# Lösen polynomieller Gleichungssysteme über Semiringen

Michael Luttenberger

Institut für Informatik (I7)
Technische Universität München
luttenbe@in.tum.de

**Abstract:** Diese Dissertation betracht Gleichungssysteme der Form

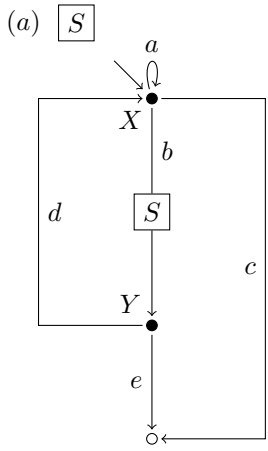
$$X_1 = f_1(X_1, \ldots, X_n), \ldots, X_n = f_n(X_1, \ldots, X_n)$$

(kurz: $X = f(X)$), wobei deren „rechte Seiten" $f_i$ durch multivariate Polynome über sogenannten $\omega$-stetigen Semiringen gegeben sind, und diskutierte Verfahren zur Approximation bzw. Bestimmung deren kleinster Lösung $\mu f$. Diese Gleichungssysteme treten in verschiedensten Bereichen der Informatik auf. Die bekanntesten Beispiele stellen die Datenflussanalyse (prozeduraler) Programme (in diesem Fall ist das Analyseergebnis durch $\mu f$ gegeben) und die Theorie der formalen Sprachen (hier entspricht $X = f(X)$ einer kontextfreien Grammatik mit $\mu f$ z.B. die repräsentierte Sprache oder deren Parikh-Bild) dar.

Zentrales Ergebnis der Dissertation ist eine Verallgemeinerung des Newtonschen Näherungsverfahrens: Es wird gezeigt, dass sich dieses Verfahren allgemein zur Approximation bzw. Bestimmung der kleinsten Lösung eines multivariaten polynomiellen Gleichungssystems über beliebigen $\omega$-stetigen Semiringen verwenden lässt. Das diesen Resultaten zugrundeliegende Beweisverfahren wird weiterhin verwendet, um für spezielle Unterklassen von $\omega$-stetigen Semiringen effizientere Verfahren nicht nur zur Approximation, sondern auch zur Bestimmung von $\mu f$ herzuleiten. Anwendung finden diese Resultate u.a. in der Bestimmung des Parikh-Bilds kontextfreier Sprachen, der Analyse paralleler rekursiver Programme, stochastischer Prozesse, der Analyse der Performanz von Netzwerkalgorithmen.

## 1 Einführung

Gleichungssysteme $X_1 = f_1(X_1, \ldots, X_n), \ldots, X_n = f_n(X_1, \ldots, X_n)$ ergeben sich in natürlicher Weise als Beschreibungen zusammengesetzter Systeme: Die Gleichung $X_i = f_i(X_1, \ldots, X_n)$ beschreibt, wie sich das Verhalten der $i$-ten Komponente aus dem Verhalten aller Systemkomponenten ergibt. Die Lösungen eines solchen Gleichungssystems geben Auskunft über die Eigenschaften des zusammengesetzten Systems. Insbesondere polynomielle Gleichungssysteme sind eine natürliche Darstellungsform des Datenflusses prozeduraler Programme (*abstrakte Datenflussgleichungen*), siehe z.B. die Arbeiten von Cousot und Cousot [CC76] und Sharir und Pnueli [SP81]: Die abstrakten Datenflussgleichungen ordnen jedem Kontrollpunkt des Programms eine Variable $X_i$ und ein multivariates Polynom $f_i$ zu, wobei das Polynom den Datenfluss beginnend (bzw. endend) in

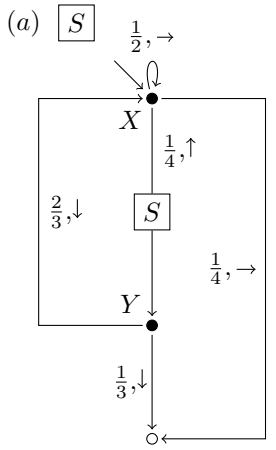



$$(b)\quad \begin{aligned} X &= a \odot X \oplus b \odot X \odot Y \oplus c \\ Y &= d \odot X \oplus e \end{aligned}$$

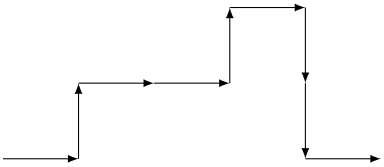
$$(c)\quad \begin{aligned} X &\to aX \mid bXY \mid c \\ Y &\to dX \mid e \end{aligned}$$

