Efficient algorithms for infinite-state recursive stochastic models and Newton's method

Alistair Stewart

Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2014

Abstract

Some well-studied infinite-state stochastic models give rise to systems of nonlinear equations. These systems of equations have solutions that are probabilities, generally probabilities of termination in the model. We are interested in finding efficient, preferably polynomial time, algorithms for calculating probabilities associated with these models. The chief tool we use to solve systems of polynomial equations will be Newton's method as suggested by [EY09]. The main contribution of this thesis is to the analysis of this and related algorithms. We give polynomial-time algorithms for calculating probabilities for broad classes of models for which none were known before.

Stochastic models that give rise to such systems of equations include such classic and heavily-studied models as Multi-type Branching Processes, Stochastic Context-Free Grammars(SCFGs) and Quasi Birth-Death Processes. We also consider models that give rise to infinite-state Markov Decision Processes (MDPs) by giving algorithms for approximating optimal probabilities and finding policies that give probabilities close to the optimal probability, in several classes of infinite-state MDPs. Our algorithms for analysing infinite-state MDPs rely on a non-trivial generalization of Newton's method that works for the max/min polynomial systems that arise as Bellman optimality equations in these models. For SCFGs, which are used in statistical natural language processing, in addition to approximating termination probabilities, we analyse algorithms for approximating the probability that a grammar produces a given string, or produces a string in a given regular language.

In most cases, we show that we can calculate an approximation to the relevant probability in time polynomial in the size of the model and the number of bits of desired precision.

We also consider more general systems of monotone polynomial equations. For such systems we cannot give a polynomial-time algorithm, which pre-existing hardness results render unlikely, but we can still give an algorithm with a complexity upper bound which is exponential only in some parameters that are likely to be bounded for the monotone polynomial equations that arise for many interesting stochastic models.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Alistair Stewart*)

Table of Contents

Chapter 1

Introduction

In this thesis we study various classes of infinite-state recursive probabilistic models, including recursive Markov chains, probabilistic pushdown systems, stochastic context-free grammars, multi-type branching processes, quasi-birth-death processes and probabilistic 1-counter automata.

These are infinite-state stochastic models, but thanks to their recursive structure they can be finitely presented. This recursive structure means that certain probabilities associated with the models, the termination probabilities, are the solution of a system of nonlinear equations. Many of the central computational problems associated with these systems can be rephrased as (or else be reduced to) the problem of computing the non-negative *least fixed point* solution of the associated nonlinear system of equations. We are interested in the worst case complexity of these problems.

In recent years, there has been extensive work on the analysis of such models- see, e.g., [EY09, EY12, EKM06, EWY10, BKK11]. These classes of models arise in a variety of fields and have been studied by various communities. Recursive Markov Chains, and the equivalent model of probabilistic pushdown systems, are natural models for probabilistic programs with recursive procedures [EY09, EKM06]. Quasi-birth-death processes, which are essentially equivalent (in discrete-time) to probabilistic 1-counter automata (p1CA), are used in queueing theory and performance evaluation [Neu81, LR99]. Stochastic Context-Free Grammars(SCFGs) are a central model in natural language processing and are used also in biology [DEKM99], and branching processes are a classical probabilistic model with many applications, including in population genetics ([Har63]).

These are all purely stochastic models which can be considered as infinite state Markov chains. They give rise to sets of polynomial equations. We also consider controlled variants, such as Branching Markov Decision Processes, which could be considered as infinite-state Markov Decision Processes. Such models give rise to non-linear equations which include maxima and minima as well as polynomial terms.

The systems of equations that arise are multi-dimensional fixed-point equations. We have n equations in n variables of the form $x_i = P_i(x),\ i = 1,\dots,n$ where $P_i(x)$ is a monotone function of the variables $x = (x_1,\dots,x_n)$. We denote the entire system of equations by $x = P(x)$. The system is a *monotone polynomial system*(MPS) if each $P_i(x)$ is a multi-variate polynomial with only nonnegative coefficients. It is a *probabilistic polynomial system* (PPS) if in addition the coefficients of each polynomial sum to at most 1. All the purely stochastic models we consider give rise to MPSs. Many of the systems we deal with are PPSs, for which we can get much better results than the more general MPSs.

We are interested in the least non-negative solution of $x = P(x)$ which is the least (non-negative) fixed-point (LFP) of $P(x)$. This LFP solution vector has coordinates which are the *termination probabilities* of the associated model. The computation of these termination probabilities is a central problem for the analysis and model checking of these models.

The LFP solution is in general irrational even when the system is a PPS and all the coefficients of the polynomials (and the numerical input data of the given probabilistic model) are rational. Hence we seek to compute the desired quantities up to a desired precision $\varepsilon > 0$. The goal is to compute them as efficiently as possible, as a function of the encoding size of the input (the given probabilistic model, or the system of equation) and the accuracy ε. Where possible, we seek worst-case complexity bounds which are polynomial.

Whenever we need to approximate the solution to a system of polynomial equations, our algorithms employ Newton's method. Thus the analysis of Newton's method is crucial to most of our results. In the case of systems which include minima or maxima, we consider a generalisation of Newton's method which uses linear programming.

1.1 A simple example

A context-free grammar(CFG) consists of a set of terminals, a set of non-terminals, and a set of rules for turning single non-terminals into strings of terminals and non-terminals. We start with a string containing non-terminals and apply rules to expand these non-terminals into strings. For definiteness, we expand the leftmost nonterminal. We do this repeatedly. If we get a string consisting entirely of terminals, we stop and this process constitutes a leftmost derivation of this string of terminals from the initial string in the grammar.

A stochastic context-free grammar(SCFG) is a CFG, together with, for each non-terminal, a probability distribution over rules starting with that non-terminal. SCFGs are defined formally in chapter 2. Leftmost derivation starting with an initial string is a stochastic process. Here is a simple SCFG G. G has three non-terminals A,B and C; three terminals a,b and c; and six rules:

$$\begin{array}{lcl}
A & \xrightarrow{\frac{3}{5}} & aBC \\
A & \xrightarrow{\frac{2}{5}} & b \\
B & \xrightarrow{\frac{1}{2}} & AA \\
B & \xrightarrow{\frac{1}{2}} & c \\
C & \xrightarrow{\frac{1}{2}} & b \\
C & \xrightarrow{\frac{1}{2}} & cA
\end{array}$$

Leftmost derivation may give us a terminal string of a's, b's, and c's.

While the probability of generating a particular string of terminals is important (and was considered in our paper [ESY12b]), a more fundamental question is what is the probability of generating any string of terminals at all, as opposed to the process carrying on forever with all strings containing non-terminals. We can ask this question about G starting with the string A. With $\frac{2}{5}$ probability, we use the second rule on the first step and terminate immediately with the string of terminals b. With $\frac{2}{3}$ probability, we have the string aBC.

During leftmost derivation from aBC, the probability that we ever expand this C is the same as the probability that leftmost derivation terminates starting at B. The conditional probability that the leftmost derivation starting from aBC terminates, given that we eventually expand this C, is just the probability that leftmost derivation starting from C terminates. We introduce the variables x_A, x_B and x_C to denote the probability that leftmost derivation starting at a given single non-terminal terminates. Then the probability that leftmost derivation terminates starting at aBC is $x_B x_C$.

Now we may return to considering leftmost derivation starting at A. Our previous reasoning gives us that $x_A = \frac{2}{5} + \frac{3}{5} x_B x_C$.

In general, if we have a string containing several non-terminals, we may consider each of these non-terminals to independently generate a substring of the final string of terminals if there is one. To obtain the probability of a conjunction of independent events we have to multiply the individual probabilities. So we just multiply the probabilities of termination starting with each of the individual non-terminals. This gives us a monomial associated with each rule of the grammar, and the coefficient of the monomial is the probability of that rule. For each non-terminal, we have a probability distribution over rules. So the probability that leftmost derivation starting at a non-terminal terminates is given by a probabilistic combination of such monomials.

In this case, doing the same for B and C as for A gives us this system of equations:

$$x_A = \frac{2}{5} + \frac{3}{5}x_B x_C$$

$$x_B = \frac{1}{2}x_A^2 + \frac{1}{2}$$

$$x_C = \frac{1}{2} + \frac{1}{2}x_A$$

This set of equations has more than one solution. For example $x_A = x_B = x_C = 1$ is one such solution, but this turns out not to give the probabilities we want. The correct probabilities are given by the least non-negative solution, which here is approximately $x_A \approx 0.9247$, $x_B \approx 0.9087$, $x_C \approx 0.9623$.

1.2 Results and outline of the thesis

We are trying to approximate the solution to a system of nonlinear equations. The complexity of this can be considered as a function of the encoding size of the system of equations and as a function of the desired error ε. We are interested in the complexity of approximating probabilities in terms of the encoding size of the underlying model. However, the encoding size of the system of equations will be at worst polynomial in the encoding size of the model, and frequently it will be linear, and the probabilities are often just coordinates of the solution.

We first show in Chapter 3 that for a system of probabilistic polynomial equations, we can approximate the least fixed-point solution to within $\varepsilon > 0$ in time polynomial in the the encoding size of the system and $\log(1/\varepsilon)$. This of course means that we can approximate extinction probabilities of multi-type branching processes, and termination probabilities for stochastic context-free grammars, to within error $\varepsilon > 0$, in time polynomial in the encoding size of the model and $\log(1/\varepsilon)$.

Additionally we show that for using exact arithmetic, we can approximate the LFP to within log log $(1/\varepsilon)$ in a polynomial number of arithmetic operations. Using this we show that the decision problem, that is given a rational number $r \in [0,1]$, decide whether $q_i^* > r$ for some variable x_i of a given PPS $x = P(x)$, is decidable in polynomial time in the unit-cost arithmetic model. Combining this with a hardness result in [EY09] we have that this decision problem is many-one reducible to the problem PosSLP, and vice versa. PosSLP (see [ABKPM09]) is a decision problem on arithmetic circuits which captures the power of polynomial time with unit-cost exact rational arithmetic.

The content of Chapter 3 corresponds roughly to the content of our paper [ESY12b], but it excludes the material in that paper on computing the probability that a given SCFG generates a given string, as well as related results about computing an (approximate) Chomsky normal form for a given SCFG. Chapter 3 also includes some norm bounds from our paper [ESY12a].

In Chapter 4 we extend the polynomial time result to probabilistic min/max polynomial equations. This allows us to approximate the optimal extinction probabilities for Branching Markov Decision Processes also to within ε in time polynomial in the encoding size of the model and $\log(1/\varepsilon)$. As this is a controlled system, we give an algorithm for finding an ε-optimal policy for the controller also in polynomial time.

The content of Chapter 4 corresponds roughly to the content of our paper [ESY12a].

In Chapter 5 we consider monotone polynomial systems, a generalisation of probabilistic polynomial systems. In [EY09] hardness results were established for such systems, which mean that the same polynomial time bound would be unlikely. So we give an algorithm for approximating the least fixed point with a complexity bound which is exponential in some parameters of the system but is otherwise polynomial in the encoding size of the system and $\log(1/\varepsilon)$. While this bound is exponential, for at least one model, probabilistic one-counter automata, we have good enough bounds on all the parameters ([EWY08]) to get the desired polynomial time upper bound (in the standard Turing model of computation).

The content of Chapter 5 corresponds roughly to the content of our paper [ESY13b].

In Chapter 6, we consider further problems about stochastic context free grammars. Let us note that, although it is not included in this thesis, in the paper [ESY12b], we also considered the probability that a grammar produces a given string, and by utilising the algorithm described in Chapter 3 for termination probabilities, we obtained a polynomial time algorithm for approximating this. In Chapter 6 we consider the more general problem of computing the probability that an SCFG generates a word in a given regular language, specified by a deterministic finite automaton(DFA). We show that if the SCFG satisfies a mild assumption (that it is non-critical) then we can approximate the probability in polynomial time in the encoding size of the SCFG, the DFA and $\log(1/\varepsilon)$ (to desired precision ε). Otherwise, there is a parameter of the SCFG, the critical depth, using which we can get an exponential upper bound on the complexity.

The content of Chapter 6 corresponds roughly to the content of our paper [ESY13a].

We conclude in Chapter 7, by describing some of the remaining open problems and future directions of research related to this thesis.

1.3 Previous work

Some of the previous work for specific models is discussed in the relevant chapters, in particular for Branching Markov Decision processes in Chapter 4, and some relevant work from the natural language processing literature on problems related to SCFGs is mentioned in Chapter 6.

Literature on many of the models considered goes back decades. Computing the *extinction probabilities* for multi-type branching processes was first studied in the 1940s by Kolmogorov and Sevastyanov [KS47]. Branching processes are a basic stochastic model in probability theory, with applications in diverse areas ranging from population biology to the physics of nuclear chain reactions (see [Har63] for the classic theoretical text on BPs, and [KA02, HJV05, PP08] for some of the more recent applied textbooks on BPs). BPs describe the stochastic evolution of a population of objects of distinct types.

Despite decades of applied work on BPs and SCFGs, as well as theoretical work on their computational problems, no polynomial time algorithm was known for computing extinction probabilities for BPs, nor for termination probabilities for SCFGs, nor even for approximating them within any nontrivial constant: prior to this work it was not even known whether one can distinguish in P-time the case where the probability is close to 0 from the case where it is close to 1.

An algorithm for computing the LFP of MPSs, based on Newton's method, was proposed in [EY09]. Given a MPS, we can first identify in polynomial time the variables that have value 0 in the LFP and remove them from the system, yielding a new so-called *cleaned* system. Then a dependency graph between the variables is constructed, the variables and the MPS are decomposed into strongly connected components (SCCs), and Newton's method is applied bottom-up on the SCCs, starting from the all-0 vector. It was shown in [EY09] that, for any MPS that has a (nonnegative) solution, the decomposed variant of Newton's method converges monotonically to the

LFP. Optimized variants of decomposed Newton's method have by now been implemented in several tools (see, e.g., [WE07, NS09]), and they perform quite well in practice on many instances.

Esparza, Kiefer and Luttenberger studied in detail the rate of convergence of Newton's method on MPSs [EKL10] (with or without decomposition). On the negative side, they showed that there are instances of MPSs $x = P(x)$ (which happen to be PPSs), with n variables, where it takes an exponential number of iterations in the input size to get even within just one bit of precision (i.e. accuracy 1/2). On the positive side, they showed that after some initial number k_P of iterations in a first phase, Newton's method thereafter gains bits of precision at a linear rate, meaning that $k_P + c_P \cdot i$ iterations suffice to gain i bits of precision, where both k_P and c_P depend on the input, $x = P(x)$. For strongly connected MPSs, they showed that the length, k_P, of the initial phase is upper bounded by an exponential function of the input size $|P|$, and that $c_P = 1$. For general MPSs that are not strongly connected, they showed that $c_P = n2^n$ suffices, but they provided no upper bound at all on k_P. Thus, they obtained no upper bounds, as a function of the size of the input, $x = P(x)$, for the number of iterations required to get to within even the first bit of precision for general MPSs. Proving such a general bound was left as an open problem in [EKL10].

Etessami, Wojtczak, and Yannakakis [EWY10] analysed probabilistic 1-counter automata(p1CAs) or equivalently quasi-birth death processes. Using the results of [EKL10], and analysis of the specific MPSs which arise from these stochastic models, they were able to obtain a polynomial time bound for approximating the termination probabilities in the unit-cost arithmetic model of computation. They showed that the decomposed Newton's method algorithm from [EY09] converges in a polynomial number of iterations in the size of the input and the bits of precision, and hence the desired termination probabilities of a given p1CA M can be computed within absolute error $\varepsilon = 2^{-i}$ in a number of arithmetic operations that is polynomial in the size $|M|$ of the input and the number $i = \log(1/\varepsilon)$ of bits of precision. The question whether the termination probabilities of a p1CA can be computed in polynomial time (in the standard model i.e. without access to unit cost arithmetic) was left open in [EWY10].

An equivalent way to formulate the problem of computing the LFP, q^*, of a PPS, $x = P(x)$, is as a mathematical optimization problem: *minimize:* $\sum_{i=1}^{n} x_i$; *subject to:* $\{P(x) - x \leq 0;\ x \geq 0\}$. This program has a unique optimal solution, which is the LFP q^*. If the constraints were convex, the solution could be computed approximately using convex optimization methods. In general, the PPS constraints are *not* convex (e.g.,

$x_2x_3 - x_1 \leq 0$ is not a convex constraint), however for certain restricted subclasses of PPSs they are. This is so for *backbutton processes* which were introduced and studied by Fagin et. al. in [FKK$^+$00] and used there to analyze random walks on the web. Backbutton processes constitute a restricted subclass of SCFGs (see [EY09]). Fagin et. al. applied semidefinite programming to approximate the corresponding termination probabilities for backbutton processes, and used this as a basis for approximating other important quantities associated with them.

For PPSs, [EY09] showed the *qualitative* problem of determining which probabilities are exactly 1 (or 0) can be solved in P-time, by exploiting basic results from the theory of branching processes. They proved however that the *decision* problem of determining whether some coordinate of q^* is $\geq 1/2$ is at least as hard as some longstanding open problems in the complexity of numerical computation, namely, the *square-root sum problem*, and PosSLP, a much more general problem, and hence it is very unlikely that the decision problem can be solved in P-time. For MPSs, they show that in fact this hardness holds for computing *any* nontrivial *approximation* of q^*.

Chapter 2

Background and Definitions

2.1 Systems of Equations

As different models can give rise to similar systems of equations, we will start by defining the systems of equations. Only then we will describe the stochastic models which give rise to them.

