wilke. *Automata, logics, and infinite games*. Springer, 2002.

[Hig52] G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc.*, 2, 1952.

[HK66] A. Hoffman and R. Karp. On nonterminating stochastic games. *Management Science*, 12, 1966.

[HK99] M. W. Hopkins and D. Kozen. Parikh's theorem in commutative Kleene algebra. In *Logic in Computer Science*, pages 394–401, 1999.

[How60] R. A. Howard. Dynamic programming and markov processes. *The M.I.T. Press*, 1960.

[JM82] N. Jones and S. Muchnick. A flexible approach to interprocedural data flow analysis and programs with recursive data structures. In *Proceedings of POPL*, pages 66–74. ACM, 1982.

[Jur98] M. Jurdziński. Deciding the winner in parity games is in up $\cap$ co-up. *Inf. Process. Lett.*, 68(3):119–124, 1998.

[Kar92] G. Karner. On limits in complete semirings. *Semigroup Forum*, 45(1):148–165, 1992.

[Kil73] G. A. Kildall. A unified approach to global program optimization. In *POPL*, pages 194–206. ACM, 1973.

[KLE07] S. Kiefer, M. Luttenberger, and J. Esparza. On the convergence of Newton's method for monotone systems of polynomial equations. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, pages 217–226, San Diego, California, USA, 2007. ACM.

[Knu77] D. E. Knuth. A generalization of dijkstra's algorithm. *Inf. Process. Lett.*, 6(1):1–5, 1977.

[Koz90] D. Kozen. On Kleene algebras and closed semirings. In Rovan, editor, *Proc. Math. Found. Comput. Sci.*, volume 452 of *Lecture*

Notes in Computer Science, pages 26–47, Banska-Bystrica, Slovakia, 1990. Springer-Verlag.

[Koz91] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Logic in Computer Science*, pages 214–225, 1991.

[KS92] J. Knoop and B. Steffen. The interprocedural coincidence theorem. In *International Conference on Compiler Construction*, volume 641 of *LNCS*, pages 125–140. Springer-Verlag, 1992.

[KU77] J. B. Kam and J. D. Ullman. Monotone data flow analysis frameworks. *Acta Inf.*, 7:305–317, 1977.

[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.

[Lin76] T. Lindvall. On the maximum of a branching process. *Scandinavian Journal of Statistics*, 3:209–214, 1976.

[Lut08] M. Luttenberger. Strategy iteration using non-deterministic strategies for solving parity games. Technical report, Technische Universität München, Institut für Informatik, April 2008.

[Mos91] A.W. Mostowski. Games with forbidden positions. Technical Report 78, University of Gdańsk, 1991.

[MS99] Y. Mansour and S. Singh. On the complexity of policy iteration. In *UAI 1999*, 1999.

[Ner77] O. Nerman. On the maximal generation size of a non-critical galton-watson process. *Scandinavian Journal of Statistics*, 4(3):131–135, 1977.

[NNH99] F. Nielson, H.R. Nielson, and C. Hankin. *Principles of Program Analysis*. Springer, 1999.

[OR70] J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.

[Par66] R. J. Parikh. On context-free languages. *J. ACM*, 13(4):570–581, 1966.

[Pur95] A. Puri. *Theory of Hybrid Systems and Discrete Event Systems*. PhD thesis, Electronic Research Laboratory, College of Engineering, University of California, Berkley, 1995.

[RHS95] T. Reps, S. Horwitz, and M. Sagiv. Precise interprocedural dataflow analysis via graph reachability. In *Proceedings of POPL*, pages 49–61. ACM, 1995.

[RSJM05] T. Reps, S. Schwoon, S. Jha, and D. Melski. Weighted pushdown systems and their application to interprocedural dataflow analysis. *Science of Computer Programming*, 58(1–2):206–263, October 2005. Special Issue on the Static Analysis Symposium 2003.

[Sal69] A. Salomaa. On the index of a context-free grammar and language. *Information and Control*, 14:474–477, 1969.

[Sch07] S. Schewe. An optimal strategy improvement algorithm for solving parity games. Technical Report 28, Universität Saarbrücken, 2007.

[SF00] H. Seidl and C. Fecht. Interprocedural analyses: A comparison. *Journal of Logic Programming (JLP*, 43:123–156, 2000.

[SP81] M. Sharir and A. Pnueli. *Program Flow Analysis: Theory and Applications*, chapter 7: Two Approaches to Interprocedural Data Flow Analysis, pages 189–233. Prentice-Hall, 1981.

[SRH96] S. Sagiv, T. W. Reps, and S. Horwitz. Precise interprocedural dataflow analysis with applications to constant propagation. *Theoretical Computer Science*, 167(1&2):131–170, 1996.

[VJ00] J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games (Extended abstract). In *CAV'00*, volume 1855 of *LNCS*, 2000.

[WG75] H. W. Watson and Francis Galton. On the probability of the extinction of families. *Journal of the Anthropological Institute of Great Britain*, 4:138–144, 1875.

[Ynt67] M.K. Yntema. Inclusion relations among families of context-free languages. *Information and Control*, 10:572–597, 1967.

[ZP96] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.