Abbildung 1: $(a)$ stellt den Kontrollflussgraphen einer rekursiven Prozedur $S$ dar. Die einzelnen Anweisungen bzw. Kontrollpunkte werden durch $a, b, c, d, e$ bzw. $X, Y$ bezeichnet, wobei der Kontrollpunkt $X$ dem Beginn der Prozedur entspricht. (Zur Vereinfachung wurde das Prozedurende selbst nicht als Kontrollpunkt aufgefasst.) Die durch die Box unterbrochene Kante von $X$ nach $Y$ stellt einen rekursiven Selbstaufruf der Prozedur dar. In $(b)$ ist das abstrakte Gleichungssystem abgebildet, welches die terminierenden Programmabläufe beschreibt; so besagt die Gleichung $X = a \odot X \oplus b \odot X \odot Y \oplus c$, dass ein terminierender Programmablauf, welcher im Kontrollpunkt $Y$ beginnt, entweder (i) zunächst die Operation $a$ ausführt, um dann durch einen in $X$ beginnenden Ablauf fortgesetzt zu werden, oder (ii) sich aus der Konkatenation der Aktion $b$ gefolgt von einem terminierenden (rekursiven) Ablauf der Prozedur $S$ und vervollständigt durch einen terminierenden, in $Y$ beginnenden Ablauf ergibt, oder (iii) sofort nach Ausführung der Anweisung $c$ terminiert. (In Abhängigkeit der betrachteten Analyse können die Gleichungen auch so gewählt werden, dass die Abläufe beschreiben, welche in einem gegeben Programmpunkt terminieren.) $(c)$ zeigt die Interpretation der abstrakten Datenflussgleichungen über dem Sprachsemiring $\langle 2^{\Sigma^*}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$ als kontextfreie Grammatik.

$X_i$ beschreibt (siehe Abbildung 1). Die prinzipielle Idee ist, dass die Koeffizientenbezeichner der Polynome den Effekt der entsprechenden atomaren Programmanweisungen repräsentieren, aus denen sich die Abläufe des Programms zusammensetzen; die Multiplikation dient der Beschreibung des Effekts zweier aufeinanderfolgender partieller Programmabläufe; die Addition fasst den Effekt mehrere Programmabläufe, die an einem gegeben Kontrollpunkt beginnen (oder enden), zusammen. In Abhängigkeit von der zu bestimmenden Information werden die abstrakten Gleichungen über einer geeigneten algebraische Struktur interpretiert. Die gesuchte Information ergibt sich in den meisten Anwendungen aus der kleinsten Lösung $\mu f$ des Gleichungssystems. Siehe Abbildung 2. Im Allgemeinen muss hierbei die Multiplikation weder kommutativ sein, noch muss sie das Distributivitätsgesetz erfüllen [NNH99]. Im Rahmen dieser Disseration werden speziell *$\omega$-stetigen Semiringen* [Kui97] betrachtet, welche stets das Distributivätsgesetz erfüllen, weswegen im Folgenden stets die Distributivität angenommen wird. Dann entspricht die kleinste Lösung $\mu f$ dem Effekt aller terminierenden Programmabläufe.




(b)



$$(c)\quad \begin{aligned} X &= \tfrac{1}{2}X + \tfrac{1}{4}XY + \tfrac{1}{4} \\ Y &= \tfrac{2}{3}X + \tfrac{1}{3} \end{aligned}$$

$$(d)\quad \begin{aligned} X &= \{\rightarrow\} \cdot X \cup \{\uparrow\} \cdot XY \cup \{\rightarrow\} \\ Y &= \{\downarrow\} \cdot X \cup \{\downarrow\} \end{aligned}$$

Abbildung 2: $(a)$ zeigt den Kontrollflussgraphen aus Abbildung 1 instantiiert für ein Steuerprogramms eines Plotters, welches zufällig „Skylines" zeichnet (siehe $(b)$). Beispielhaft betrachte man die Frage, ob das rekursive Programm nur endliche Linienzüge generiert, d.h. ob das Programm stets mit Wahrscheinlichkeit 1 terminiert. Die entsprechende Antwort ergibt sich hierbei aus der Lösung des unter $(c)$ abgebildeten Gleichunggsystems. In diesem Beispiel lässt sich das Gleichungssystem auf eine univariate Gleichung durch Einsetzen reduzieren, was sofort auf die Lösungen $X = 1, Y = 1$ und $X = \frac{3}{2}, Y = \frac{4}{3}$ führt. Die kleinste Lösung liefert daher die gewünschte Antwort, dass das Progamm stets sicher terminiert. Als ein weiteres, einfaches Beispiel zeigt $(d)$ eine mögliche Instanziierung der abstrakten Datenflussgleichungen für die Frage, ob das Programm „inkorrekte Skylines" erzeugt, in welchen ein Punkt mehrmals besucht wird. Zur Beantwortung dieser Frage kann der Sprachsemiring betrachtet werden, welcher von den Steuerbefehlen $\{\rightarrow, \downarrow, \uparrow\}$ zuzüglich einem Fehlerwert $\lightning$ erzeugt wird modulo der Äquivalenzrelation, welche jede inkorrekte Steuersequenz mit $\lightning$ identifiziert und jede korrekte auf den ersten und letzten Befehl der Sequenz reduziert: so ist die unter $(b)$ dargestellte Steuersequenz mit $\rightarrow$ identifiziert. Die kleinste Lösung ist hier $X = \{\rightarrow, \downarrow\uparrow, \uparrow\rightarrow, \downarrow\uparrow\lightning\}$ und $Y = \{\downarrow, \downarrow\rightarrow, \lightning\}$.