2.1.1 Monotone and Probabilistic Polynomial Systems

For an n-vector of variables $x = (x_1, \ldots, x_n)$, and a vector $v \in \mathbb{N}^n$, we use the shorthand notation x^v to denote the monomial $x_1^{v_1} \ldots x_n^{v_n}$. Let $\langle \alpha_r \in \mathbb{N}^n \mid r \in R \rangle$ be a multi-set of n-vectors of natural numbers, indexed by the finite set R.[1] Consider a multi-variate polynomial $P_i(x) = \sum_{r \in R} p_r x^{\alpha_r}$, for some rational-valued coefficients p_r, $r \in R$. We shall call $P_i(x)$ a ***monotone polynomial*** if $p_r \geq 0$ for all $r \in R$. If in addition, we also have $\sum_{r \in R} p_r \leq 1$, then we shall call $P_i(x)$ a ***probabilistic polynomial***.

Definition 2.1. *A* **monotone** *(respectively,* **probabilistic***)* **polynomial system of equations***, $x = P(x)$, which we shall call a* **MPS** *(respectively, a* **PPS***), is a system of n equations, $x_i = P_i(x)$, in n variables $x = (x_1, x_2, \ldots, x_n)$, where for all $i \in \{1, 2, \ldots, n\}$, $P_i(x)$ is a monotone (respectively, probabilistic) polynomial.*

For computational purposes, we assume that any MPS that appears as the input to a computation has rational coefficients. (We will occasionally need to reason also about MPSs with irrational coefficients.) For a MPS $x = P(x)$ with rational coefficients, we

[1] For computational purposes, we assume that the n-vectors α_r are encoded in *sparse representation*, by specifying the non-zero coordinates, and with their positive integer coordinate values encoded in binary.

shall use $|P|$ to denote the sum of the number n of variables and the numbers of bits of all the nonzero coefficients and nonzero exponents of all the polynomials in the MPS (rational coefficients are encoded by giving their numerator and denominator in binary). Note that the encoding length of a MPS in sparse representation is at least $|P|$ (as we need to encode all the coefficients and need at leasr one bit for each variable) and at most $O(|P|\log n)$ (since we only need $\log_2 n$ bits to describe each variable).

For any PPS, $x = P(x)$, $P(x)$ defines a *monotone* operator $P : [0,1]^n \to [0,1]^n$, i.e., if $y \geq x \geq \mathbf{0}$ then $P(y) \geq P(x)$.

The monotone operator $P : [0,1]^n \to [0,1]^n$ has a *least fixed point* (**LFP**), $q^* \in [0,1]^n$. In other words, $q^* = P(q^*)$ and for all vectors $q' \in \mathbb{R}^n_{\geq 0}$, if $q' = P(q')$ then $q^* \leq q'$ (coordinate-wise inequality). It is this LFP vector, q^*, that generally contains the probabilities of interest for related stochastic models that we wish to compute.

An MPS, $x = P(x)$, also defines a monotone operator $P : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ on the non-negative orthant $\mathbb{R}^n_{\geq 0}$. An MPS need not in general have any (finite) solution in $\mathbb{R}^n_{\geq 0}$, but when it does so, it has a *least fixed point* solution $q^* = P(q^*)$ such that $0 \leq q' = P(q')$ implies $q^* \leq q'$.

Indeed, even if an MPS does not have a *finite* LFP solution $q^* \in \mathbb{R}^n_{\geq 0}$, it always does have an LFP solution *over the extended non-negative reals*. Namely, we can define the LFP of any MPS, $x = P(x)$, to be the vector $q^* \in \overline{\mathbb{R}}^n_{\geq 0}$ over $\overline{\mathbb{R}}_{\geq 0} = (\mathbb{R}_{\geq 0} \cup \{+\infty\})$, given by $q^* := \lim_{k\to\infty} P^k(\mathbf{0})$.

2.1.2 max-minPPSs and max/minPPSs

Systems of equations arising from infinite state Markov decision processes or games can contain expressions which are not merely polynomials but include maxima or minima.

Definition 2.2. *A* **maximum-minimum probabilistic polynomial system of equations**, *$x = P(x)$, called a* **max-minPPS** *is a system of n equations in n variables $x = (x_1, x_2, \ldots, x_n)$, where for all $i \in \{1, 2, \ldots, n\}$, either:*

- `Max-polynomial`*: $P_i(x) = \max\{p_{i,j}(x) : j \in \{1,\ldots,m_i\}\}$, Or:*
- `Min-polynomial`*: $P_i(x) = \min\{p_{i,j}(x) : j \in \{1,\ldots,m_i\}\}$*

where each $p_{i,j}(x)$ is a probabilistic polynomial, for every $j \in \{1,\ldots,m_i\}$.

We shall call such a system a **maxPPS** *(respectively, a* **minPPS***) if for every $i \in \{1,\ldots,n\}$, $P_i(x)$ is a* `Max-polynomial` *(respectively, a* `Min-polynomial`*).*

A **max/minPPS** *is either a maxPPS or a minPPS.*

Note that we can view a PPS in n variables as a maxPPS, or as a minPPS, where $m_i = 1$ for every $i \in \{1,\dots,n\}$.

For a max-min-MPS $x = P(x)$, with rational coefficients, we shall again use $|P|$ in the same way as for MPSs. Namely, $|P|$ is the sum of the number, n, of variables, and the numbers of bits of all the nonzero coefficients and nonzero exponents of all the polynomials in the max-min-PPS, and we have an additional bit for each equation, encoding whether it is a max or min equation. Note that again the encoding length of a max-min-PPS in sparse representation is at least $|P|$ and at most $O(|P|\log n)$.

Just as for PPSs, any max-minPPS, $x = P(x)$ defines a *monotone* operator $P : [0,1]^n \to [0,1]^n$, and the operator has a *least fixed point* (LFP), $q^* \in [0,1]^n$. Again, it is often the vector q^* that we wish to compute, in relation to an associated class of infinite-state MDPs or stochastic games.

Definition 2.3. *We define a* **policy** *for a max/minPPS, $x = P(x)$, to be a function $\sigma : \{1,\dots,n\} \to \mathbb{N}$ such that $1 \leq \sigma(i) \leq m_i$.*

Intuitively, for each variable, x_i, a policy selects one of the probabilistic polynomials, $p_{i,\sigma(i)}(x)$, that appear on the RHS of the equation $x_i = P_i(x)$, and which $P_i(x)$ is the maximum/minimum over.

Definition 2.4. *Given a max/minPPS $x = P(x)$ over n variables, and a policy σ for $x = P(x)$, we define the PPS $x = P_\sigma(x)$ by:*

$$(P_\sigma)_i(x) = p_{i,\sigma(i)}$$

for all $i \in \{1,\dots,n\}$.

Given a max/minPPS, $x = P(x)$, and a policy, σ, we use q^*_σ to denote the LFP solution vector for the PPS $x = P_\sigma(x)$.

Definition 2.5. *For a maxPPS, $x = P(x)$, a policy σ^* is called* **optimal** *if for all other policies σ, $q^*_{\sigma^*} \geq q^*_\sigma$. For a minPPS $x = P(x)$ a policy σ^* is called* **optimal** *if for all other policies σ, $q^*_{\sigma^*} \leq q^*_\sigma$. A policy σ is* **ε-optimal** *for $\varepsilon > 0$ if $||q^*_\sigma - q^*||_\infty \leq \varepsilon$.*

A non-trivial fact is that optimal policies always exist, and furthermore that they actually attain the LFP q^* of the max/minPPS:

Theorem 2.6 ([EY05], Theorem 2)**.** *For any max/minPPS, $x = P(x)$, there always exists an optimal policy σ^*, and furthermore $q^* = q^*_{\sigma^*}$.*[2] *In other words, fixing an optimal policy, σ^*, the LFP, $q^*_{\sigma^*}$ of the resulting PPS, $x = P_\sigma(x)$, is the same as the LFP, q^*, of the entire max/minPPS, $x = P(x)$.*

2.1.3 Simple normal form

At various points in this thesis, we will find it convenient to put MPSs and max-minPPSs in a normal form that makes our analyses simpler. As shown below, we can always efficiently convert these equation systems to normal form with only linear blowup.

Definition 2.7. *An MPS in* **simple normal form (SNF)***, $x = P(x)$, is a system of n monotone polynomial equations in n variables $x_1, x_2, \ldots, x_n$ where each $P_i(x)$ for $i = 1, 2, \ldots, n$ is in one of two forms:*

- `Form L`*: $P_i(x) = a_{i,0} + \sum_{j=1}^{n} a_{i,j} x_j$,*
- `Form Q`*: $P_i(x) = x_j x_k$, for some j,k*

For max/minPPSs, we will need three forms:

Definition 2.8. *A maxPPS in* **simple normal form (SNF)***, $x = P(x)$, is a system of n equations in n variables $x_1, x_2, \ldots, x_n$ where each $P_i(x)$ for $i = 1, 2, \ldots, n$ is in one of three forms:*

- `Form L`*: $P_i(x) = a_{i,0} + \sum_{j=1}^{n} a_{i,j} x_j$*
- `Form Q`*: $P_i(x) = x_j x_k$, for some j,k*
- `Form M`*: $P_i(x) = \max\{x_j, x_k\}$, for some j,k*

We define **SNF form** *for minPPSs analogously: only the definition of "*`Form M`*" changes, replacing* max *with* min.

[2] Theorem 2 of [EY05] is in fact a more general result, proved in the context of 1-exit Recursive Simple Stochastic Games, which shows that even under a much more general definition of strategies, both players always have optimal so called "deterministic, stackless, and memoryless" strategies for termination. A direct corollary of that result is that for max-minPPSs, both the max player and the min player have optimal policies that attain the LFP q^*.

Proposition 2.9 (cf. Proposition 7.3 [EY09]). *Every MPS (or max/minPPS), $x = P(x)$, can be transformed in P-time to an* "equivalent" *MPS (max/minPPS) , $y = Q(y)$, in SNF form, such that $|Q| \in O(|P|)$. More precisely, the variables x are a subset of the variables y, the LFP of $x = P(x)$ is the projection of the LFP of $y = Q(y)$ onto the x coordinates, and, for a max/minPPS, an optimal policy (respectively, ε-optimal policy) for $x = P(x)$ can be obtained in P-time from an optimal (respectively, ε-optimal) policy of $y = Q(y)$.*

Proof. We can easily convert, in P-time, any MPS into SNF form, using the following procedure.

- For each equation $x_i = P_i(x) = \sum_{j=1}^{m} p_j x^{\alpha_j}$, where $P_i(x)$ is a polynomial that is not just a constant or a single monomial, replace every monomial x^{α_j} on the right-hand-side that is not a single variable by a new variable x_{i_j} and add the equation $x_{i_j} = x^{\alpha_j}$.

- For each variable x_i that occurs in some polynomial with exponent higher than 1, introduce new variables $x_{i_1}, \ldots, x_{i_k}$ where k is the logarithm of the highest exponent of x_i that occurs in $P(x)$, and add equations $x_{i_1} = x_i^2$, $x_{i_2} = x_{i_1}^2$, ..., $x_{i_k} = x_{i_{k-1}}^2$. For every occurrence of a higher power x_i^l, $l > 1$, of x_i in $P(x)$, if the binary representation of the exponent l is $a_k \ldots a_2 a_1 a_0$, then we replace x_i^l by the product of the variables x_{i_j} such that the corresponding bit a_j is 1, and x_i if $a_0 = 1$. After we perform this replacement for all the higher powers of all the variables, every polynomial of total degree >2 is just a product of variables.

- If a polynomial $P_i(x) = x_{j_1} \cdots x_{j_m}$ in the current system is the product of $m > 2$ variables, then add $m - 2$ new variables $x_{i_1}, \ldots, x_{i_{m-2}}$, set $P_i(x) = x_{j_1} x_{i_1}$, and add the equations $x_{i_1} = x_{j_2} x_{i_2}$, $x_{i_2} = x_{j_3} x_{i_3}$, ..., $x_{i_{m-2}} = x_{j_{m-1}} x_{j_m}$.

Now all equations are of the form L or Q.

To convert max/minPPSs into SNF form, we need to do the following steps before the above procedure:

- For each equation $x_i = P_i(x) = \max\{p_1(x), \ldots, p_m(x)\}$, for each $p_j(x)$ on the right-hand-side that is not a variable, add a new variable x_k, replace $p_j(x)$ with x_k in $P_i(x)$, and add the new equation $x_k = p_j(x)$. Do similarly if $P_i(x) = \min\{p_1(x), \ldots, p_m(x)\}$.

- If $P_i(x) = \max\ \{x_{j_1}, \ldots, x_{j_m}\}$ with $m > 2$, then add $m-2$ new variables $x_{i_1}, \ldots, x_{i_{m-2}}$, set $P_i(x) = \max\ \{x_{j_1}, x_{i_1}\}$, and add the equations $x_{i_1} = \max\ \{x_{j_2}, x_{i_2}\}$, $x_{i_2} = \max\ \{x_{j_3}, x_{i_3}\}$, ..., $x_{i_{m-2}} = \max\ \{x_{j_{m-1}}, x_{j_m}\}$. Do similarly if $P_i(x) = \min\{x_{j_1}, \ldots, x_{j_m}\}$ with $m > 2$.

Now all equations are of the form `L`, `Q` or `M`.

The above procedures allows us to convert any MPS or max/minPPS into one in SNF form by introducing $O(|P|)$ new variables and blowing up the size of P by a constant factor $O(1)$.

Furthermore in the max/minPPS case, there is an obvious (and easy to compute) bijection between policies for the resulting SNF form max/minPPS and the original max/minPPS. And, it is not difficult to show that an optimal (respectively, ε-optimal) policy for $y = Q(y)$ maps to an optimal (respectively ε-optimal) policy for $x = P(x)$. □

For any MPS in SNF form, every polynomial $P_i(x)$ has multivariate degree bounded by at most 2 in the variables x. We will call such MPSs **quadratic**. Many theorems in this thesis will apply only to systems of equations which are quadratic, or are in SNF.

2.2 Stochastic Models

2.2.1 Multi-Type Branching Processes

A (finite) ***multi-type Branching Process*** (**BP**), $G = (V, R)$, consists of a (finite) set $V = \{S_1, \ldots, S_n\}$ of *types*, and a (finite) set $R = \cup_{i=1}^{n} R_i$ of *rules*, which are partitioned into distinct rule sets, R_i, associated with each type S_i. Each rule $r \in R_i$ has the form $S_i \overset{p_r}{\to} \alpha_r$, where $p_r \in (0,1]$, and α_r is a finite multiset (possibly the empty multiset) whose elements are in V. Furthermore, for every type S_i, we have $\sum_{r \in R_i} p_r = 1$. The rule $S_i \overset{p_r}{\to} \alpha_r$ specifies the probability with which an entity (or object) of type S_i generates the multiset α_r of offspring in the next generation. As usual, rule probabilities p_r are assumed to be rational for computational purposes. Multisets α_r over V can be encoded by giving a vector $v(\alpha_r) \in \mathbb{N}^n$, with the i'th coordinate $v(\alpha_r)_i$ representing the number of elements of type S_i in the multiset α_r. We assume instead that the multisets α_r are represented even more succinctly in *sparse representation*, by specifying only the non-zero coordinates of the vector $v(\alpha_r)$, encoded in binary.

A BP, $G = (V, R)$, defines a discrete-time stochastic (Markov) process, whose states are multisets over V, or equivalently elements of $\mathbb{N}^n$. If the state at time t is α^t, then the next state α^{t+1} at time $t+1$ is determined by *independently* choosing, for each object of each type S_i in the multiset α^t, a random rule $r \in R_i$ of the form $S_i \xrightarrow{p_r} \alpha_r$, according to the probability p_r of that rule, yielding the multiset α_r as the "offspring" of that object in one generation. The multiset α^{t+1} is then given by the *multiset union* of all such offspring multisets, randomly and independently chosen for each object in the multiset α^t. A trajectory (*sample path*) of this stochastic process, starting at time 0 in initial multiset α^0, is a sequence $\alpha^0, \alpha^1, \alpha^2, \ldots$ of multisets over V. Note that if ever the process reaches *extinction*, i.e., if ever $\alpha^t = \{\}$ at some time $t \geq 0$, then $\alpha^{t'} = \{\}$ for all times $t' \geq t$.

Very fundamental quantities associated with a BP, which are a key to many analyses of BPs, are its vector of ***extinction probabilities***, $q^* \in [0,1]^n$, where q_i^* is defined as the probability that, starting with initial multiset $\alpha^0 := \{S_i\}$ at time 0, i.e., starting with a single object of type S_i, the stochastic process eventually reaches extinction, i.e., that $\alpha^t = \{\}$ at some time $t > 0$.

Given a BP, $G = (V,R)$, there is a system of polynomial equations in $n = |V|$ variables, $x = P(x)$, that we can associate with G, such that the *least* non-negative solution vector for $x = P(x)$ is the vector of extinction probabilities q^* (see, e.g., [Har63, EY09]). Let us define these equations. For an n-vector of variables $x = (x_1, \ldots, x_n)$, and a vector $v \in \mathbb{N}^n$, we use the shorthand x^v to denote the monomial $x_1^{v_1} \ldots x_n^{v_n}$. Given BP $G = (V,R)$, we define equation $x_i = P_i(x)$ by: $x_i = \sum_{r \in R_i} p_r x^{v(\alpha_r)}$. This yields the PPS, $x = P(x)$. It is not hard to establish that $q^* = P(q^*)$. In fact, q^* is the LFP solution of $x = P(x)$, which we have encountered before when discussing PPSs.

2.2.2 Stochastic Context-Free Grammars

A *weighted context-free grammar* (WCFG), $G = (V, \Sigma, R, p)$, has a finite set V of *non-terminals*, a finite set Σ of *terminals* (alphabet symbols), and a finite list of *rules*, $R \subset V \times (V \cup \Sigma)^*$, where each rule $r \in R$ is a pair (A, γ), which we usually denote by $A \rightarrow \gamma$, where $A \in V$ and $\gamma \in (V \cup \Sigma)^*$. Finally $p : R \rightarrow \mathbb{R}^+$ maps each rule $r \in R$ to a positive *weight*, $p(r) > 0$. We often denote a rule $r = (A \rightarrow \gamma)$ together with its weight by writing $A \xrightarrow{p(r)} \gamma$. We will sometimes also specify a specific non-terminal $S \in V$ as the starting symbol.

Note that we allow $\gamma \in (V \cup \Sigma)^*$ to possibly be the empty string, denoted by ε. A rule of the form $A \rightarrow \varepsilon$ is called an ε-*rule*. For a rule $r = (A \rightarrow \gamma)$, we let $\texttt{left}(r) :=$

A and $\texttt{right}(r) := \gamma$. We let $R_A = \{r \in R \mid \texttt{left}(r) = A\}$. For $A \in V$, let $p(A) = \sum_{r \in R_A} p(r)$. A WCFG, G, is called a *stochastic* or *probabilistic context-free grammar* (SCFG or PCFG; we shall use SCFG), if for $\forall A \in V,\ p(A) \leq 1$. An SCFG is called *proper* if $\forall A \in V,\ p(A) = 1$.

For a WCFG, G, a *leftmost derivation* relation, $\Rightarrow$, is defined as follows. For $\alpha, \beta \in (V \cup \Sigma)^*$, and for a rule $r = (A \to \gamma) \in R$, the ternary relation $\alpha \overset{r}{\Rightarrow} \beta$ holds if and only if $\alpha = wAz$, and $\beta = w\gamma z$, where $w \in \Sigma^*$, and $z \in (V \cup \Sigma)^*$. This relation is extended to sequences of rules as follows: for a nonempty string $\pi = r_1 r_2 \ldots r_k \in R^*$ of rules, and strings $\alpha_0, \alpha_k \in (V \cup \Sigma)^*$, we write $\alpha_0 \overset{\pi}{\Rightarrow} \alpha_k$ if and only if $\alpha_0 \overset{r_1}{\Rightarrow} \alpha_1 \overset{r_2}{\Rightarrow} \alpha_2 \overset{r_2}{\Rightarrow} \ldots \overset{r_k}{\Rightarrow} \alpha_k$, for some $\alpha_1, \alpha_2, \ldots, \alpha_{k-1} \in (V \cup \Sigma)^*$.

We define the *weight* (*probability*) of a derivation as follows: For nonempty string $\pi = r_1 \ldots r_k \in R^*$, and $\alpha, \beta \in (V \cup \Sigma)^*$, we let $p(\alpha \overset{\pi}{\Rightarrow} \beta) = \prod_{i=1}^{k} p(r_k)$ if $\alpha \overset{\pi}{\Rightarrow} \beta$, and $p(\alpha \overset{\pi}{\Rightarrow} \beta) = 0$ otherwise. For a grammar $G = (V, \Sigma, R, p)$, a derivation $\pi \in R^*$ is called a *complete derivation* of the string $w \in \Sigma^*$ starting at nonterminal A, if $A \overset{\pi}{\Rightarrow} w$. For any WCFG, G, string $w \in \Sigma^*$, and non-terminal A, there is a natural one-to-one correspondence between the complete derivations of w starting at A and the *parse trees* of w rooted at A, and this correspondence preserves weights. So, rather than defining parse trees separately, we equate *parse trees* rooted at A with complete derivations starting at A.

If π is a complete derivation starting at A, we let $y(\pi)$, the *yield* of π, be the unique string $w \in \Sigma^*$ generated by π, i.e., such that $A \overset{\pi}{\Rightarrow} w$. Note that we can view leftmost derivation for a SCFG, starting at a nonterminal A, or starting at any string $\alpha \in (V \cup \Sigma)^*$ as specifying a stochastic process (Markov chain) whose states are strings in $(V \cup \Sigma)^*$, with start state α, and whose one-step transition relation is given by $\Rightarrow$, where the probability of a transition $\alpha' \overset{r}{\Rightarrow} \beta'$ is $p(r)$. It is not hard to see that this indeed defines a (countable state, discrete-time, time homogeneous) Markov chain.[3]

For a WCFG, $G = (V, \Sigma, R, p)$, nonterminal $A \in \Sigma$, and terminal string $w \in \Sigma^*$, we let $p_A^{G,w} = \sum_{\{\pi \mid y(\pi) = w\}} p(A \overset{\pi}{\Rightarrow} w)$. For a general WCFG, $p_A^{G,w}$ need not be a finite value (it may be $+\infty$, since the sum may not converge). Note however that if G is an SCFG, then $p_A^{G,w}$ defines the probability that, starting at nonterminal A, G generates the terminal string w, and thus it is clearly finite.

The *termination probability* (*termination weight*) of an SCFG (WCFG), G, starting at nonterminal A, denoted q_A^G, is defined by $q_A^G = \sum_{w \in \Sigma^*} p_A^{G,w}$. For an arbitrary WCFG

[3] Technically, strings $w \in \Sigma^*$ are states of this Markov chain, but have no outgoing transitions defined. We can view $w \in \Sigma^*$ as absorbing states of the MC, with self-loop transitions $w \Rightarrow w$ having probability 1.

q_A^G need not be a finite number. A WCFG G is called *convergent* if q_A^G is finite for all $A \in V$. We will only encounter convergent WCFGs in this thesis, so when we say WCFG we mean convergent WCFG, unless otherwise specified.

For any WCFG, $G = (V, \Sigma, R, p)$, with $n = |V|$, assume the nonterminals in V are indexed as $A_1, \ldots, A_n$. We define the following MPS (respectively, PPS) associated with a WCFG (respectively, SCFG), G, denoted $x = P_G(x)$. Corresponding to each nonterminal $A_i \in V$, there will be one variable x_i and one equation, namely $x_i = P_G(x)_i$, where:

$$P_G(x)_i \equiv \sum_{r=(A \to \alpha) \in R_{A_i}} p(r) x^{\kappa(\alpha)}$$

where $\kappa_j(\alpha)$ is the number of occurrences of A_j in the string α. Just as with Multi-Type-Branching processes, it is the LFP solution of this system that gives the probabilities we are interested in:

Proposition 2.10. *(cf. [EY09] or see [NS08]) For any SCFG (or convergent WCFG), G, with n nonterminals $A_1, \ldots, A_n$, the LFP solution of $x = P_G(x)$ is the n-vector $q^G = (q_{A_1}^G, \ldots, q_{A_n}^G)$ of termination probabilities (termination weights) of G.*

In G is an SCFG, then q_A^G is just the total probability with which the derivation process starting at A eventually generates a finite string and (thus) stops, so SCFGs are clearly convergent.

An SCFG, G, is called *consistent starting at A* if $q_A^G = 1$, and G is called *consistent* if it is consistent starting at every nonterminal. Note that even if an SCFG, G, is proper this does not necessarily imply that G is consistent. Indeed there are simple examples of proper SCFGs which are not consistent.

2.2.3 Branching Markov Decision Processes

A **Branching Markov Decision Process** (BMDP) consists of a finite set $V = \{T_1, \ldots, T_n\}$ of types, a finite set A_i of actions for each type, $i = 1, \ldots, n$, and a finite set $R(T_i, a)$ of probabilistic rules for each type T_i and action $a \in A_i$. Each rule $r \in R(T_i, a)$ has the form $T_i \xrightarrow{p_r} \alpha_r$, where α_r is a finite multi-set whose elements are in V, $p_r \in (0, 1]$ is the probability of the rule, and the sum of the probabilities of all the rules in $R(T_i, a)$ is equal to 1: $\sum_{r \in R(T_i, a)} p_r = 1$.

Intuitively, a BMDP describes the stochastic evolution of entities of given types in the presence of a controller that can influence the evolution. Starting from an initial population (i.e. set of entities of given types) X_0 at time (generation) 0, a sequence of populations $X_1, X_2, \ldots$ is generated, where X_k is obtained from X_{k-1} as follows. First the controller selects for each entity of X_{k-1} an available action for the type of the entity; then a rule is chosen independently and simultaneously for every entity of X_{k-1} probabilistically according to the probabilities of the rules for the type of the entity and the selected action, and the entity is replaced by a new set of entities with the types specified by the right-hand side of the rule. The process is repeated as long as the current population X_k is nonempty, and terminates if and when X_k becomes empty. The objective of the controller is either to minimize the probability of termination (i.e., extinction of the population), in which case the process is a minBMDP, or to maximize the termination probability, in which case it is a maxBMDP. At each stage, k, the controller is allowed in principle to select the actions for the entities of X_k based on the whole past history, may use randomization (a mixed strategy) and may make different choices for entities of the same type. However, it turns out that these flexibilities do not increase the controller's power, and there is always an optimal pure, memoryless strategy that always chooses the same action for all entities of the same type ([EY05]).

For each type T_i of a minBMDP (respectively, maxBMDP), let q_i^* be the minimum (respectively maximum) probability of termination if the initial population consists of a single entity of type T_i. From the given minBMDP (maxBMDP) we can construct a minPPS (respectively maxPPS) $x = P(x)$ whose LFP is precisely the vector q^* of optimal termination (extinction) probabilities (see Theorem 20 in the full version of [EY05]): The min/max polynomial $P_i(x)$ for each type T_i contains one polynomial $p_{i,j}(x)$ for each action $j \in A_i$, with $p_{i,j}(x) = \sum_{r \in R(T_i,j)} p_r x^{\alpha_r}$.

2.3 Newton's method

To find a solution for a differentiable system of equations $F(x) = \mathbf{0}$, in n variables, *Newton's method* uses the following iteration scheme: start with some initial vector $x^{(0)} \in \mathbb{R}^n$, and for $k > 0$ let:
$x^{(k+1)} := x^{(k)} - F'(x^{(k)})^{-1}(F(x^{(k)}))$, where $F'(x)$ is the Jacobian matrix of $F(x)$.

Let $x = P(x)$ be a given MPS in n variables. Let $B(x) := P'(x)$ denote the Jacobian matrix of $P(x)$. In other words, $B(x)$ is an $n \times n$ matrix such that $B(x)_{i,j} = \frac{\partial P_i(x)}{\partial x_j}$. Using Newton iteration, starting at n-vector $x^{(0)} := \mathbf{0}$, yields the following iteration:

$$x^{(k+1)} := x^{(k)} + (I - B(x^{(k)}))^{-1}(P(x^{(k)}) - x^{(k)})) \tag{2.1}$$

For a vector $z \in \mathbb{R}^n$, assuming that matrix $(I - B(z))$ is non-singular, we define a single iteration of Newton's method for $x = P(x)$ on z via the following operator:

$$\mathcal{N}_P(z) := z + (I - B(z))^{-1}(P(z) - z) \tag{2.2}$$

2.3.1 Dependency graph and strongly-connected components

For a MPS, $x = P(x)$ with n variables, its variable *dependency graph* is defined to be the digraph $H = (V, E)$, with vertices $V = \{x_1, \ldots, x_n\}$, such that $(x_i, x_j) \in E$ if and only if x_i appears with a non-zero coefficient in $P_i(x)$. For an MPS, that is if and only if in $P_i(x) \equiv \sum_{r \in R_i} p_r x^{v(\alpha_r)}$ there is a coefficient $p_r > 0$ such that $v(\alpha_r)_j > 0$. Intuitively, $(x_i, x_j) \in E$ means that x_i depends directly on x_j.

An MPS, $x = P(x)$, is called ***strongly connected*** if its dependency graph H is strongly connected i.e. if every variable depends, possibly indirectly, on every other.

We can consider the strongly-connected components(SCCs) of an MPS, which are sets of variables corresponding to the SCCs of its dependency graph.

2.3.2 stuff

It was shown in [EY09] that for any MPS, $x = P(x)$, with LFP $q^* \in \mathbb{R}^N_{\geq 0}$, if we first find and remove the variables that have value 0 in the LFP, q^*, and apply a decomposed variant of Newton's method that decomposes the system according to the strongly connected components (SCCs) of the dependency graph, and process them bottom-up, then the values converge *monotonically* to q^*. In [EKL10], it was pointed out that if $q^* > \mathbf{0}$, i.e., after we remove the variables x_i where $q^*_i = 0$, decomposition into SCCs isn't strictly necessary. Decomposition is nevertheless very useful in practice. While in some cases decomposition also simplifies the analysis, in other cases it can complicate the analysis. In Chapter 5, we use the decomposed Newton's method and analyse it, whereas in other chapters we will not use the decomposed Newton's method, and instead analyse Newton's method applied to the entire system.

Proposition 2.11 (cf. Theorem 6.1 of [EY09] and Theorem 4.1 of [EKL10])**.** *Let $x = P(x)$ be an MPS, with LFP $q^* > \mathbf{0}$. Then starting at $x^{(0)} := \mathbf{0}$, the Newton iterations $x^{(k+1)} := \mathcal{N}_P(x^{(k)})$ are well defined and monotonically converge to q^*, i.e.* $\lim_{k\to\infty} x^{(k)} = q^*$*, and $x^{(k+1)} \geq x^{(k)} \geq \mathbf{0}$ for all $k \geq 0$.*

However, when we consider a version of the Newton iteration with rounding, we may not have monotone convergence like this.

Even if we do not decompose, we may still need to use the algorithm from [EY09] that detects zeros. Formally, for any MPS, $x = P(x)$, we can in P-time find and remove any variables x_i, such that the LFP solution has $q^*_i = 0$.[4]

Proposition 2.12. *(Proposition 7.4 of [EY09]) There is a P-time algorithm that, given any MPS*[4]*, $x = P(x)$, over n variables, determines for each $i \in \{1,\dots,n\}$ whether $q^*_i = 0$.*

Thus, for every MPS, we can detect in P-time all the variables x_j such that $q^*_j = 0$, remove their equation $x_j = P_j(x)$, and set the variable x_j to 0 on the RHS of the remaining equations. We obtain as a result a ***cleaned*** MPS, $x' = Q(x')$, which has an LFP $q^* > \mathbf{0}$.

[4] This proposition holds regardless whether the LFP q^* is *finite* or is over the *extended non-negative reals*, $\overline{\mathbb{R}}_{\geq 0}$. Such an extended LFP exists for any MPS.

Chapter 3

Extinction Probabilities of Multi-type Branching Processes

In this chapter we provide the first polynomial time algorithm for computing, to any desired accuracy, the least fixed point solution, q^*, of probabilistic polynomial systems of equations, and thus also provide the first polynomial time approximation algorithm for extinction probabilities of BPs, and termination probabilities of SCFGs. The algorithm proceeds roughly as follows:

1. We begin with a preprocessing step, in which we determine all variables x_i which have value 0 or 1 in the LFP q^* and remove them from the system.

2. On the remaining system of equations, $x = P(x)$, with an LFP q^* such that $\mathbf{0} < q^* < \mathbf{1}$, we apply Newton's method, starting at initial vector $x^{(0)} := \mathbf{0}$. Our key result is to show that, once variables x_i with $q_i^* \in \{0,1\}$ have been removed, Newton's method only requires polynomially many iterations (in fact, only *linearly* many iterations) as a function of both the encoding size of the equation system and of $\log(1/\varepsilon)$ to converge to within additive error $\varepsilon > 0$ of the vector q^*. To do this, we build on the previous works [EY09, EKL10, EY10], and extend them with new techniques.

3. The result in the previous step applies to the unit-cost arithmetic RAM model of computation, where we assume that each iteration of Newton's method is carried out in *exact* arithmetic. The problem with this, of course, is that in general after only a linear number of iterations, the number of bits required to represent the rational numbers in Newton's method can be exponential in the input's encoding size. We resolve this by showing, via a careful round-off analysis, that if after each iteration of Newton's method the positive rational numbers in question are all *rounded down* to a suitably long but polynomial encoding length (as a function of both the input size and

of the desired error $\varepsilon > 0$), then the resulting "approximate" Newton iterations will still be well-defined and will still converge to q^*, within the desired error $\varepsilon > 0$, in polynomially (in fact *linearly*) many iterations.

In section 3.2, we give a linear bound on the number of iterations of Newton's method with exact arithmetic and no rounding. In section 3.3, we extend this result to include rounding, giving a polynomial time algorithm. In 3.4, we establish certain norm bounds for some crucial matrices that arise during application of Newton's method to PPSs (these important bounds will also be used in subsequent chapters). In section 3.5, we return to Newton's method with exact arithmetic and derive a quadratic convergence result, and we use this to show that we can solve decision problems for the LFP of PPSs using only polynomially many iterations of Newton's method with exact arithmetic.

3.1 Preliminaries

Proposition 2.12, tells us that we can remove variables x_i with $q^*_i = 0$, from both MPSs and PPSs, in polynomial time. In [EY09] Etessami and Yannakakis also gave a P-time algorithm to detect whether $q^*_i = 1$ for PPSs. So we can also remove such variables in polynomial time.

Proposition 3.1 ([EY09]). *There is a P-time algorithm that, given a PPS, $x = P(x)$, over n variables, with LFP $q^* \in \mathbb{R}^n_{\geq 0}$, determines for every $i = 1, \ldots, n$ whether or not $q^*_i = 1$.*

The algorithm of [EY09], used to decided whether $q^*_i = 1$, combines decomposition into strongly connected components with a spectral radius test that was initially developed by Sevastyanov and Kolmogorov [KS47]. This latter lest is carried out in [EY09] using linear programming. Esparza, Gaiser, and Kiefer [EGK13] subsequently gave a more efficient algorithm for this test which runs in strongly polynomial time, and only involves solving linear systems of equations (rather than linear programs).