Beispiele $\omega$-stetiger Semiringe sind der *reelle Semiring* $\langle [0, \infty], +, \cdot, 0, 1\rangle$ über den nichtnegativen reellen Zahlen $[0, \infty]$ erweitert um ein maximales Element $\infty$ mit der kanonischen Addition und Multiplikation (mit $0 \cdot \infty = 0$) oder der *Sprachsemiring* $\langle 2^{\Sigma^*}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$ erzeugt von einem endlichen Alphabet $\Sigma$ mit der Menge aller Sprachen über $\Sigma$ als Träger und der Vereinigung bzw. Konkatenion von Sprachen als Addition bzw. Multiplikation. Das Standardverfahren zur Approximation bzw. Bestimmung der kleinsten Lösung $\mu f$ ist die Fixpunktiteration, welche ausgehend von einer geeigneten Unterapproximation $x$ diese iterativ $f(x), f^2(x), \ldots$ zu verbessern versucht. Eine kanonische Wahl für den Startwert $x$ ist dabei der Wert, welcher einer leeren Menge von Programmabläufen entspricht, im Fall von $\omega$-stetigen Semiringen ist dies das additiv Neutrale 0. Für diese Wahl entsprechen die Iteranten $f^k(x)$ (unter Annahme der Distributivität) stets dem Gesamteffekt einer *endlichen* Teilmenge der terminierenden Programmabläufe und können daher im Allgemeinen nie den Effekt aller terminierenden Programmabläufe darstellen. In der Praxis werden deshalb häufig algebraische Strukturen verwendet, welche

die Bedingung erfüllen, dass jede monoton anwachsende Sequenz schließlich stationär wird (*ascending chain property*). Ist diese ascending chain property erfüllt, so ist auch garantiert, dass die kleinste Lösung durch die Fixpunktiteration schließlich erreicht wird. Viele interessante algebraische Strukturen erfüllen die ascending chain property jedoch nicht. Als Beispiel betrachte man den Fall, dass die möglichen terminierenden Abläufe (aber nicht deren Häufigkeit) zu bestimmen sind. In diesem Fall wird das Gleichungssystem über Sprachsemiring $\langle 2^{\Sigma^*}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$ interpretiert, der von den Koeffizientenbezeichnern $\Sigma$ des Gleichungssystems erzeugt wird. Mit anderen Worten, das Gleichungssystem wird als kontextfreie Grammatik interpretiert, dessen kleinste Lösung gerade durch die repräsentierte kontextfreie Sprache gegeben ist. Die Iteranten $f^k(\emptyset)$ entsprechen dann endlichen Teilsprachen, womit im Allgemeinen die kleinste Lösung erst im Grenzwert erreicht wird.

Diese Dissertation beschäftigt sich mit der Frage nach besseren Approximation- und Lösungsverfahren für polynomielle Gleichungssysteme über $\omega$-stetigen Semiringen: Es werden Verfahren vorgestellt, die sich dadurch auszeichnen, dass die einzelnen Approximanten nicht auf endliche Teilmengen der terminierenden Programmabläufe beschränkt sind. Zentrale Idee ist die Reduktion des Probelms der Bestimmung der Nullstelle einer nichtlinearen Funktion auf das Lösen einer Sequenz linearer Approximationen, wie sie in der Analysis bereits von Newton in seinem klassischen Näherungsverfahren verwendet wurde. Insbesondere wird gezeigt, dass sich das Newton-Verfahren selbst in natürlicher Weise zu einem Approximationsverfahren für polynomielle Gleichungssysteme über $\omega$-stetigen Semiringen verallgemeinern lässt. Diese Verallgemeinerung des Newton-Verfahrens subsumiert die klassische Fixpunktiteration, d.h. die $i$-te Approximation des verallgemeinerten Newton-Verfahrens beinhaltet alle Information, die die $i$-te Iteration der Fixpunktiteration enthält. Speziell für den Fall kommutativer Multiplikation und idempotenter Addition ($a \oplus a = a$) wird weiterhin gezeigt, dass das verallgemeinerte Newton-Verfahren, im Gegensatz zu der gewöhnlichen Fixpunktiteration, die gesuchte Lösung nicht nur approximiert, sondern auch 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 [GS68, Ynt67] oder dem Satz von Parikh [Par66].