It turns out that once we have removed the variables x_i with $q^*_i = 0$ or $q^*_i = 1$, Newton's method behaves well and we can obtain a polynomial time bound. Sometimes, in Chapters 4 and 6, we will need to analyse the behaviour of Newton's method on PPSs in which some variables have $q^*_i = 0$ or $q^*_i = 1$, in order to obtain results about systems other than PPSs. This is responsible for many of the complications of the analysis in those chapters. Chapter 6 discusses critical PPSs which contain the pathology that requires us to remove variables x_i with $q_i^* = 1$ (indeed Theorem 6.10 gives us a version of the main polynomial time result of this chapter under weaker assumptions on q^*).

3.2 Polynomial upper bounds for Newton on PPSs

The main goal of this section is to show that for PPSs, $x = P(x)$, with LFP $\mathbf{0} < q^* < \mathbf{1}$, polynomially many iterations of Newton's method, *using exact rational arithmetic*, suffice, as a function of $|P|$ and j, to compute q^* to within additive error $1/2^j$. In fact, we show a much stronger *linear* upper bound with small explicit constants:

Theorem 3.2 (Main Theorem of Section 3.2). *Let $x = P(x)$ be any PPS in SNF form, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$. If we start Newton iteration at $x^{(0)} := \mathbf{0}$, with $x^{(k+1)} := \mathcal{N}_P(x^{(k)})$, then for any integer $j \geq 0$ the following inequality holds:*

$$\|q^* - x^{(j+4|P|)}\|_\infty \leq 2^{-j} .$$

To prove their exponential upper bounds for *strongly connected* PPSs, [EKL10] used the notion of a *cone vector* for the matrix $B(q^*)$, that is a vector $d > 0$ such that $B(q^*)d \leq d$. For a strongly connected MPS, $x = P(x)$, with $q^* > \mathbf{0}$, the matrix $B(q^*) \geq 0$ is irreducible, and thus has a positive eigenvector. They used this eigenvector as their cone vector $d > 0$. However, such an eigenvector yields only weak (exponential) bounds. Instead, we show there is a different cone vector $\mathbf{1} - q^*$ for $B(q^*)$ that works for arbitrary (not necessarily strongly-connected) PPSs (Lemma 3.5). Then we show some results similar that those from [EKL10] that allow us to derive bounds on the convergence of Newton iteration using cone vectors. Because $\mathbf{1} - q^*$ depends on the unknown q^*, we will need to obtain a bound on the minimum separation between q^* and $\mathbf{1}$ in terms of $|P|$ to apply these results.

We need a sequence of Lemmas.

Lemma 3.3. *Let $x = P(x)$ be a quadratic MPS with n variables and let $a, b \in \mathbb{R}^n$. Then:*

$$P(a) - P(b) = B(\frac{a+b}{2})(a-b) = \frac{B(a)+B(b)}{2}(a-b)$$

Proof. Let the function $f : \mathbb{R} \to \mathbb{R}^n$ be given by $f(t) := ta + (1-t)b = b + t(a-b)$. Define $G(t) := P(f(t))$.

From the fundamental theorem of calculus, and using the matrix form of the chain rule from multi-variable calculus (see, e.g., [Apo74] Section 12.10), we have:

$$P(a) - P(b) = G(1) - G(0) = \int_0^1 B(f(t))(a-b)\,dt$$

By linearity, we can just take out $(a-b)$ from the integral as a constant, and we get:

$$P(a) - P(b) = (\int_0^1 B(ta + (1-t)b)\,dt)(a-b)$$

We need to show that

$$\int_0^1 B(ta + (1-t)b)\,dt = B(\frac{a+b}{2}) = \frac{B(a)+B(b)}{2}$$

Since all monomials in $P(x)$ have degree at most 2, each entry of the Jacobian matrix $B(x)$ is a polynomial of degree 1 over variables in x. For any integers i, j, with $0 \le i \le n$, $0 \le j \le n$, there are thus real values α and β with

$$(B(ta + (1-t)b))_{ij} = \alpha + \beta t$$

Then

$$(\int_0^1 B(ta + (1-t)b)\,dt)_{ij} = \int_0^1 (\alpha + \beta t)\,dt = \alpha + \frac{\beta}{2}$$

$$(B(\frac{a+b}{2}))_{ij} = \alpha + \frac{\beta}{2}$$

$$(\frac{B(a)+B(b)}{2})_{ij} = \frac{1}{2}((\alpha+\beta)+\alpha) = \alpha + \frac{\beta}{2}$$

□

Lemma 3.4. *Let $x = P(x)$ be a quadratic MPS. Let $z \in \mathbb{R}^n$ be any vector such that $(I - B(z))$ is non-singular, and thus $\mathcal{N}_P(z)$ is defined. Let $q \in \mathbb{R}^n$ be a point with $P(q) = q$ (such as q^*). Then :*

$$q - \mathcal{N}_P(z) = (I - B(z))^{-1}\frac{B(q)-B(z)}{2}(q-z)$$

Proof. Lemma 3.3, applied to q and z, gives: $q - P(z) = \frac{B(q)+B(z)}{2}(q-z)$. Rearranging, we get:

$$P(z) - z = (I - \frac{B(q)+B(z)}{2})(q-z) \tag{3.1}$$

Replacing $(P(z)-z)$ in equation (2.2) by the right hand side of equation (3.1) and subtracting both sides of (2.2) from q^*, gives:

$$\begin{aligned} q-\mathcal{N}_P(z) &= (q-z)-(I-B(z))^{-1}(I-\frac{B(q)+B(z)}{2})(q-z) \\ &= (I-B(z))^{-1}(I-B(z))(q-z)-(I-B(z))^{-1}(I-\frac{B(q)+B(z)}{2})(q-z) \\ &= (I-B(z))^{-1}((I-B(z))-(I-\frac{B(q)+B(z)}{2}))(q-z) \\ &= (I-B(z))^{-1}(\frac{B(q)-B(z)}{2})(q-z)\ . \end{aligned}$$

□

Now we can show that $\mathbf{1}-q^*$ is a cone vector for $B(q^*)$:

Lemma 3.5. *If $x=P(x)$ is a quadratic PPS in n variables with LFP $\mathbf{0}<q^*<\mathbf{1}$, and where $P(x)$ has Jacobian $B(x)$, then $\forall z\in\mathbb{R}^n$ such that $\mathbf{0}\leq z\leq\frac{1}{2}(\mathbf{1}+q^*)$: $B(z)(\mathbf{1}-q^*)\leq(\mathbf{1}-q^*)$.*
In particular, $B(\frac{1}{2}(\mathbf{1}+q^*))(\mathbf{1}-q^*)\leq(\mathbf{1}-q^*)$, and $B(q^*)(\mathbf{1}-q^*)\leq(\mathbf{1}-q^*)$.

Proof. Lemma 3.3 applied to $\mathbf{1}$ and q^* gives: $P(\mathbf{1})-P(q^*)=P(\mathbf{1})-q^*=B(\frac{1}{2}(\mathbf{1}+q^*))(\mathbf{1}-q^*)$. But note that $P(\mathbf{1})\leq\mathbf{1}$, because for any PPS, since the nonnegative coefficients of each polynomial $P_i(x)$ sum to ≤ 1, $P(x)$ maps $[0,1]^n$ to $[0,1]^n$. Thus $\mathbf{1}-q^*\geq P(\mathbf{1})-q^*=B(\frac{1}{2}(\mathbf{1}+q^*))(\mathbf{1}-q^*)$. Now observe that for $0\leq z\leq\frac{1}{2}(\mathbf{1}+q^*)$, $B(\frac{1}{2}(\mathbf{1}+q^*))\geq B(z)\geq 0$, because the entries of Jacobian $B(x)$ have nonnegative coefficients. Thus since $(\mathbf{1}-q^*)\geq 0$, we have $(\mathbf{1}-q^*)\geq B(z)(\mathbf{1}-q^*)$. □

For a square matrix A, let $\rho(A)$ denote the spectral radius of A (i.e. the largest absolute value of an eigenvalue of A).

Theorem 3.6. *For any quadratic PPS, $x=P(x)$ if we have $0<q^*<1$, then for all $0\leq z\leq q^*$, $\rho(B(z))<1$ and $(I-B(z))^{-1}$ exists and is nonnegative.*

Proof. For any square matrix A, let $\rho(A)$ denote the spectral radius of A. We need the following basic fact:

Lemma 3.7 (see, e.g., [HJ85]). *If A is a square matrix with $\rho(A)<1$ then $(I-A)$ is non-singular, the series $\sum_{k=0}^{\infty}A^k$ converges, and $(I-A)^{-1}=\sum_{k=0}^{\infty}A^k$.*

For all $0\leq z\leq q^*$, $B(z)$ is a nonnegative matrix, and since the entries of the Jacobian matrix $B(x)$ have nonnegative coefficients, $B(x)$ is monotone in x, i.e., if $0\leq z\leq q^*$, then $0\leq B(z)\leq B(q^*)$, and thus by basic facts about non-negative matrices $\rho(B(z))\leq$

$\rho(B(q^*))$. Thus by Lemma 3.7 it suffices to establish that $\rho(B(q^*)) < 1$. We will first prove this for *strongly connected* PPSs:

Lemma 3.8. *For any strongly connected PPS, $x = P(x)$, in SNF form with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, we have $\rho(B(q^*)) < 1$.*

Proof. If the Jacobian $B(x)$ is constant, then $B(q^*) = B(\mathbf{1}) = B$. In this case, B is actually an irreducible substochastic matrix, and since we have removed all variables x_i such that $q_i^* = 0$, it is easy to see that some polynomial $P_i(x)$ must have contained a positive constant term, and therefore, in the (constant) Jacobian matrix B there is some row whose entries sum to < 1. Since B is also irreducible, we then clearly have that $\lim_{m\to\infty} B^m = 0$. But this is equivalent to saying that $\rho(B) < 1$. Thus we can assume that the Jacobian $B(x)$ is non-constant. By Lemma 3.5:

$$B(\frac{1}{2}(\mathbf{1}+q^*))(\mathbf{1}-q^*) \leq (\mathbf{1}-q^*)$$

We have $\mathbf{1}-q^* > 0$, and $B(\frac{1}{2}(\mathbf{1}+q^*)) \geq 0$. Thus, by induction, for any positive integer power k, we have

$$B(\frac{1}{2}(\mathbf{1}+q^*))^k(\mathbf{1}-q^*) \leq (\mathbf{1}-q^*) \tag{3.2}$$

Now, since $B(x)$ is non-constant, and $B(x)$ is monotone in x, and since $q^* < \frac{1}{2}(\mathbf{1}+q^*)$, we have $B(q^*) \leq B(\frac{1}{2}(\mathbf{1}+q^*))$ and furthermore there is some entry (i,j) such that $B(q^*)_{i,j} < B(\frac{1}{2}(\mathbf{1}+q^*))_{i,j}$, it follows that:

$$(B(q^*)(\mathbf{1}-q^*))_i < (B(\frac{1}{2}(\mathbf{1}+q^*))(\mathbf{1}-q^*))_i \leq (\mathbf{1}-q^*)_i$$

Therefore, since $B(q^*)$ is irreducible, it follows that for any coordinate r there exists a power $k \leq n$ such that $(B(q^*)^k(\mathbf{1}-q^*))_r < (\mathbf{1}-q^*)_r$. Therefore, $B(q^*)^n(\mathbf{1}-q^*) < (\mathbf{1}-q^*)$. Thus, there exists some $0 < \beta < 1$, such that $B(q^*)^n(\mathbf{1}-q^*) \leq \beta(\mathbf{1}-q^*)$. Thus, by induction on m, for all $m \geq 1$, we have $B(q^*)^{nm}(\mathbf{1}-q^*) \leq \beta^m(\mathbf{1}-q^*)$. But $\lim_{m\to\infty} \beta^m = 0$, and thus since $(\mathbf{1}-q^*) > 0$, it must be the case that $\lim_{m\to\infty} B(q^*)^{nm} = 0$ (in all coordinates). But this last statement is equivalent to saying that $\rho(B(q^*)) < 1$. □

Now we can proceed to arbitrary PPSs. We want to show that $\rho(B(q^*)) < 1$. Consider an eigenvector $v \in \mathbb{R}^n_{\geq 0}$, $v \neq 0$, of $B(q^*)$, associated with the eigenvalue $\rho(B(q^*))$, with $B(q^*)v = \rho(B(q^*))v$. Such an eigenvector exists by standard fact in Perron-Frobenius theory (see, e.g., Theorem 8.3.1 [HJ85]).

Consider any subset $S \subseteq \{1,\ldots,n\}$ of variable indices, and let $x_S = P_S(x_S,x_{D_S})$ denote the subsystem of $x = P(x)$ associated with the vector x_S of variables in set S, where x_{D_S} denotes the variables not in S. Note that $x_S = P_S(x_S, q^*_{D_S})$ is itself a PPS. We call S *strongly connected* if $x_S = P_S(x_S, q^*_{D_S})$ is a strongly connected PPS.

By Lemma 3.8, for any such strongly connected PPS given by indices S, if we define its Jacobian by $B_S(x)$, then $\rho(B_S(q^*)) < 1$. If S defines a bottom strongly connected component that depends on no other components in the system $x = P(x)$, then we would have that $B_S(q^*)v_S = \rho(B(q^*))v_S$ where v_S is the subvector of v with coordinates in S. Unfortunately v_S might in general be the zero vector. However, if we take S to be a strongly connected component that has $v_S \neq 0$ and such that the SCC S only depends on SCCs S' with $v_{S'} = 0$, then we still have $B_S(q^*)v_S = \rho(B(q^*))v_S$. Thus, by another standard fact from Perron-Frobenius theory (see Theorem 8.3.2 of [HJ85]), $\rho(B_S(q^*)) \geq \rho(B(q^*))$. But since $\rho(B_S(q^*)) < 1$, this implies $\rho(B(q^*)) < 1$. □

Note that this theorem tells us, in particular, that for *every* z (including q^*), such that $\mathbf{0} \leq z \leq q^*$, the Newton iteration $\mathcal{N}_P(z)$ is well-defined. This will be important in Section 3.3. We need the following Lemma from [EKL10]. (To be self-contained, and to clarify our assumptions, we provide a short proof. The version in [EKL10] is valid for non-quadratic MPSs but this complicates the proof.)

Lemma 3.9 (Lemma 5.4 from [EKL10])**.** *Let $x = P(x)$ be a quadratic MPS with LFP, $q^* \geq 0$. Let $B(x)$ denote the Jacobian matrix of $P(x)$. For any positive vector $\boldsymbol{d} \in \mathbb{R}^n_{>0}$ that satisfies $B(q^*)\boldsymbol{d} \leq \boldsymbol{d}$, any positive real value $\lambda > 0$, and any nonnegative vector $z \in \mathbb{R}^n_{\geq 0}$, if $q^* - z \leq \lambda\boldsymbol{d}$, and $(I - B(z))^{-1}$ exists and is nonnegative, then $q^* - \mathcal{N}_P(z) \leq \frac{\lambda}{2}\boldsymbol{d}$.*

Proof. By Lemma 3.4, $q^* - \mathcal{N}_P(z) = (I - B(z))^{-1}\frac{1}{2}(B(q^*) - B(z))(q^* - z)$. Note that matrix $(I - B(z))^{-1}\frac{1}{2}(B(q^*) - B(z))$ is nonnegative: we assumed $(I - B(z))^{-1} \geq 0$ and the positive coefficients in $P(x)$ and in $B(x)$ mean $(B(q^*) - B(z)) \geq 0$. This and the assumption that $q^* - z \leq \lambda d$ yields: $q^* - \mathcal{N}_P(z) \leq (I - B(z))^{-1}\frac{1}{2}(B(q^*) - B(z))\lambda\mathbf{d}$. We can rearrange as follows:

$$\begin{aligned}
q^* - \mathcal{N}_P(z) &\leq (I - B(z))^{-1}\frac{1}{2}(B(q^*) - B(z))\lambda\mathbf{d} \\
&= (I - B(z))^{-1}\frac{1}{2}((I - B(z)) - (I - B(q^*)))\lambda\mathbf{d} \\
&= \frac{\lambda}{2}(I - \ (I - B(z))^{-1}(I - B(q^*))\ )\mathbf{d} \\
&= \frac{\lambda}{2}\mathbf{d} - \ \frac{\lambda}{2}(I - B(z))^{-1}(I - B(q^*))\mathbf{d}
\end{aligned}$$

If we can show that $\frac{\lambda}{2}(I - B(z))^{-1}(I - B(q^*))\mathbf{d} \geq 0$, we are done. By assumption: $(I - B(q^*))\mathbf{d} \geq 0$, and since we assumed $(I - B(z))^{-1} \geq 0$ and $\lambda > 0$, we have: $\frac{\lambda}{2}(I - B(z))^{-1}(I - B(q^*))\mathbf{d} \geq 0$. □

Corollary 3.10. *Let $x = P(x)$ be an MPS, with LFP $q^* > \mathbf{0}$, and let $B(x)$ be the Jacobian matrix for $P(x)$. Suppose there is a vector $d \in \mathbb{R}^n$, $\mathbf{0} < d \leq \mathbf{1}$, such that $B(q^*)d \leq d$. For any positive integer $j > 0$, if we perform Newton's method starting at $x^{(0)} := \mathbf{0}$, then $\|q^* - x^{(j - \lfloor \log_2 d_{\min} \rfloor)}\|_\infty \leq 2^{-j}$ where $d_{\min}$ is the smallest coordinate of d.*

Proof. By induction on k, we show $q^* - x^{(k)} \leq 2^{-k}\frac{1}{d_{\min}}d$. For the base case, $k = 0$, since $d > 0$, $\frac{1}{d_{\min}}d \geq \mathbf{1} \geq q^* = q^* - x^{(0)}$. For $k > 0$, apply Lemma 3.9, setting $z := x^{(k-1)}$, $\lambda := \frac{1}{d_{\min}}2^{-(k-1)}$ and $\mathbf{d} := d$. This yields $q^* - x^{(k)} \leq \frac{\lambda}{2}\mathbf{d} = 2^{-k}\frac{1}{d_{\min}}d$. Since we assume $\|d\|_\infty \leq 1$, we have $\|2^{-(j - \lfloor \log_2 d_{\min} \rfloor)}\frac{1}{d_{\min}}d\|_\infty \leq 2^{-j}$, and thus $\|q^* - x^{(j - \lfloor \log_2 d_{\min} \rfloor)}\|_\infty \leq 2^{-j}$. □

Lemma 3.11. *For a quadratic PPS $x = P(x)$, with LFP q^*, where $\mathbf{0} < q^* < \mathbf{1}$, if we start Newton iteration at $x^{(0)} := \mathbf{0}$, then:*

$$\|q^* - x^{(j + \lceil (\log_2 \frac{(\mathbf{1} - q^*)_{max}}{(\mathbf{1} - q^*)_{min}}) \rceil)}\|_\infty \leq 2^{-j}$$

Proof. For $d := \frac{\mathbf{1} - q^*}{\|\mathbf{1} - q^*\|_\infty}$, $d_{min} = \frac{(\mathbf{1} - q^*)_{\min}}{(\mathbf{1} - q^*)_{\max}}$. By Lemma 3.5, $B(q^*)d \leq d$. Apply Corollary 3.10. □

Lemma 3.12. *For a strongly connected quadratic PPS, $x = P(x)$, with LFP q^*, where $\mathbf{0} < q^* < \mathbf{1}$, for any two coordinates k, l of $\mathbf{1} - q^*$:*

$$\frac{(\mathbf{1} - q^*)_k}{(\mathbf{1} - q^*)_l} \geq 2^{-(2|P|)}$$

Proof. Lemma 3.5 says that $B(\frac{1}{2}(\mathbf{1} + q^*))(\mathbf{1} - q^*) \leq (\mathbf{1} - q^*)$. Since every entry of the vector $\frac{1}{2}(\mathbf{1} + q^*))$ is $\geq 1/2$, every non-zero entry of the matrix $B(\frac{1}{2}(\mathbf{1} + q^*))$ is at least $1/2$ times a coefficient of some monomial in some polynomial $P_i(x)$ of $P(x)$. Moreover, $B(\frac{1}{2}(\mathbf{1} + q^*))$ is irreducible. Calling the entries of $B(\frac{1}{2}(\mathbf{1} + q^*))$, $b_{i,j}$, we have a sequence of *distinct* indices, $i_1, i_2, \ldots, i_m$, with $l = i_1$, $k = i_m$, $m \leq n$, where each $b_{i_j i_{j+1}} > 0$. (Just take the "shortest positive path" from l to k.) For any j:

$$(B(\frac{1}{2}(\mathbf{1} + q^*))(\mathbf{1} - q^*))_{i_{j+1}} \geq b_{i_j i_{j+1}}(\mathbf{1} - q^*)_j$$

Using Lemma 3.5 again, $(\mathbf{1} - q^*)_{i_{j+1}} \geq b_{i_j i_{j+1}}(\mathbf{1} - q^*)_{i_j}$. By simple induction: $(\mathbf{1} - q^*)_k \geq (\prod_{j=1}^{l-1} b_{i_j i_{j+1}})(\mathbf{1} - q^*)_l$. Note that $|P|$ includes the encoding size of each positive coefficient of every polynomial $P_i(x)$. We argued before that each $b_{i_j i_{j+1}} \geq c_i/2$ for some coefficient $c_i > 0$ of some monomial in $P_i(x)$. Therefore, since each such c_i is a distinct coefficient that is accounted for in $|P|$, we must have $\prod_{j=1}^{l-1} b_{i_j i_{j+1}} \geq 2^{-(|P|+n)} \geq 2^{-(2|P|)}$, and thus we have: $(\mathbf{1} - q^*)_k \geq 2^{-(2|P|)}(\mathbf{1} - q^*)_l$. □

Combining Lemma 3.11 with Lemma 3.12 establishes the following:

Theorem 3.13. *For a strongly connected PPS, $x = P(x)$ in n variables, in SNF form, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, if we start Newton iteration at $x^{(0)} := \mathbf{0}$, then: $\|q^* - x^{(j+2|P|)}\|_\infty \leq 2^{-j}$.*

To get a polynomial upper bound on the number of iterations of Newton's method for general PPSs, we can apply Lemma 3.11 combined with a Lemma in [EY10] (Lemma 7.2 of [EY10]), which implies that for a PPS $x = P(x)$ with n variables, in SNF form, with LFP q^*, where $q^* < \mathbf{1}$, $(\mathbf{1} - q^*)_{\min} \geq 1/2n2^{|P|^c}$ for some constant c. Instead, we prove the following much stronger result:

Theorem 3.14. *For a PPS, $x = P(x)$ in n variables, in SNF form, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, for all $i = 1, \ldots, n$: $\mathbf{1} - q_i^* \geq 2^{-4|P|}$. In other words, $\|q^*\|_\infty \leq 1 - 2^{-4|P|}$.*

Recall again that we assume that the PPS, $x = P(x)$, is in SNF form, where each equation $x_i = P_i(x)$ is either of the form $x_i = x_j x_k$, or is of the form $x_i = \sum_j p_{i,j} x_j + p_{i,0}$. There is one equation for each variable. If n is the number of variables, we can assume w.l.o.g. that $|P| \geq 3n$ (i.e. the input has at least 3 bits per variable).

We know that the ratio of largest and smallest non-zero components of $\mathbf{1} - q^*$ is smaller than $2^{2|P|}$ in the strongly connected case (Lemma 3.12). In the general case, two variables may not depend on each other, even indirectly. Nevertheless, we can establish a good upper bound on coordinates of $q^* < \mathbf{1}$. As before, we start with the strongly connected case:

Theorem 3.15. *Given a strongly connected PPS, $x = P(x)$, in SNF form, with $P(\mathbf{1}) = \mathbf{1}$, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, and with rational coefficients, then*

$$q_i^* < 1 - 2^{-3|P|}$$

for some $1 \leq i \leq n$.

Proof. Consider the vector $(I - B(\mathbf{1}))(\mathbf{1} - q^*)$. As $P(\mathbf{1}) = \mathbf{1}$, by Lemma 3.3 we have $B(\frac{1}{2}(\mathbf{1} + q^*))(\mathbf{1} - q^*) = \mathbf{1} - q^*$ and so

$$(B(\mathbf{1}) - I)(\mathbf{1} - q^*) = (B(\mathbf{1}) - B(\frac{1}{2}(\mathbf{1} + q^*)))(1 - q^*)$$

This is zero except for coordinates of form `Q` as rows of $B(\frac{1}{2}(\mathbf{1} + q^*))$ and $B(\mathbf{1})$ that correspond to form `L` equations are identical. If we have an expression of form `Q`,

$(P(x))_i = x_j x_k$, then

$$\begin{aligned}(B(\mathbf{1})-I)(\mathbf{1}-q^*)_i &= (B(\mathbf{1})-B(\frac{1}{2}(\mathbf{1}+q^*)))(1-q^*))_i \\ &= (1/2)(1-q_k^*)(\mathbf{1}-q_j^*)+(1/2)(\mathbf{1}-q_j^*)(\mathbf{1}-q_k^*) \\ &= (\mathbf{1}-q_k^*)(\mathbf{1}-q_j^*)\end{aligned}$$

Consequently:

$$||(I-B(\mathbf{1}))(\mathbf{1}-q^*)||_\infty \leq ||(\mathbf{1}-q^*)||_\infty^2 \tag{3.3}$$

Now suppose that $(I-B(\mathbf{1}))$ is non-singular. In that case, we have that:

$$\mathbf{1}-q^* = (I-B(\mathbf{1}))^{-1}(I-B(\mathbf{1}))(\mathbf{1}-q^*)$$

$$||\mathbf{1}-q^*||_\infty \leq ||(I-B(\mathbf{1}))^{-1}||_\infty ||(I-B(\mathbf{1}))(\mathbf{1}-q^*)||_\infty$$

$$||\mathbf{1}-q^*||_\infty \leq ||(I-B(\mathbf{1}))^{-1}||_\infty ||(\mathbf{1}-q^*)||_\infty^2$$

$$||\mathbf{1}-q^*||_\infty \geq \frac{1}{||(I-B(\mathbf{1}))^{-1}||_\infty} \tag{3.4}$$

where $||\cdot||_\infty$ on matrices is the induced norm of $||\cdot||_\infty$ on vectors. $||A||_\infty$ for an $n \times m$ matrix A with entries a_{ij} is the maximum absolute value row sum $\max_{i=1}^n \sum_{j=1}^n |a_{ij}|$.

So an upper bound on $||(I-B(\mathbf{1}))^{-1}||_\infty$ will give the lower bound on $||\mathbf{1}-q^*||_\infty$ we are looking for.

Lemma 3.16. *Let A be a non-singular $n \times n$ matrix with rational entries. If the product of the denominators of all these entries is m, then*

$$||A^{-1}||_\infty \leq nm||A||_\infty^n$$

Proof. The i,jth entry of A^{-1} satisfies:

$$(A^{-1})_{ij} = \frac{\det(M_{ij})}{\det(A)}$$

where M_{ij} is the i,jth minor of A, made by deleting row i and column j. $||M_{ij}||_\infty \leq ||A||_\infty$ as we've removed entries from rows. We always have $|\det(M_{ij})| \leq ||M_{ij}||_\infty^n$ (see, e.g., [HJ85] page 351), so:

$$|(A^{-1})_{ij}| \leq \frac{||A||_\infty^n}{|\det(A)|} \tag{3.5}$$

Meanwhile $\det(A)$ is a non-zero rational number (because by assumption A is non-singular). If we consider the expansion for the determinant $\det(A) = \sum_\sigma \mathrm{sgn}\sigma \prod_{i=1}^n a_{i\sigma(i)}$, then the denominator of each term $\prod_{i=1}^{n} a_{i\sigma(i)}$ is a product of denominators of distinct entries $a_{i\sigma(i)}$ and therefore divides m. Since every term can thus be rewritten with denominator m, the sum can also be written with denominator m, and therefore $|\det(A)| \geq \frac{1}{m}$. Thus, plugging into inequality (3.5), we have:

$$|(A^{-1})_{ij}| \leq m\|A\|_{\infty}^{n}$$

Taking the maximum row sum $\|A^{-1}\|_{\infty}$,

$$\|A^{-1}\|_{\infty} \leq nm\|A\|_{\infty}^{n}$$

□

If we take $(I - B(\mathbf{1}))$ to be the matrix A of Lemma 3.16, then noting that the product of all the denominators in $(I - B(\mathbf{1}))$ is at most $2^{|P|}$, this gives:

$$\|(I - B(\mathbf{1}))^{-1}\|_{\infty} \leq n2^{|P|}\|(I - B(\mathbf{1}))\|_{\infty}^{n}$$

Of course $\|(I - B(\mathbf{1}))\|_{\infty} \leq 1 + \|B(\mathbf{1})\|_{\infty} \leq 3$ (note that here we are using the fact that the system is in SNF normal form). Thus

$$\|(I - B(\mathbf{1}))^{-1}\|_{\infty} \leq 3^{n}n2^{|P|}$$

Using inequality (3.4), and since as discussed, w.l.o.g., $|P| \geq 3n \geq n\log 3 + \log n$, this gives:

$$\|\mathbf{1} - q^{*}\|_{\infty} \quad \geq \quad \frac{1}{n}2^{-|P|}3^{-n} > 2^{-2|P|}$$

Now consider the other case where $(I - B(\mathbf{1}))$ is singular. We can look for a small solution v to:

$$(I - B(\mathbf{1}))v = (I - B(\mathbf{1}))(\mathbf{1} - q^{*}) \tag{3.6}$$

Lemma 3.17. *Suppose we have an equation $Ax = b$, with A a singular $n \times n$ matrix, b a non-zero vector, and we know that $Ax = b$ has a solution. Then it must have a solution $AA'^{-1}b = b$ where A' is a non-singular matrix generated from A by replacing some rows with rows that have a single* 1 *entry and the rest* 0.

Proof. If A has rank $r < n$, then there are linearly independent vectors $a_1, a_2, \ldots, a_r$ such that $a_1^T, a_2^T, \ldots, a_r^T$ are rows of A and other rows of A are linear combinations of these. Let $e_1, e_2, \ldots, e_n$ be the canonical basis of $\mathbb{R}^n$, i.e. each e_i has ith coordinate 1 and the rest 0. By the well known fact that the set of linearly independent subsets of a vector space form a matroid, and in particular satisfy the exchange property of a matroid (see any good linear algebra or combinatorics text, e.g,. [Cam94], Proposition 12.8.2) , we know there is a basis for $\mathbb{R}^n$ of the form $\{a_1, a_2, \ldots, a_r, e_{i_{r+1}}, e_{i_{r+2}}, \ldots, e_{i_n}\}$ for some choice of $i_{r+1}, i_{r+2}, \ldots, i_n$. We form a matrix A' with elements of this basis as rows by starting with A and keeping r rows corresponding to $a_1^T, a_2^T, \ldots a_r^T$, and replacing the others in some order with $e_{i_{r+1}}^T, e_{i_{r+2}}^T, \ldots, e_{i_n}^T$. Specifically, there is a permutation σ of $\{1, \ldots, n\}$ such that if $1 \leq k \leq r$, the $\sigma(k)$'th row of A' and A are a_k^T and if $r < k \leq n$, the $\sigma(k)$'th row of A' is $e_{i_k}^T$.

A' is non-singular since its rows form a basis of $\mathbb{R}^n$. It remains to show that $AA'^{-1}b = b$. Since $Ax = b$ has a solution and the set R of rows $a_1^T, \ldots, a_r^T$ spans the row space of A, every equation corresponding to a row of $Ax = b$ is a linear combination of the r equations corresponding to the rows in R. Therefore, if x any vector that satisfies the r equations corresponding to the rows in R then it satisfies all the equations of $Ax = b$. The vector $A'^{-1}b$ satisfies these r equations by the definition of A'. Therefore, $AA'^{-1}b = b$. □

We can replace some rows of $(I - B(\mathbf{1}))$ to get an A' using this Lemma and then use Lemma 3.16 on

$$v' = A'^{-1}(I - B(\mathbf{1}))(\mathbf{1} - q^*)$$

We still have $||A'||_\infty \leq 3$ and the product of all the denominators of non-zero entries is smaller than $2^{|P|}$. As for $||(I - B(1))^{-1}||_\infty$ before:

$$||A'^{-1}||_\infty \leq 3^n n 2^{|P|}$$

Now, using inequality (3.3), we have

$$||v'||_\infty \leq 3^n n 2^{|P|} ||(\mathbf{1} - q^*)||_\infty^2 \tag{3.7}$$

Now by equation (3.6), we have that $(I - B(1))((\mathbf{1} - q^*) - v') = 0$. Thus $(\mathbf{1} - q^*) - v'$ is an eigenvector of $B(\mathbf{1})$ with eigenvalue 1. But we know that $B(\mathbf{1})$ is nonnegative, irreducible, and has spectral radius bigger than 1 (because $q^* < \mathbf{1}$ by assumption, see e.g., [EY09] proof of Theorem 8.1). Thus Perron-Frobenius theory (e.g., see Corollary 8.1.29 in [HJ85]) gives us that $(\mathbf{1} - q^*) - v'_i$ is not a positive vector (because the only positive eigenvectors are associated with the top eigenvalue). Thus some coordinate i has:

$$v'_i \geq 1 - q^*_i$$

Thus, by inequality (3.7), we have:

$$\mathbf{1} - q_i^* \leq 3^n n 2^{|P|} \|(\mathbf{1} - q^*)\|_\infty^2$$

but the proof of Lemma 3.12 gave that:

$$(\mathbf{1} - q_i^*)2^{|P|+n} \geq \|(\mathbf{1} - q^*)\|_\infty$$

Combining these inequalities, we have

$$\begin{aligned} \mathbf{1} - q_i^* &\leq 3^n n 2^{|P|} \|(\mathbf{1} - q^*)\|_\infty^2 \\ &\leq 3^n n 2^{|P|} (\mathbf{1} - q_i^*) 2^{|P|+n} \|(\mathbf{1} - q^*)\|_\infty \end{aligned}$$

Dividing both sides by $(\mathbf{1} - q_i^*)$, we have that:

$$\begin{aligned} \|(\mathbf{1} - q^*)\|_\infty &\geq \frac{1}{6^n n 2^{2|P|}} \\ &> 2^{-3|P|} \end{aligned}$$

□

Now

Proof of Theorem 3.14.