Auf diesen Resultaten aufbauend werden in der Dissertation weitere Klassen von Semiringen beschrieben, welche es erlauben, die kleinste Lösung eines polynomiellen Gleichungssystems mittels maximal einer einzigen Linearisierungen zu bestimmen. Anwendungen dieser Resultate sind z.B. die Bestimmung des Durchsatzes einer kontextfreien Grammatik [CCFR07], welcher zur Analyse der Performanz von Netzwerkalgorithmen genutzt werden, und die Datenflussanalyse über unbeschränkten Semiringen [KSSK09]. Weitere Verwendung finden die diesen Resultaten zu Grunde liegenden Technicken und Ideen in der Lösung von Min-Max-Gleichungssysteme über bestimmten total geordneten kommutativen idempotenten Semiringen mit Hilfe von iterativer Verbesserung nichtdeterministischer Strategien, welche Ergebnisse aus [VJ00, BSV04, Sch07, GS08] vereinheitlicht und verallgemeinert.

Im Folgenden soll das zentrale Ergebnis der Dissertation, die Verallgemeinerung des Newton-Verfahren, detaillierter vorgestellt werden.

## 2 Das Newton-Verfahren über $\omega$-stetigen Semiringen

### 2.1 Notation und Grundlagen

Ein *$\omega$-stetiger Semiring (kurz: $\omega$-Semiring im Folgenden)* $\langle S, +, \cdot, 0, 1\rangle$ setzt sich aus zwei Monoiden $\langle S, +, 0\rangle$, $\langle S, \cdot, 1\rangle$ zusammen, wobei Kommutativität nur für die Addition gefordert wird. Der $\omega$-Semiring wird als kommutativ bezeichnet, falls auch die Multiplikation kommutativ ist; er wird als *idempotent* bezeichnet, falls $1 + 1 = 1$ gilt. Die Null ist absorbierened ($0 \cdot a = a \cdot 0 = 0$); die Multiplikation distributiert sowohl von rechts als auch von links über die Addition. Durch die Relation $a \sqsubseteq b :\Leftrightarrow \exists d \in S : a + d = b$ ist eine partielle (*natürliche*) Ordnung auf $S$ gegeben. Bezüglich dieser Ordnung existiert für jede aufsteigende Sequenz $a_1 \sqsubseteq a_2 \sqsubseteq \ldots$ in $S$ das Supremum $\sup_{i\in\mathbb{N}}^{\sqsubseteq} a_i$. Hierdurch ist die Addition auch für abzählbare Sequenzen $(b_i)_{i\in\mathbb{N}}$ in $S$ durch $\sum_{i\in\mathbb{N}} b_i := \sup_{i\in\mathbb{N}}\{b_1 + \ldots + b_i\}$ erklärt. Wie sich formal zeigen lässt, siehe z.B. [Kui97], verhält sich diese $\omega$-Addition wie von absolut konvergenten Reihen bekannt. Insbesondere ist das Ergebnis von der Summationsreihenfolge unabhängig. Beispiele: Bei dem reellen Semiring $\langle[0, \infty], +, \cdot, 0, 1\rangle$ handelt es sich um einen kommutativen $\omega$-Semiring; bei dem Sprachsemiring $\langle 2^{\Sigma^*}, \cup, \cdot, \emptyset, \{\varepsilon\}\rangle$ um einen idempotenten $\omega$-Semiring. Betrachtet man den Sprachsemiring modulo kommutativer Konkatenation, indem man z.B. jedes Wort $w \in \Sigma^*$ auf seinen *Parikh-Vektor* $\Pi(w)$ abbildet, welcher die Vorkommen eines Zeichens $a \in \Sigma$ in $w$ zählt, so erhält man einen kommutativen und idempotenten $\omega$-Semiring mit $\langle 2^{\mathbb{N}^\Sigma}, \cup, +, \emptyset, \{\underline{0}\}\rangle$ als einer möglichen Darstellung, wobei $\underline{0}$ den Nullvektor bezeichnet und die Addition von Mengen elementweise definiert ist.

Ein Polynom über einem $\omega$-Semiring in den Variablen $\mathcal{X}$ ist eine endliche Summe von endlichen Produkten $a_0 X_1 a_1 \ldots X_k a_k$ aus Semiringelementen $a_j$ und Variablen $X_j \in \mathcal{X}$. Ist die Multiplikation kommutativ, so vereinfacht sich diese Definition zu der gebräuchlichen. Jedes Polynom definiert in kanonischer Weise eine Funktion von $S^{\mathcal{X}}$ nach $S$. Ein polynomielles System $f$ definiert für jede Variable $X \in \mathcal{X}$ ein Polynom $f_X$, und somit eine Funktion über $S^{\mathcal{X}}$. Addition und natürliche Ordnung werden komponentenweise von $S$ auf $S^{\mathcal{X}}$ ausgedehnt.