Lemma 3.18. *Any variable x_i either depends (directly or indirectly)*[1] *on a variable in a bottom SCC S such that $P_S(\mathit{1}) = \mathit{1}$, or it depends (directly or indirectly) on some variable x_j of form* L *with $P_j(x) = p_{j,0} + \sum_{j=1}^n p_{i,j} x_j$ where $\sum_{j=0}^m p_{i,j} < 1$.*

Proof. Suppose that in the set of variables x_i depends on, D_i, every variable of form L, x_j, with $P_j(x) = p_{j,0} + \sum_{k=1}^n p_{j,k} x_k$ has $\sum_{j=0}^m p_{i,j} = 1$. Then we can verify that $P_{D_i}(\mathbf{1}) = \mathbf{1}$. D_i contains some bottom SCC $S \subseteq D_i$. For this SCC $P_S(\mathbf{1}) = \mathbf{1}$ □

Suppose that x_j is of form L with $P_j(x) = p_{j,0} + \sum_{k=1}^n p_{j,k} x_k$ where $\sum_{k=0}^m p_{j,k} < 1$. Then $q_j^* = P(q^*)_j$ has $q_j^* \leq \sum_{k=0}^m p_{j,k}$. $1 - \sum_{k=0}^m p_{j,k}$ is a rational with a denominator smaller than the product of the denominators of all the $p_{j,k}$. We have:

$$1 - \sum_{k=0}^m p_{j,k} \geq 2^{-|P|}$$

Thus in such a case:

$$q_j^* \leq 1 - 2^{-|P|}$$

[1] meaning that in the dependency graph the other variable's node can be reached from the node corresponding to x_i.

Lemma 3.18 says that any x_i either depends on such a variable, or on a variable to which Theorem 3.15 applies. That is, x_i depends on some x_j with

$$q_j^* \leq 1 - 2^{-3|P|}$$

There is some sequence $x_{l_l}, x_{l_2}, \ldots, x_{l_m}$ with $l_1 = j$, $l_2 - i$ and for every $0 \leq k < m$, $P(x_{l_{k+1}})$ contains a term with $x_{l_{k+1}}$. If $x_{l_{k+1}}$ has form $\mathtt{Q}$, then $q^*_{l_{k+1}} \leq q^*_{l_k}$. If $x_{l_{k+1}}$ has form $\mathtt{L}$, then $1 - q^*_{l_{k+1}} \geq p_{l_{k+1}, l_k}(\mathbf{1} - q^*_{l_k})$. By an easy induction:

$$1 - q_i^* \geq \Big(\prod_{x_{l_k} \text{ has form } \mathtt{L}} p_{l_{k+1}, l_k} \Big)(1 - q_j^*)$$

Again, $|P|$ is at least the number of bits describing these rationals p_{l_{k+1}, l_k}, and thus

$$1 - q_i^* \geq 2^{-|P|}(1 - q_j^*)$$

Since we already know that $q_j^* \leq 1 - 2^{-3|P|}$, i.e., that $(\mathbf{1} - q_j^*) \geq 2^{-3|P|}$, we obtain:

$$1 - q_i^* \geq 2^{-|P|} 2^{-3|P|} = 2^{-4|P|}$$

This completes the proof of the theorem. □

We thus get the Main Theorem of this section:

Proof of Theorem 3.2 (**Main Theorem of Section 3.2**). By Lemma 3.11, $\|q^* - x^{(j + \lceil (\log \frac{(\mathbf{1}-q^*)_{\max}}{(\mathbf{1}-q^*)_{\min}}) \rceil)}\|_\infty \leq 2^{-j}$. But by Theorem 3.14, $\lceil (\log \frac{(\mathbf{1}-q^*)_{max}}{(\mathbf{1}-q^*)_{min}}) \rceil \leq \lceil \log \frac{1}{(\mathbf{1}-q^*)_{min}} \rceil \leq \lceil \log 2^{4|P|} \rceil = 4|P|$. □

In section 3.5 we extend Theorem 3.2, to show that, given a PPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < \mathbf{1}$, if we start Newton iteration at $x^{(0)} := \mathbf{0}$, then for all $i \geq 1$, $\|q^* - x^{(32|P|+2+2i)}\|_\infty \leq \frac{1}{2^{2^i}}$. We then use this (explicit) "quadratic convergence" result to show that the quantitative ***decision problem*** for the LFP q^* of PPSs, which asks, given a PPS $x = P(x)$ over n variables, and given a rational number $r \in [0,1]$, decide whether $q_i^* > r$, is decidable *in the unit-cost arithmetic RAM model of computation* in polynomial time (and thus is reducible to PosSLP).

3.3 Polynomial time in the standard Turing model of computation

The previous section showed that for a PPS, $x = P(x)$, using $(4|P| + j)$ iterations of Newton's method starting at $x^{(0)} := 0$, we obtain q^* within additive error 2^{-j}. However, performing even $|P|$ iterations of Newton's method *exactly* may not be feasible in P-time in the *Turing* model, because the encoding size of iterates $x^{(k)}$ can become very large. Specifically, by repeated squaring, the rational numbers representing the iterate $x^{(|P|)}$ may require encoding size exponential in $|P|$.

In this section, we show that we can nevertheless approximate in P-time the LFP q^* of a PPS, $x = P(x)$. We do so by showing that we can *round down* all coordinates of each Newton iterate $x^{(k)}$ to a suitable polynomial length, and still have a well-defined iteration that converges in nearly the same number of iterations to q^*. Throughout this section we assume every PPS is in SNF form.

Definition 3.19. *("Rounded down Newton's method", with rounding parameter h.) Given a PPS, $x = P(x)$, with LFP q^*, where $\mathbf{0} < q^* < \mathbf{1}$, in the "rounded down Newton's method" with integer rounding parameter $h > 0$, we compute a sequence of iteration vectors $x^{[k]}$, where the initial starting vector is again $x^{[0]} := \mathbf{0}$, and such that for each $k \geq 0$, given $x^{[k]}$, we compute $x^{[k+1]}$ as follows:*

1. *First, compute $x^{\{k+1\}} := \mathcal{N}_P(x^{[k]})$, where the Newton iteration operator $\mathcal{N}_P(x)$ was defined in equation (2.2). (Of course we need to show that all such Newton iterations are defined.)*

2. *For each coordinate $i = 1, \ldots, n$, set $x_i^{[k+1]}$ to be equal to the maximum (non-negative) multiple of 2^{-h} which is $\leq \max(x_i^{\{k+1\}}, 0)$. (In other words, round down $x^{\{k+1\}}$ to the nearest multiple of 2^{-h}, while making sure that the result is non-negative.)*

Theorem 3.20 (Main Theorem of Section 3.3). *Given a PPS, $x = P(x)$, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, if we use the rounded down Newton's method with parameter $h = j + 2 + 4|P|$, then the iterations are all defined, for every $k \geq 0$ we have $0 \leq x^{[k]} \leq q^*$, and furthermore after $h = j + 2 + 4|P|$ iterations we have: $\|q^* - x^{[j+2+4|P|]}\|_\infty \leq 2^{-j}$.*

We prove this via some lemmas. The next lemma proves that the iterations are always well-defined, and yield vectors $x^{[k]}$ such that $\mathbf{0} \leq x^{[k]} \leq q^*$. Note however that, unlike Newton iteration using exact arithmetic, we *do not* claim (as in Proposition 2.11) that $x^{[k]}$ converges *monotonically* to q^*. It may not. It turns out we don't need this: all we need is that $0 \leq x^{[k]} \leq q^*$, for all k. In particular, it may not hold that $P(x^{[k]}) \geq x^{[k]}$. For establishing the monotone convergence of Newton's method on MPSs (Proposition 2.11), the fact that $P(x^{(k)}) \geq x^{(k)}$ is key (see [EY09]). However, $P(x^{[k]}) \geq x^{[k]}$ may no longer hold after rounding down. If, for instance, the polynomial $P_i(x)$ has degree 1 (i.e., has form L), then one can show that after any positive number of iterations $k \geq 1$, we will have that $P_i(x^{\{k\}}) = x_i^{\{k\}}$. So, if we are unlucky, rounding down each coordinate of $x^{\{k\}}$ to a multiple of 2^{-h} could indeed give $(P(x^{[k+1]}))_i < x_i^{[k+1]}$.

Lemma 3.21. *If we run the rounded down Newton method starting with $x^{[0]} := \mathbf{0}$ on a PPS, $x = P(x)$, with LFP q^*, $\mathbf{0} < q^* < \mathbf{1}$, then for all $k \geq 0$, $x^{[k]}$ is well-defined and $0 \leq x^{[k]} \leq q^*$.*

Proof. We prove this by induction on k. The base case $x^{[0]} = 0$ is immediate. Suppose the claim holds for k and thus $0 \leq x^{[k]} \leq q^*$. Lemma 3.4 tells us that

$$q^* - x^{\{k+1\}} = (I - B(x^{[k]}))^{-1} \frac{B(q^*) - B(x^{[k]})}{2} (q^* - x^{[k]})$$

Now the fact that $0 \leq x^{[k]} \leq q^*$ yields that each of the following inequalities hold: $(q^* - x^{[k]}) \geq 0$, $B(q^*) - B(x^{[k]}) \geq 0$. Furthermore, by Theorem 3.6, we have that $\rho(B(x^{[k]})) < 1$, and thus that $(I - B(x^{[k]}))$ is non-singular and $(I - B(x^{[k]}))^{-1} \geq 0$. We thus conclude that $q^* - x^{\{k+1\}} \geq 0$, i.e., that $x^{\{k\}} \leq q^*$. The rounding down ensures that $0 \leq x_i^{[k+1]} \leq x_i^{\{k+1\}}$ unless $x_i^{\{k+1\}} < 0$, in which case $x_i^{[k+1]} = 0$. in both cases, we have that $0 \leq x^{[k+1]} \leq q^*$. So we are done by induction. □

The next key lemma shows that the rounded version still makes good progress toward the LFP.

Lemma 3.22. *For a PPS, $x = P(x)$, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, if we apply the rounded down Newton's method with parameter h, starting at $x^{[0]} := \mathbf{0}$, then for all $j' \geq 0$, we have:*

$$||q^* - x^{[j'+1]}||_\infty \leq 2^{-j'} + 2^{-h+1+4|P|}$$

Proof. Since $x^{[0]} := 0$:

$$q^* - x^{[0]} = q^* \leq \mathbf{1} \leq \frac{1}{(\mathbf{1} - q^*)_{\min}} (\mathbf{1} - q^*) \tag{3.8}$$

For any $k \geq 0$, if $q^* - x^{[k]} \leq \lambda(\mathbf{1} - q^*)$, then by Lemma 3.9 we have:

$$q^* - x^{\{k+1\}} \leq (\frac{\lambda}{2})(\mathbf{1} - q^*) \tag{3.9}$$

Observe that after every iteration $k > 0$, in every coordinate i we have:

$$x_i^{[k]} \geq x_i^{\{k\}} - 2^{-h} \tag{3.10}$$

This holds simply because we are rounding down $x_i^{\{k\}}$ by at most 2^{-h}, unless it is negative in which case $x_i^{[k]} = 0 > x_i^{\{k\}}$. Combining the two inequalities (3.9) and (3.10) yields the following inequality:

$$q^* - x^{[k+1]} \leq (\frac{\lambda}{2})(\mathbf{1} - q^*) + 2^{-h}\mathbf{1} \leq (\frac{\lambda}{2} + \frac{2^{-h}}{(\mathbf{1} - q^*)_{\min}})(\mathbf{1} - q^*)$$

Taking inequality (3.8) as the base case (with $\lambda = \frac{1}{(\mathbf{1}-q^*)_{\min}}$), by induction on k, for all $k \geq 0$:

$$q^* - x^{[k+1]} \leq (2^{-k} + \sum_{i=0}^{k} 2^{-(h+i)}) \frac{1}{(\mathbf{1} - q^*)_{\min}} (\mathbf{1} - q^*)$$

But $\sum_{i=0}^{k} 2^{-(h+i)} \leq 2^{-h+1}$ and $\frac{\|\mathbf{1}-q^*\|_\infty}{(\mathbf{1}-q^*)_{\min}} \leq \frac{1}{(\mathbf{1}-q^*)_{\min}} \leq 2^{4|P|}$, by Theorem 3.14. Thus:

$$q^* - x^{[k+1]} \leq (2^{-k} + 2^{-h+1}) 2^{4|P|}\mathbf{1}$$

Clearly, we have $q^* - x^{[k]} \geq 0$ for all k. Thus we have shown that for all $k \geq 0$:

$$\|q^* - x^{[k+1]}\|_\infty \leq (2^{-k} + 2^{-h+1}) 2^{4|P|} = 2^{-k} + 2^{-h+1+4|P|}.$$

□

We can then show the main theorem:

*Proof of Theorem 3.20 (***Main Theorem of Section 3.3***).* In Lemma 3.22 let $j' := j + 4|P| + 1$ and $h := j + 2 + 4|P|$. We have: $\|q^* - x^{[j+2+4|P|]}\|_\infty \leq 2^{-(j+1+4|P|)} + 2^{-(j+1)} \leq 2^{-(j+1)} + 2^{-(j+1)} = 2^{-j}$. □

Corollary 3.23. *Given any PPS, $x = P(x)$, with LFP q^*, we can approximate q^* within additive error 2^{-j} in time polynomial in $|P|$ and j (in the standard Turing model of computation). More precisely, we can compute a vector v, $\mathbf{0} \leq v \leq q^*$, such that $\|q^* - v\|_\infty \leq 1/2^{-j}$.*

Proof. Firstly, by Propositions 2.9, 3.1, and 2.12, we can assume $x = P(x)$ is in SNF form, and that $\mathbf{0} < q^* < \mathbf{1}$. By Theorem 3.20, the rounded down Newton's method with parameter $h = j + 2 + 4|P|$, for $h = j + 2 + 4|P|$ iterations, computes a rational vector $v = x^{[h]}$ such that $v \in [0,1]^n$, and $\|q^* - v\|_\infty \leq 1/2^{-h}$.

Furthermore, for all k, with $0 \leq k \leq h$, $x^{[k]}$ has encoding size polynomial in $|P|$ and j. We then simply need to note that all the linear algebra operations, that is: matrix multiplication, addition, and matrix inversion, required in a single iteration of Newton's method, can be performed exactly on rational inputs in polynomial time and yield rational results with a polynomial size. □

3.4 Norm bounds

Before we can get results about quadratic convergence of Newton's method for PPSs, and the decision problem for PPSs, we need a bound on the norm of the matrix $(I - B(q^*))^-1$ when $q^* < 1$. We use the $\|.\|_\infty$ norm (which is the maximum absolute value row sum).

For a PPS, $x = P(x)$ with n variables, recall that its variable *dependency graph* is defined to be the digraph $H = (V,E)$, with vertices $V = \{x_1,\dots,x_n\}$, such that $(x_i,x_j) \in E$ if and only if in $P_i(x) \equiv \sum_{r \in R_i} p_r x^{v(\alpha_r)}$ there is a coefficient $p_r > 0$ such that $v(\alpha_r)_j > 0$. Intuitively, $(x_i,x_j) \in E$ means that x_i "depends directly" on x_j. An MPS or PPS, $x = P(x)$, is called ***strongly connected*** if its dependency graph H is strongly connected.

The aim of this section is to prove the following Theorem:

Theorem 3.24. *If $x = P(x)$ is a PPS in SNF form with LFP $q^* > \mathbf{0}$, then*
(i) If $q^* < \mathbf{1}$ and $0 \leq y < 1$, then $(I - B(\frac{1}{2}(y+q^*)))^{-1}$ exists and is non-negative, and

$$\|(I - B(\frac{1}{2}(y+q^*)))^{-1}\|_\infty \leq 2^{10|P|} max\,\{2(1-y)_{\min}^{-1}, 2^{|P|}\}$$

(ii) If $q^* = 1$ and $x = P(x)$ is strongly connected (i.e. every variable depends directly or indirectly on every other) and $0 \leq y < 1 = q^*$, then $(I - B(y))^{-1}$ exists and is non-negative, and

$$\|(I - B(y))^{-1}\|_\infty \leq 2^{4|P|} \frac{1}{(1-y)_{\min}}$$

Before proving this Theorem, we shall need to develop some more definitions and lemmas.

Definition 3.25. *A* path *in the dependency graph $H = (V,E)$ of a PPS $x = P(x)$ is a sequence of variables x_{k_1}, ... ,x_{k_m}, with $m \geq 2$, such that $(x_{k_i}, x_{k_{i+1}}) \in E$, for $i \in \{1,\dots,m-1\}$. In other words, for each $i \in \{1,\dots,m-1\}$, $x_{k_{i+1}}$ appears (with a non-zero coefficient) in the polynomial $P_{k_i}(x)$.*

*We say that x_i **depends on** x_j (directly or indirectly) if there is a path in the dependency graph starting at x_i and ending at x_j.*

We shall need to be more quantitative about dependency:

Lemma 3.26. *Given a PPS $x = P(x)$ in SNF form, and variables x_i,x_j:*

(i) If x_i depends on x_j then there is a positive integer k, with $1 \leq k \leq n$, such that

$$(B(1)^k)_{ij} \geq 2^{-|P|}$$

(ii) If $(B(1)^k)_{ij} > 0$ for some positive integer k, with $1 \leq k \leq n$, then x_i depends on x_j.

(iii) If x_i depends on x_j "only via variables of form L*", i.e., if there is a path $x_{l_1}, \ldots, x_{l_m}$ in the dependency graph such that $l_1 = i$ and $l_m = j$, and such that for each $1 \leq h \leq m-1$, $x_{l_h} = P_{l_h}(x) = p_{l_h,0} + \sum_{g=1}^{n} p_{l_h,g} x_g$ has form* L *with $p_{l_h,l_{h+1}} > 0$, then there is a $1 \leq k \leq n$ such that, for any vector x, such that $0 \leq x \leq 1$,*

$$(B(x)^k)_{ij} \geq 2^{-|P|}$$

Proof.

(i) Let the sequence of variables $x_{l_1}, \ldots, x_{l_k}$ constitute a shortest path from x_i and x_j, such that $k \geq 2$. Such a shortest path exists, since x_i depends on x_j. So $x_i = x_{l_1}$, and $x_j = x_{l_k}$, and $x_{l_{h+1}}$ appears in the expression for $P_{l_h}(x)$, and $1 \leq h \leq k-1$. Note that we must have $k \leq n$. Thus $(B(1))_{l_h l_{h+1}} > 0$ for $1 \leq h \leq k-1$. But note that since $B(1)$ is a non-negative matrix, $(B(1)^{k-1})_{ij} \geq \prod_{h=1}^{k-1} (B(1))_{l_h l_{h+1}}$. Since we have chosen a shortest (non-empty) path from x_i to x_j, and since $x = P(x)$ is in SNF form, each $(B(1))_{l_h l_{h+1}}$ that is not exactly 1 must be a distinct rational coefficient in P, not appearing elsewhere along the path, and thus $\prod_{h=1}^{k-1} (B(1))_{l_h l_{h+1}} \geq 2^{-|P|}$.

(ii) For $k \geq 1$, we can expand $(B(1)^k)_{ij}$ into a sum of n^{k-1} terms of the form $\prod_{h=1}^{k} (B(1))_{l_h l_{h+1}}$ with $l_1 = i$, $l_{k+1} = j$ and $(l_2, \ldots, l_k) \in \{1, \ldots, n\}^{k-1}$. At least one of these has $\prod_{h=1}^{k} (B(1))_{l_h l_{h+1}} > 0$. In that case, $x_{h_1}, \ldots, x_{h_{k+1}}$ is a path in the dependency graph starting at x_i and ending at x_j.

(iii) Let us choose $x_{l_1}, \ldots, x_{l_k}$ to be a shortest path from x_i to x_j, with $k \geq 2$, and such that every equation $x_{l_h} = P_{l_h}(x)$ along the path, for all $h \in \{1, \ldots, k-1\}$ has form L. Clearly, we must have $k \leq n$. By monotonicity of $B(z)$ in $z \geq 0$, we have $(B(1)^{k-1})_{ij} \geq B(x)^{k-1}$. Furthermore, since $x_{l_1}, \ldots, x_{l_k}$ is a path from x_i to x_j, we have $(B(x))_{i,j}^{k-1} \geq \prod_{h=1}^{k-1} (B(x))_{l_h l_{h+1}}$. Moreover, since each equation $x_{l_h} = P(x)_{l_h}$ has form L, for every $h \in \{1, \ldots, k-1\}$, we must have $(B(x))_{l_h l_{h+1}} = (B(1))_{l_h l_{h+1}}$ (because all the partial derivatives of linear expressions are constants). But we argued in (i) that, when $x_{l_1}, \ldots, x_{l_k}$ constitutes a shortest path from x_i to x_j, $\prod_{h=1}^{k-1} (B(1))_{l_h l_{h+1}} \geq 2^{-|P|}$.

□

We need a basic result from the Perron-Frobenius theory of non-negative matrices. We are not aware of a source that contains a statement exactly equivalent to (or implying) the following Lemma, so we shall provide a proof, however it is entirely possible (and likely) that such a Lemma has appeared elsewhere. Lemma 19 of [EWY10] provides a similar result for the case when the matrix A is irreducible.

Lemma 3.27. *If A is a non-negative matrix, and vector $u > 0$ is such that $Au \leq u$ and $\|u\|_\infty \leq 1$, and $\alpha, \beta \in (0,1)$ are constants such that for every $i \in \{1,\dots,n\}$, one of the following two conditions holds:*

(I) $(Au)_i \leq (1-\beta)u_i$

(II) there is some k, $1 \leq k \leq n$, and some j, such that $(A^k)_{ij} \geq \alpha$ and $(Au)_j \leq (1-\beta)u_j$.

then $(I - A)$ is non-singular, $\rho(A) < 1$, and

$$\|(I-A)^{-1}\|_\infty \leq \frac{n}{u_{\min}^2 \alpha\beta}$$

Proof. First, suppose that some $i \in \{1,\dots,n\}$, satisfies condition (I). Then, we claim that it satisfies condition (II), except that we must take $k = 0$. Specifically, if we let $k = 0$, then since $A^0 = I$, and $(A^0)_{ii} = I_{ii} = 1 \geq \alpha$, condition (II) boils down to $(Au)_i \leq (1-\beta)u_i$. So, to prove the statement, it suffices to only consider condition (II) but to allow $k = 0$ in that condition.

So, by assumption, given any $i \in \{1,\dots,n\}$, there is some $0 \leq k \leq n$ and some j, such that

$$(A^k)_{ij} \geq \alpha > 0 \tag{3.11}$$

and moreover $(Au)_j \leq (1-\beta)u_j$, which we can rewrite as:

$$u_j - (Au)_j \geq \beta u_j \ \ (> 0 \ ) \tag{3.12}$$

Let $u_{\min} = \min_i u_i$. We thus have that for every i:

$$\begin{array}{rcll}
(A^n u)_i & = & (u - \sum_{l=0}^{n-1} A^l (u - Au))_i & \\
& \leq & (u - A^k(u - Au))_i & \text{(because } A^l \geq 0 \text{ and } (u - Au) \geq 0) \\
& = & (u_i - \sum_{j'=1}^{n} A^k_{ij'}(u_{j'} - (Au)_{j'}) & \\
& \leq & (u_i - A^k_{ij}(u_j - (Au)_j) & \text{(again, because } A^k_{i,j'} \geq 0 \text{ and } (u_{j'} - (Au)_{j'}) \geq 0 \text{ for every } j') \\
& \leq & u_i - \alpha\beta u_j & \text{(by (3.11) and (3.12))} \\
& \leq & u_i - \alpha\beta u_{\min} & \\
& \leq & u_i - u_{\min}\alpha\beta u_i & \text{(recalling that by assumption } ||u||_\infty \leq 1)
\end{array}$$

We have that $A^n u \leq (1 - u_{\min}\alpha\beta)u$. Of course $(1 - u_{\min}\alpha\beta) < 1$. So we have that

$$A^{mn}u \leq (1 - u_{\min}\alpha\beta)^m u$$

For any integer $d \geq 0$, $A^d u \leq u$. Thus also, for every $d \geq 0$,

$$A^d u \leq (1 - u_{\min}\alpha\beta)^{\lfloor \frac{d}{n} \rfloor} u \tag{3.13}$$

We thus have that, as $m \to \infty$, $A^m u \to 0$. Since $u > 0$ and $A \geq 0$, this implies that as $m \to \infty$, $A^m \to 0$ (coordinate-wise), or in other words that $\lim_{m\to\infty} ||A^m||_\infty = 0$. This is equivalent to saying that the spectral radius $\rho(A) < 1$. This implies that the inverse matrix $(I - A)^{-1} = \sum_{k=0}^{\infty} A^k \geq 0$ exists, by Lemma 3.7.

We will use the following easy fact:

Lemma 3.28. *If M is a nonnegative $n \times n$ matrix, $u > 0$ is a vector with $||u||_\infty \leq 1$, and $\lambda > 0$ is a real number satisfying $Mu \leq \lambda u$ then*

$$||M||_\infty \leq \frac{\lambda}{u_{\min}}$$

Proof. Since M is non-negative, $||M||_\infty$ is the maximum row sum of M. There is thus an i such that

$$||M||_\infty = \sum_j m_{ij}$$

where $m_{i,j}$ are the entries of M. For this i:

$$\begin{array}{rcl}
\lambda u_i & \geq & (Mu)_i \\
& = & \sum_j m_{ij} u_j \\
& \geq & \sum_j m_{ij} u_{\min} \\
& = & ||M||_\infty u_{\min}
\end{array}$$

but $u_i \leq 1$ giving us $||M||_\infty \leq \frac{\lambda}{u_{\min}}$. □

Now we can complete the proof of Lemma 3.27:

$$\begin{aligned}
(I-A)^{-1}u = (\sum_{k=0}^{\infty} A^k)u &= \sum_{k=0}^{\infty} A^k u \\
&\leq \sum_{k=0}^{\infty} (1-u_{\min}\alpha\beta)^{\lfloor \frac{k}{n} \rfloor} u \qquad \text{(by (3.13))} \\
&= (\sum_{m=0}^{\infty} n(1-u_{\min}\alpha\beta)^m u \\
&= n\frac{1}{u_{\min}\alpha\beta}u
\end{aligned}$$

the last equality holding because the geometric series sum gives $\sum_{m=0}^{\infty}(1-u_{\min}\alpha\beta)^m = \frac{1}{u_{\min}\alpha\beta}$. Lemma 3.28, with $M := (I-A)^{-1} = \sum_{k=0}^{\infty} A^k$, and $\lambda := n\frac{1}{u_{\min}\alpha\beta}$, now yields:

$$||(I-A)^{-1}||_\infty \leq n\frac{1}{u_{\min}^2\alpha\beta}$$

and this completes the proof of Lemma 3.27. □

Proof of Theorem 3.24. Before we start to prove cases **(i)** and **(ii)** of the Theorem we need to develop some more lemmas.

Proposition 3.29. *For a PPS, $x = P(x)$, with LFP $q^* > \mathbf{0}$, for every variable x_i either $P_i(0) > 0$ or x_i depends on a variable x_j with $P_j(0) > 0$.*

Proof. Suppose, for contradiction, that a variable x_i has $P_i(0) = 0$ and depends only on variables x_j which have $P_j(0) = 0$. Then $P_i^n(0) = 0$ for all n. But $P^n(0) \to q^*$ as $n \to \infty$ (see, e.g., Theorem 3.1 from [EY09]). So $q_i^* = 0$. □

The case when all the equations, $x_i = P_i(x)$, are linear has to be treated a little differently, and we tackle that first:

Lemma 3.30. *If $x = P(x)$ is a PPS in SNF form that has no equations of form* Q*, and has LFP $q^* > \mathbf{0}$, then*

$$||(I-B)^{-1}||_\infty \leq n2^{2|P|}$$

where B is the constant Jacobian matrix of $P(x)$, (i.e., $B = B(x)$ for all x).

Proof. First, note that B is a sub-stochastic matrix i.e. $B1 \leq 1$. We will now call a variable, x_i, *leaky*, if $(B1)_i < 1$. Note that since $P_i(x) \equiv \sum_{i=1}^{n} p_{i,j}x_j + p_{i,0}$, this means that $(B1)_i = \sum_{j=1}^{n} \frac{\partial P_i(x)}{\partial x_j} = \sum_{j=1}^{n} p_{i,j} < 1$.

Note that since $q^* > \mathbf{0}$, it must be the case that for every variable x_i, either x_i itself is leaky, or x_i depends (possibly indirectly) on a leaky variable x_j. This is because if a variable x_i doesn't satisfy this, then $q_i^* = 0$, which can't be the case.

Since the entries of B are either 0, 1, or coefficients $p_{i,j}$ from $P(x)$, we see that for every *leaky* variable x_i, we have that $(B1)_i = \sum_{j=1}^n p_{i,j} \leq (1 - 2^{-|P|})$ holds.[2]

For any *non-leaky* variable x_r, there is a leaky variable x_i that x_r depends on. x_r does not depend on any variables of form Q. Thus, by Lemma 3.26 (iii), there is a k, $1 \leq k \leq n$, such that $((B)^k)_{ri} \geq 2^{-|P|}$.

We can thus apply Lemma 3.27 with matrix $A := B$ and vector $u := 1$, with $\alpha := \beta := 2^{-|P|}$, because we have just established that condition (I) of that Lemma applies to leaky variables x_i, and condition (II) of that Lemma applies to non-leaky variables. Thus Lemma 3.27 give us that

$$\|(I-B)^{-1}\|_\infty \leq (\frac{1}{1_{\min}})^2 n 2^{2|P|}$$

Of course, $1_{\min} = 1$. □

We are now ready to prove parts **(i)** and **(ii)** of Theorem 3.24.

(i) When $q^* < \mathbf{1}$, we can say something stronger than Proposition 3.29.

Lemma 3.31. *For any PPS, x=P(x), in SNF form, with LFP $\mathbf{0} < q^* < \mathbf{1}$, for any variable x_i, either:*

(I) the equation $x_i = P_i(x)$ is of form Q*, or else $P_i(1) < 1$. Or,*

(II) x_i depends (directly or indirectly) on a variable x_j, such that $x_j = P_j(x)$ is of form Q*, or else $P_j(1) < 1$.*

Proof. Suppose, for contradiction, that there is a variable x_i for which neither (I) nor (II) holds. Let D_i be the set of variables that x_i depends on, unioned together with $\{x_i\}$ itself. For any vector x, consider the subvector x_{D_j}, which consists of the components of x with coordinates in D_i. We can consider the subset of the equations $x_{D_i} = P_{D_i}(x)$. By transitivity of dependency, $P_{D_i}(x)$ contains only terms in the variables x_{D_i}. So $x_{D_i} = P_{D_i}(x) = P_{D_i}(x_{D_i})$ is itself a PPS. Since by assumption neither (I) nor (II) hold for x_i, we have that $x_{D_i} = P_{D_i}(x_{D_i})$ contains no equations of form Q and $P_{D_i}(1) = 1$. Since,

[2] This inequality holds because we assume each positive input probability $p_{i,j}$ is represented as a ratio $\frac{a_j}{b_j}$ of positive integers in the encoding of $x = P(x)$, and thus $1 - \sum_{j=1}^n \frac{a_j}{b_j}$ can be represented as a ratio $\frac{a}{b}$ of two positive integers where the denominator is $b = \prod_{j=1}^n b_j$. But then $(1 - \sum_{j=1}^n \frac{a_j}{b_j}) = \frac{a}{b} \geq 1/\prod_{j=1}^n b_j \geq \frac{1}{2^{|P|}}$.

therefore, $P_{D_i}(x_{D_i})$ is linear, we can rewrite $x_{D_i} = P_{D_i}(x_{D_i})$ as $x_{D_i} = B_{D_i}x_{D_i} + P_{D_i}(0)$ and hence $(I - B_{D_i})x_{D_i} = P_{D_i}(0)$. Lemma 3.30 applied to the PPS $x_{D_i} = P_{D_i}(x_{D_i})$ gives us that, in particular, $(I - B_{D_i})$ is non-singular. Consequently $x_{D_i} = P_{D_i}(x_{D_i})$ has a unique solution. But we already said that 1 is a solution, $P_{D_i}(1) = 1$, and so $q^*_{D_i} = 1$. This contradicts $q^* < \mathbf{1}$. So there can be no x_i for which neither (I) nor (II) holds. □

To obtain the conclusion of case **(i)** of Theorem 3.24, assuming all of the premises of the Theorem's statement, we will now aim to use Lemma 3.27, applied to $A := B(\frac{1}{2}(y+q^*)$, and $u := \mathbf{1} - q^*$.

By Lemma 3.31, every variable x_i either depends on a variable, or is itself equal to a variable, x_j, such that $x_j = P_j(x)$ is of form `Q` or $P_j(1) < 1$. We can clearly assume that such a dependence is linear in the sense of Lemma 3.26 (iii), and thus for any x_i there is a $0 \leq k \leq n$ with $(B(1)^k)_{ij} \geq 2^{-|P|}$, for some x_j with either $x_j = P_j(x)$ of form `Q` or $P_j(1) < 1$.

We need to show and that for such an x_j we have $(B(\frac{1}{2}(y+q^*))(\mathbf{1}-q^*) < \mathbf{1}-q^*$.

For any variable x_j such that $x_j = P_j(x)$ has form `Q`, we have that $x_j = x_k x_l$ for some variables k and l. Thus, since $\frac{\partial P_j(x)}{\partial x_k} = x_l$ and $\frac{\partial P_j(x)}{\partial x_l} = x_k$, we have that:

$$\begin{aligned}
& (B(\frac{1}{2}(q^*+y))(\mathbf{1}-q^*))_j \\
= \;& \frac{1}{2}(q^*_k + y_k)(\mathbf{1}-q^*_l) + \frac{1}{2}(q^*_l + y_l)(\mathbf{1}-q^*_k) \\
= \;& \frac{1}{2}((q^*_k+1)-(1-y_k))(\mathbf{1}-q^*_l) + \frac{1}{2}((q^*_l+1)-(1-y_l))(\mathbf{1}-q^*_k) \\
= \;& \frac{1}{2}((q^*_k+1)(\mathbf{1}-q^*_l) - (1-y_k)(\mathbf{1}-q^*_l) + (q^*_l+1)(\mathbf{1}-q^*_k) - (1-y_l)(\mathbf{1}-q^*_k)) \\
= \;& \frac{1}{2}(2 - 2q^*_k q^*_l - (1-y_l)(\mathbf{1}-q^*_k) - (1-y_k)(\mathbf{1}-q^*_l)) \\
\leq \;& \frac{1}{2}(2 - 2q^*_k q^*_l - (1-y)_{\min}((\mathbf{1}-q^*)_k + (\mathbf{1}-q^*)_l)) \\
\leq \;& \frac{1}{2}(2 - 2q^*_k q^*_l - (1-y)_{\min}((\mathbf{1}-q^*)_k + (\mathbf{1}-q^*)_l - (\mathbf{1}-q^*)_k(\mathbf{1}-q^*)_l)) \\
= \;& (\mathbf{1}-q^*_j) - \frac{1}{2}(1-y)_{\min}(\mathbf{1}-q^*_j) \\
= \;& (1 - \frac{1}{2}(1-y)_{\min})(1-q^*)_j
\end{aligned}$$

If, on the other hand, x_j has $P_j(1) < 1$, then $x_j = P_j(1)$ has form `L`, and, as in the proof of Lemma 3.30, and specifically footnote (2), we must have

$$P_j(1) \leq 1 - 2^{-|P|} \tag{3.14}$$

We thus have that:

$$\begin{aligned}
(B(\frac{1}{2}(q^*+y))(\mathbf{1}-q^*))_j &= \sum_{l=1}^{n} p_{j,l}(\mathbf{1}-q^*)_l \\
&= (\sum_{l=1}^{n} p_{j,l}) + p_{j,0} - (\sum_{l=1}^{n} p_{j,l} q^*_l) - p_{j,0} \\
&= P_j(1) - P_j(q^*) \\
&= P_j(1) - q^*_j \\
&\leq (1-2^{-|P|}) - q^*_j \qquad \text{(by (3.14))} \\
&= (1-q^*)_j - 2^{-|P|} \\
&\leq (1-2^{-|P|})(1-q^*)_j
\end{aligned}$$

To be able to apply Lemma 3.27, it only remains to show that $B(\frac{1}{2}(y+q^*)))(\mathbf{1}-q^*) \leq (\mathbf{1}-q^*)$. But this is just Lemma 3.5. Since $0 \leq y < 1$, it follows by monotonicity of $B(z)$ in z that $B(\frac{1}{2}(y+q^*)))(\mathbf{1}-q^*) \leq (\mathbf{1}-q^*)$.