### 2.2 Approximation der kleinsten Lösung

Ein polynomielles System $f$ bestimmt das Gleichungssystem $\{X = f_X \mid X \in \mathcal{X}\}$, im Weiteren durch $\mathcal{X} = f(\mathcal{X})$ abgekürzt. Es kann gezeigt werden [Kui97], dass ein jedes solches System eine wohldefinierte kleinste Lösung $\mu f$ besitzt, welche durch den Grenzwert der monoton anwachsenden Folge $f(\underline{0}) \sqsubseteq f^1(\underline{0}) \sqsubseteq f^2(\underline{0}) \sqsubseteq \ldots$ gegeben ist.[1] Dieses Resultat erlaubt somit stets die kleinste Lösung zu approximieren. Im Allgemeinen ist die Approximationsgeschwindigkeit jedoch nicht sehr hoch. Dies lässt sich bereits an einem sehr einfachen Beispiel quantifizieren. Die Gleichung $X = f(X) := \frac{1}{2}X^2 + \frac{1}{2}$ hat genau eine (doppelte) Lösung mit $X = 1$ über $\mathbb{R}$ bzw. dem reellen Semiring. Betrachtet

[1] $\underline{0}$ bezeichne hierbei die Funktion/Vektor, welche jeder Variablen $X \in \mathcal{X}$ die Null zuordnet.

Sei $g : \mathbb{R} \to \mathbb{R}$ eine differenzierbare Funktion, wobei mit $g'|_x$ die Ableitung von $g$ ausgewertet an $x \in \mathbb{R}$ bezeichnet sei. Die Linearisierung von $g$ in $x$ ist dann durch

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

gegeben. Ist $x$ nahe genug an einer Nullstelle $z$ von $g$ gelegen, so lässt sich $z$ durch die Nullstelle

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

von $l_{g;x}(X)$ approximieren (unter der Annahme, dass $g'|_x \neq 0$). Durch Iteration des Newton-Operators $\mathcal{N}_g$ ausgehend von einem geeigneten Startwert $x$ ergibt sich die zugehörige Newton-Sequenz.

Abbildung 3: Newton-Verfahren

man die Approximanten $f^k(0)$, so lässt sich zeigen [EY09], dass man $2^{k-3}$ viele Iterationen benötigt, damit der Approximationsfehler $|1 - f^k(0)|$ durch $2^{-k}$ beschränkt ist. Auch über einem Sprachsemiring lässt sich die Konvergenzgeschwindigkeit quantifizieren: So lässt sich für die verwandte Gleichung $X = aX + b$ über dem von $\Sigma = \{a, b\}$ erzeugten Sprachsemiring zeigen, dass der Approximant $f^k(\emptyset)$ gerade aus den Wörtern besteht, welche einen Ableitungsbaum der Höhe maximal $k + 1$ bezüglich der Grammatik $X \to aXX \mid b$ besitzen. Das heißt, dass jeder Approximant stets nur eine endliche Teilsprache darstellt.

Um die Konvergenz zu beschleunigen, wird in der Disseration daher die dem Newton-Verfahren (siehe Abbildung 3) zu Grunde liegende Idee der Linearisierung des polynomiellen Systems auf ihre Anwendbarkeit über den reellen bzw. komplexen Zahlen hinaus untersucht. Bekanntlich konvergiert das Newton-Verfahren im besten Fall quadratisch, d.h. der Approximationsfehler $\varepsilon_k$ im $k$-ten Schritt erfüllt $\varepsilon_{k+1} \leq C \cdot \varepsilon_k^2$ für eine von der Funktion abhängige Konstante $C$. Speziell für den reellen Semiring lässt sich weiterhin zeigen [EKL10a], dass die Konvergenz auch stets (bis auf endlich viele Ausnahmen) mindestens linear ist, d.h., $\varepsilon_{k+1} \leq C\varepsilon_k$ gilt mit $C < 1$. Speziell für die Gleichung $X = f(X) = \frac{1}{2}X^2 + \frac{1}{2}$ mit zugehöriger Funktion $g(X) = f(X) - X$ ergibt für die Approximationsfehler der Newton-Iteranten $\varepsilon_{k+1} = \frac{1}{2}\varepsilon_k$, d.h. nach bereits $k$ Iterationen ist der Fehler durch $2^{-k}$ beschränkt.

## 2.3 Newton-Verfahren

Um das Newton-Verfahren auf allgemeinen Semiringen zu definieren, muss zunächst eine entsprechende Linearisierung definiert werden. Hierbei zeigt sich, dass die übliche algebraische Definition der Ableitung eines Polynoms verwendet werden kann, es muss nur der Fall der nicht kommutativen Multiplikation entsprechend beachtet werden. So ergibt sich der nicht konstante Term der Linearisierung des Polynoms $b \cup XaX$ über dem Sprachsemiring erzeugt von $\{a, b\}$ an der Stelle $X = b$ zu $baX \cup Xab$. Für eine formale Definition

sei auf die Dissertation [Lut10] oder auf [EKL10b] verwiesen. Mit Hilfe der Definition der Linearisierung lässt sich das Newton-Verfahren auf Semiringen wie folgt übertragen:

**Definition 1.** Es sei $\mathcal{X} = f(\mathcal{X})$ ein polynomielles Gleichungssystem. Der $i$-te *Newton-Approximant* $\nu^{(i)}$ ist induktiv definiert durch

$$\nu^{(0)} = \underline{0} \quad \text{und} \quad \nu^{(i+1)} = \nu^{(i)} + \Delta^{(i)},$$

wobei $\Delta^{(i)}$ die kleinste Lösung der Linearisierung

$$f'|_{\nu^{(i)}}(\mathcal{X}) + \delta^{(i)} = \mathcal{X}$$

ist. Für $\delta^{(i)}$ kann hierbei ein beliebiges Element aus $S^{\mathcal{X}}$ verwendet werden, welches die Gleichung $f(\nu^{(i)}) = \nu^{(i)} + \mathcal{X}$ erfüllt.

Eine *Newton-Sequenz* ist jede Sequenz $(\nu^{(i)})_{i\in\mathbb{N}}$ von Newton-Approximanten.

Abbildung 4 zeigt ein Beispiel. Die Existenz der „Newton-Updates" $\Delta^{(i)}$ ist hierbei stets garantiert, da es sich bei der Linearisierung $f'|_{\nu^{(i)}}(\mathcal{X}) + \delta^{(i)}$ selbst um ein lineares Gleichungssystem handelt. Die entsprechende Lösung wird üblicherweise mittels des Kleene-Sterns als $\Delta^{(i)} = f'|^*_{\nu^{(i)}}(\delta^{(i)})$ ausgedrückt, welcher auf jedem $\omega$-Semiring durch $a^* := \sum_{i\in\mathbb{N}} a^i$ definiert ist und sich entsprechend auf lineare Abbildungen über $S^{\mathcal{X}}$ erweitern lässt. Die Existenz der „Differenz" $\delta^{(i)}$ folgt wiederum aus der Monotonie der Newton-Sequenz bezüglich der natürlichen Ordnung. Für idempotente $\omega$-Semiringe kann stets $\delta^{(i)} = f(\nu^{(i)})$ gewählt werden. Folgendes Theorem fasst die wichtigsten Eigentschaften der verallgemeinerten Newton-Sequenz zusammen:

**Theorem 2.** *Die Newton-Sequenz ist eindeutig in $S^{\mathcal{X}}$, d.h., unabhängig von der Wahl der $\delta^{(i)}$. Insbesondere konvergiert die Newton-Sequenz monoton von unten gegen die kleinste Lösung des Gleichunggsystems $\mathcal{X} = f(\mathcal{X})$, wobei stets gilt:*

$$f^i(\underline{0}) \sqsubseteq \nu^{(i)} \sqsubseteq f(\nu^{(i)}) \sqsubseteq \nu^{(i+1)} \sqsubseteq \mu f.$$

*Für idempotente $\omega$-Semiringe gilt insbesondere $\nu^{(i+1)} = f'|^*_{\nu^{(i)}}(\underline{0})$.*

Um als praktisches Approximationsverfahren verwendet werden zu können, muss über dem jeweiligen Semiring eine effiziente Berechnung bzw. Darstellung des Newton-Updates $\Delta^{(i)} = f'|^*_{\nu^{(i)}}(\delta^{(i)})$ möglich sein. Für ein Gleichungssystem über einem Sprachsemiring, d.h. einer kontextfreien Grammatik, lässt sich z.B. der $i+1$-te Newton-Approximant im Allgemeinen nur als lineare Grammatik über dem Alphabet und dem $i$-ten Newton-Iteranten darstellen.[2]

### 2.4 Kommutative idempotente $\omega$-Semiringe

Für kommutative Semiringe $S$ lässt sich die Berechnung des Updates $\Delta^{(i)}$ auf die Berechnung des Kleene-Sterns einer quardatischen Matrix über $S$ zurückführen und damit

[2] Interessant wird diese Darstellung allerdings im Zusammenhang mit stochastischen kontextfreien Grammatiken, da dies eine Approximation durch strukturell einfacher stochastische Grammatiken gestattet [EKL07]. Die Darstellung aus Abbildung 4 muss hierfür allerdings leicht angepasst werden.

Für die abstrakten Datenflussgleichungen aus Abbildung 1

$$\begin{aligned} X &= a \odot X \oplus b \odot X \odot Y \oplus c \\ Y &= d \odot X \oplus e \end{aligned}$$

lautet die Definition eines Newton-Approximanten in Vektorschreibweise

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a \odot X \oplus b \odot \nu_X^{(i)} \odot Y \oplus b \odot X \odot \nu_Y^{(i)} \\ d \odot X \end{pmatrix} \oplus \begin{pmatrix} \delta_X^{(i)} \\ \delta_Y^{(i)} \end{pmatrix}.$$

mit $\nu^{(0)} = (c, e)$. Interpretiert über dem von den Koeffizientenbezeichnern erzeugten Sprachsemiring entsprechen diese Gleichungssysteme folgenden Grammatiken. Das ursprüngliche Gleichungssystem entspricht der Grammatik $G$ mit

$$X \to aX \mid bXY \mid c \text{ und } Y \to dX \mid e,$$

während der $i$.te Newton-Approximat durch folgende Grammatik $G^{(i)}$ gegeben ist:

$$\begin{array}{lcllcl} X^{(1)} & \to & aX^{(1)} \mid c & X^{(i)} & \to & aX^{(i)} \mid bX^{(i-1)}Y^{(i)} \mid bX^{(i)}Y^{(i-1)} \mid c \\ Y^{(1)} & \to & bX^{(1)} \mid e & Y^{(i)} & \to & bX^{(i)} \mid e \end{array}$$

An dieser Darstellung lässt sich folgende Charakterisierung der Newton-Approximanten für kontext-freie Grammatiken ablesen: Ein Wort $w$ lässt sich beginnend bei $X^{(i)}$ in $G^{(i)}$ ableiten (kurz: $X^{(i)} \Rightarrow_{G^{(i)}}^* w$) genau dann, wenn eine $i+1$-*beschränkte* Ableitung $X \Rightarrow_G \sigma_1 \Rightarrow_G \ldots \Rightarrow_G w$ von $w$ bezüglich $G$ existiert, d.h. jede Satzform $\sigma$ enthält höchstens $i+1$ Variablen (Nichtterminale) (siehe z.B. [GS68]). Nach Theorem 3 erzeugen $G^{(2)}$ und $G$ insbesondere dieselbe Sprache modulo Kommutativität, womit sich aus $G^{(i)}$ folgende reguläre Ausdrücke für das Parikh-Bild von $G$ ergeben:

$$\rho_X = (a + b^2a^*c + bb^*e)^*c \text{ und } \rho_Y = b(a + b^2a^*c + bb^*e)^*c + e.$$

Abbildung 4: Verallgemeinertes Newton-Verfahren für kontext-freie Sprachen

auf die Berechnung des Kleene-Sterns über $S$ selbst, siehe z.B. [DK09]. Ist der Semiring zudem idempotent, so folgt mit der vereinfachten Darstellung der Newton-Sequenz, dass sich jeder Newton-Iterant als regulärer Ausdruck über $S$ schreiben lässt. Insbesondere kann folgendes Konvergenzresultat gezeigt werden.

**Theorem 3.** *Sei $\mathcal{X} = f(\mathcal{X})$ ein polynomielles Gleichungssystem über einem kommutativen idempotenten $\omega$-Semiring. Dann gilt $\nu^{(n)} = \mu f$ für $n = |\mathcal{X}|$.*

Als direkter Korollar folgt der Satz von Parikh, dass zu jeder kontextfreien Sprache eine unter kommutativer Konkatenation äquivalente reguläre Sprache existiert [Par66]. Weiterhin lässt sich zeigen, dass die Newton-Sequenz in diesem Fall mit der von Hopkins und Kozen in [HK99] definierten Sequenz korrespondiert, womit die dort angegebene exponentielle obere Schranke für die Berechnung von $\mu f$ entsprechend verbessert wird.

Schließlich lässt sich zeigen, dass für kontextfreie Grammatiken der $k$-te Newton-Approximant mit der Sprache der Wörter übereinstimmt, für welche eine $k+1$-beschränkte Ableitung existiert. Dies wird in Abbildung 4 skizziert. Für eine formale Definition von beschränkten Ableitungen sei auf [FH97, GS68, Ynt67] verwiesen. Dieser Zusammenhang erlaubt eine direkte Konstruktion eines Parikh-äquivalenten endlichen Automaten für jede kontextfreie Grammatik [EGKL10]. Weitere Anwendungen dieser Charakterisierung der Newton-Approximanten finden sich im Bereich der parallelen Prozesse [BEKL10] und der Analyse paralleler rekursiver Programme [GMM10].

## Literatur

[BEKL10] Tomás Brázdil, Javier Esparza, Stefan Kiefer und Michael Luttenberger. Space-Efficient Scheduling of Stochastically Generated Tasks. In *ICALP (2)*, Seiten 539–550, 2010.

[BSV04] H. Björklund, S. Sandberg und S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. In *MFCS'04*, LNCS 3153, Seiten 673–685. Springer, 2004.

[CC76] P. Cousot und R. Cousot. Static Determination of Dynamic Properties of Programs. In *Second Int. Symp. on Programming*, Seiten 106–130, 1976.

[CCFR07] D. Caucal, J. Czyzowicz, W. Fraczak und W. Rytter. Efficient Computation of Throughput Values of Context-Free Languages. In *CIAA'07*, LNCS 4783, Seiten 203–213. Springer, 2007.

[DK09] M. Droste und W. Kuich. *Handbook of Weighted Automata*, Jgg. 1, Kapitel 1: Semirings and formal power series, Seiten 3 – 27. Springer, 2009.

[EGKL10] Javier Esparza, Pierre Ganty, Stefan Kiefer und Michael Luttenberger. Parikh's Theorem: A simple and direct construction. *Information Processing Letters (IPL) (to appear)*, 2010.

[EKL07] J. Esparza, S. Kiefer und M. Luttenberger. An Extension of Newton's Method to $\omega$-Continuous Semirings. In *Proceedings of DLT*, LNCS 4588, Seiten 157–168. Springer, 2007.

[EKL10a] Javier Esparza, Stefan Kiefer und Michael Luttenberger. Computing the Least Fixed Point of Positive Polynomial Systems. *SIAM J. Comput.*, 39(6):2282–2335, 2010.

[EKL10b] Javier Esparza, Stefan Kiefer und Michael Luttenberger. Newtonian program analysis. *J. ACM*, 57(6):33, 2010.

[EY09] Kousha Etessami und Mihalis Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *J. ACM*, 56(1), 2009.

[FH97] H. Fernau und M. Holzer. Conditional Context-Free Languages of Finite Index. In *New Trends in Formal Languages*, Seiten 10–26, 1997.

[GMM10] Pierre Ganty, Rupak Majumdar und Benjamin Monmege. Bounded Underapproximations. In *CAV*, Seiten 600–614, 2010.

[GS68] S. Ginsburg und E. Spanier. Derivation-Bounded Languages. *Journal of Computer and System Sciences*, 2:228–250, 1968.

[GS08] T. Gawlitza und H. Seidl. Precise Interval Analysis vs. Parity Games. In *FM*, Seiten 342–357, 2008.

[HK99] M. W. Hopkins und D. Kozen. Parikh's Theorem in Commutative Kleene Algebra. In *Logic in Computer Science*, Seiten 394–401, 1999.

[KSSK09] Morten Kühnrich, Stefan Schwoon, Jirí Srba und Stefan Kiefer. Interprocedural Dataflow Analysis over Weight Domains with Infinite Descending Chains. In *FOSSACS*, Seiten 440–455, 2009.

[Kui97] W. Kuich. *Handbook of Formal Languages*, Jgg. 1, Kapitel 9: Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata, Seiten 609 – 677. Springer, 1997.

[Lut10] Michael Luttenberger. *Solving Systems of Polynomial Equations: A Generalization of Newton's Method*. Dissertation, Technische Universität München, 2010.

[NNH99] F. Nielson, H.R. Nielson und C. Hankin. *Principles of Program Analysis*. Springer, 1999.

[Par66] R. J. Parikh. On Context-Free Languages. *J. ACM*, 13(4):570–581, 1966.

[Sch07] S. Schewe. An Optimal Strategy Improvement Algorithm for Solving Parity Games. Technical Report 28, Universität Saarbrücken, 2007.

[SP81] M. Sharir und A. Pnueli. *Program Flow Analysis: Theory and Applications*, Kapitel 7: Two Approaches to Interprocedural Data Flow Analysis, Seiten 189–233. Prentice-Hall, 1981.

[VJ00] J. Vöge und M. Jurdziński. A Discrete Strategy Improvement Algorithm for Solving Parity Games (Extended Abstract). In *CAV'00*, Jgg. 1855 of *LNCS*, 2000.

[Ynt67] M.K. Yntema. Inclusion relations among families of context-free languages. *Information and Control*, 10:572–597, 1967.

**Michael Luttenberger** hat an der Universität Stuttgart Informatik (1998–2004) und Mathematik (2004–2006) studiert. Von 2004 bis 2010 promovierte er bei Professor Javier Esparza, zunächst an der Universität Stuttgart, dann an der Technischen Univeristät München. Sein Interessensgebiet umfasst die Theorie der formalen Sprachen, Automaten- und Spieletheorie, insbesondere deren algebraische Grundlagen.