Thus, we can apply Lemma 3.27, by setting $A := B(\frac{1}{2}(y+q^*))$, $u := (\mathbf{1}-q^*)$, $\alpha := 2^{-|P|}$, $\beta := \min\{\frac{1}{2}(1-y)_{\min}, 2^{-|P|}\}$, and we obtain:

$$\|(I - B(\frac{1}{2}(y+q^*)))^{-1}\|_\infty \leq n(\mathbf{1}-q^*)^{-2}_{\min} \max\ \{2(1-y)^{-1}_{\min}, 2^{|P|}\} 2^{|P|}$$

Recall that, by Theorem 3.14, $(\mathbf{1}-q^*)_{\min} \geq 2^{-4|P|}$. Thus

$$\begin{aligned}
\|(I - B(\frac{1}{2}(y+q^*)))^{-1}\|_\infty &\leq n2^{9|P|}\max\ \{2(1-y)^{-1}_{\min}, 2^{|P|}\} \\
&\leq 2^{10|P|}\max\ \{2(1-y)^{-1}_{\min}, 2^{|P|}\}
\end{aligned}$$

We now prove part **(ii)** of Theorem 3.24. If $x = P(x)$ is strongly connected, then if there is an x_i with $x_i = P_i(x)$ of form $\mathtt{Q}$, then every variable depends on it. If there are no such variables, then Lemma 3.30 gives that, for any $x \in \mathbb{R}^n$, $\|I - B(x)\|_\infty \leq n2^{2|P|}$ and we are done. So we can assume that there is an x_i with $x_i = P_i(x)$ of form $\mathtt{Q}$. We quote the following from [EY09]:

Lemma 3.32 (see proof of Theorem 8.1 in [EY09]). *If $x = P(x)$ is strongly connected and $q^* > \mathbf{0}$, then $q^* = 1$ if and only if $\rho(B(1)) \leq 1$.*

$B(1)$ is a non-negative irreducible matrix. Perron-Frobenius theory gives us that there is a positive eigenvector $v > 0$, with associated eigenvalue $\rho(B(1))$, the spectral radius of $B(1)$, i.e., such that $B(1)v = \rho(B(1))v$. But $\rho(B(1)) \leq 1$ so $B(1)v \leq v$.

Lemma 3.33 (cf Lemma 5.9 of [EKL10]). $\frac{\|v\|_\infty}{v_{\min}} \leq 2^{|P|}$.

Proof. For any x_i, x_j, there is some $1 \leq k \leq n$ with $(B(1)^k)_{ij} > 0$. We know that $B(1)^k v \leq v$. So $(B(1)^k)_{ij} v_j \leq (B(1)^k v)_i = \rho(B(1))^k v_i \leq v_i$. But by Lemma 3.26 (ii), $(B(1)^k)_{ij} \geq 2^{-|P|}$. So $\frac{v_j}{v_i} \leq 2^{|P|}$. There are v_i,v_j that achieve $v_i = v_{\min}$ and $v_j = \|v\|_\infty$, so we are done. □

We can normalise the top eigenvector, v, so we can assume that $\|v\|_\infty = 1$. Then $v_{\min} \geq 2^{-|P|}$. Consider any equation $x_i = P_i(x) = x_j x_k$ of form $\mathtt{Q}$ (we have already dealt with the case where no such equation exists):

$$\begin{aligned}
(B(y)v)_i &= y_j v_k + y_k v_j \\
&\leq y_{\max} v_k + y_{\max} v_j \qquad \text{(where } y_{\max} := \max_r y_r\text{)} \\
&\leq (1-(1-y)_{\min})(v_k + v_j) \\
&= (1-(1-y)_{\min})(B(1)v)_i \\
&= (1-(1-y)_{\min})\rho(B(1))v_i \\
&\leq (1-(1-y)_{\min})v_i \qquad \text{(because } \rho(B(1)) \leq 1\text{)}
\end{aligned}$$

Now we can apply Lemma 3.27, with $A := B(y)$, $u := v$, $\alpha := 2^{-|P|}$, and $\beta := (1-y)_{\min}$, to obtain that:

$$\|(I - B(y))^{-1}\|_\infty \leq n v_{\min}^{-2} (1-y)_{\min}^{-1} 2^{|P|}$$

Inserting our bound for $v_{\min}$, namely $v_{\min} \geq 2^{-|P|}$, yields:

$$\begin{aligned}
\|(I - B(y))^{-1}\|_\infty &\leq n 2^{3|P|} (1-y)_{\min}^{-1} \\
&\leq 2^{4|P|} (1-y)_{\min}^{-1}
\end{aligned}$$

□

3.5 Quadratic convergence & decision problems for PPSs

In this section we extend Theorem 3.2 to a *quadratic convergence* result for Newton's method on PPSs, with all constants explicit. Namely, given a PPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < \mathbf{1}$, if we start Newton iteration at $x^{(0)} := \mathbf{0}$, then for all $i \geq 1$, we have

$$\|q^* - x^{(32|P|+2+2i)}\|_\infty \leq \frac{1}{2^{2^i}}$$

We then use this result to show that the ***decision problem*** for the LFP q^* of PPSs, which asks, given a PPS $x = P(x)$ over n variables, and given a rational number $r \in [0,1]$, decide whether $q_i^* > r$ (or whether $q_i^* \geq r$) is decidable *in the unit-cost arithmetic*

RAM model of computation in polynomial time, and thus this decision problem is itself reducible to the PosSLP problem. We in fact show further that deciding whether $q_i^* > r$ is P-time many-one (Karp) reducible to PosSLP.

We assume throughout this section, w.l.o.g., that every PPS, $x = P(x)$, is in *simple normal form*, and that the LFP, q^* satisfies $\mathbf{0} < q^* < \mathbf{1}$.

Corollary 3.34. *If $x = P(x)$ is a PPS with LFP q^*, and $\mathbf{0} < q^* < \mathbf{1}$, then $(I - B(q^*))^{-1}$ exists and is non-negative, and*

$$\|(I - B(q^*))^{-1}\|_\infty \leq 2^{14|P|+1}$$

Proof. Applying part (i) of Theorem 3.24, and letting $y := q^*$, we obtain

$$\begin{aligned}
\|(I - B(q^*))^{-1}\|_\infty &\leq 2^{10|P|} \max\{2(\mathbf{1} - q^*)^{-1}_{\min}, 2^{|P|}\} \\
&\leq 2^{10|P|} \max\{2(2^{-4|P|})^{-1}, 2^{|P|}\} \qquad \text{(by Theorem 3.14)} \\
&= 2 \cdot 2^{14|P|} = 2^{14|P|+1}.
\end{aligned}$$

□

Lemma 3.35. *If $x = P(x)$ is a PPS with n variables in simple normal form (SNF), with LFP $\mathbf{0} < q^* < \mathbf{1}$, then for any $z \in \mathbb{R}^n$ such that $0 \leq z \leq q^*$, then*

$$\|q^* - \mathcal{N}(z)\|_\infty \leq 2^{14|P|+1} \|q^* - z\|_\infty^2$$

Proof. Let us first note that

$$\|\frac{B(q^*) - B(z)}{2}\|_\infty \leq |(q^* - z)\|_\infty \tag{3.15}$$

This holds because $x = P(x)$ is in SNF form, and thus every equation $x_i = P_i(x)$ is either of the form $x_i = x_j x_k$, or else it is a *linear* (affine) equation, of the form $x_i = \sum_{j=1}^n p_j x_j + p_0$. Now, for every i with a nonlinear equation, i.e., where $P_i(x) \equiv x_j x_k$, the i'th row of the Jacobian matrix $B(x)$, contains exactly two non-zero entries: one is $x_j = \frac{\partial P_i(x)}{\partial x_k}$ and the other is $x_k = \frac{\partial P_i(x)}{\partial x_j}$. Thus, if we define the matrix $A = \frac{B(q^*) - B(z)}{2}$, we must have $\sum_{r=1}^n |A_{i,r}| = \frac{(q_j^* - z_j) + (q_k^* - z_k)}{2} \leq \|q^* - z\|_\infty$. Furthermore, for every i with a *linear* equation, the i'th row of the Jacobian matrix $B(x)$ consists of only constants that do not depend on x, and thus in that case $\sum_{r=1}^n |A_{i,r}| = 0 \leq \|q^* - z\|_\infty$. Thus inequality (3.15) holds.

Now, using Lemma 3.4, and the equation it gives, namely:

$$q^* - \mathcal{N}(z) = (I - B(z))^{-1} \frac{B(q^*) - B(z)}{2} (q^* - z) \tag{3.16}$$

and taking norms on both sides of this equation, we have:

$$\begin{aligned}
\|q^* - \mathcal{N}(z)\|_\infty &= \|(I - B(z))^{-1} \frac{B(q^*) - B(z)}{2} (q^* - z)\|_\infty \\
&\leq \|(I - B(z))^{-1}\|_\infty \| \frac{B(q^*) - B(z)}{2} \|_\infty \|(q^* - z)\|_\infty \\
&\leq 2^{14|P|+1} \| \frac{B(q^*) - B(z)}{2} \|_\infty \|(q^* - z)\|_\infty \qquad \text{(by Corollary 3.34)} \\
&\leq 2^{14|P|+1} \|(q^* - z)\|_\infty^2 \qquad \text{(by inequality (3.15))}
\end{aligned}$$

□

Theorem 3.36. *Let $x = P(x)$ be any PPS in SNF form, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$. If we start Newton iteration at $x^{(0)} := \mathbf{0}$, with $x^{(k+1)} := \mathcal{N}_P(x^{(k)})$, then for any integer $i \geq 1$ the following inequality holds:*

$$\|q^* - x^{(18|P|+2+i)}\|_\infty \leq \frac{1}{2^{14|P|+1+2^i}} \leq \frac{1}{2^{2^i}} \ .$$

Proof. Lemma 3.35 does not gain us much unless $\|q^* - z\|_\infty \leq \frac{1}{2^{14|P+1}}$. We need to use our previous linear convergence result until we are close enough for quadratic convergence to kick in. By Theorem 3.2, for $18|P| + 2 = (14|P| + 2) + 4|P|$, we have $\|q^* - x^{(18|P|+2)}\|_\infty \leq \frac{1}{2^{(14|P|+2)}}$ and so

$$2^{14|P+1} \|q^* - x^{(18|P|+2)}\|_\infty \leq \frac{1}{2} \tag{3.17}$$

Lemma 3.35 tell us that:

$$\|q^* - \mathcal{N}(z)\|_\infty \leq 2^{14|P|+1} \|q^* - z\|_\infty^2$$

Multiplying both sides by $2^{14|P|+1}$ gives:

$$2^{14|P|+1} \|q^* - \mathcal{N}(z)\|_\infty \leq (2^{14|P|+1} \|q^* - z\|_\infty)^2$$

By induction, for any integers $i \geq 0$, $m \geq 0$

$$2^{14|P|+1} \|q^* - x^{(i+m)}\|_\infty \leq (2^{14|P|+1} \|q^* - x^{(m)}\|_\infty)^{2^i}$$

Taking $m = 18|P| + 2$, we can use equation 3.17:

$$2^{14|P|+1} \|q^* - x^{(18|P|+2+i)}\|_\infty \leq \frac{1}{2}^{2^i}$$

and so

$$\|q^* - x^{(18|P|+2+i)}\|_\infty \leq \frac{1}{2^{(14|P|+1+2^i)}}$$

□

We next wish to use Theorem 3.36 in order to establish that, using Newton's method with *exact* arithmetic, *in the unit-cost arithmetic RAM model of computation*, we can decide, given a rational number r, whether $q_i^* \geq r$, in time polynomial in $|P|$ and the encoding size of r.

To do this, we need to first establish a separation bound relating to q^* and a given rational r.

Lemma 3.37. *Given a PPS, $x = P(x)$, with n variables, and with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, and given a rational number $r > 0$, where $r = \frac{a}{b} < 1$ is represented as the ratio of positive integers a and b given in binary, with $a \leq b$, then for any $k \in \{1, \ldots, n\}$, if $q_k^* \neq r$, then*

$$|q_k^* - r| \geq 2^{-2(n+1)(\max\{|P|, \log(b)\} + 2(n+1)\log(2n+2))5^n}$$

Proof. We shall use the following Theorem from [HKL+11] regarding explicit separation bounds for isolated real-valued solutions to polynomial systems of equations:

Theorem 3.38. *(Theorem 23 from [HKL+11]) Consider a polynomial system of equations*

$$(\Sigma) \qquad g_1(x_1, \ldots, x_n) = \cdots = g_m(x_1, \ldots, x_n) = 0 \ , \tag{3.18}$$

with polynomials of degree at most d and integer coefficients of magnitude at most 2^τ.

Then, the coordinates of any isolated *(in Euclidean topology) real solutions of the system are real algebraic numbers of degree at most $(2d+1)^n$, and their defining polynomials have coefficients of magnitude at most $2^{2n(\tau + 4n\log(dm))(2d+1)^{n-1}}$. Also, if $\gamma_j = (\gamma_{j,1}, \cdots, \gamma_{j,n})$ is an isolated solution of (Σ), then for any i, either*

$$2^{-2n(\tau+2n\log(dm))(2d+1)^{n-1}} < |\gamma_{j,i}| \quad \text{or} \quad \gamma_{j,i} = 0 \ . \tag{3.19}$$

Moreover, given coordinates of isolated solutions of two such systems, if they are not identical, they differ by at least

$$\operatorname{sep}(\Sigma) \geq 2^{-3n(\tau+2n\log(dm))(2d+1)^{2n-1} - \frac{1}{2}\log(n)} \ . \tag{3.20}$$

To apply Theorem 3.38, we need the fact that $q^* > \mathbf{0}$ is an isolated solution of the PPS. This follows immediately from a more general *unique fixed point* theorem established in [EY12] (Theorem 18 of [EY12]) for the equations corresponding to the termination probabilities of a general recursive Markov chains, and it also follows from (variants of) older results about multi-type branching processes (see [Har63], Thm.

II.7.2 and Corollary II.7.2.1) PPSs correspond to the special case of MPS equations for 1-exit Recursive Markov Chains.

Specifically, the unique fixed point theorem of [EY12] establishes that, in particular, if a PPS has LFP q^* with $\mathbf{0} < q^* < \mathbf{1}$, then q^* is the unique solution of $x = P(x)$ in the interior of $[0,1]^n$, i.e., in $(0,1)^n$. Thus, it is clearly an isolated solution.

For each x_i, let d_i be the product of the denominators of all coefficients of $P_i(x)$. Then $d_i x = d_i P_i(x)$ clearly has integer coefficients which are no larger than $2^{|P|}$. Also, consider a new variable y, and a new equation $y = x_k - r$, where $r = \frac{a}{b}$ is the given positive rational value. This equation is clearly equivalent to $by = bx_k - a$. Suppose the PPS, $x = P(x)$, has LFP $q^* > \mathbf{0}$, and for any $k \in \{1,\dots,n\}$, consider the system of $n+1$ polynomial equations, in $n+1$ variables (with an additional variable y), given by:

$$d_i x_i = d_i P_i(x) \text{ ,for all } i \in \{1,\dots,n\}; \text{ and } \quad by = bx_k - a \, . \tag{3.21}$$

Since $\mathbf{0} < q^* < \mathbf{1}$, we know from the unique fixed point theorem of [EY12] that q^* is an isolated solution of $x = P(x)$. If $z \in \mathbb{R}^n$ is any solution vector for $x = P(x)$, there is a unique $w \in \mathbb{R}$ such that $x := z$ and $y := w$ forms a solution to the equations (3.21); namely let $w = z_k - r$. So, letting $x := q^*$, and letting $y := q^*_k - r$, gives us an isolated solution of the equations (3.21). We can now apply Theorem 3.38 to the system (3.21). Since $y := q^*_k - r$, equation (3.19) in Theorem 3.38 says that

$$2^{-2(n+1)(\max\{|P|,\log(b)\}+2(n+1)\log(2n+2))5^n} < |q^*_k - r| \, , \quad \text{or else} \quad q^*_k - r = 0 \ .$$

which is just what we wanted to establish. □

We are now ready to establish the following:

Theorem 3.39. *Given a PPS, $x = P(x)$, with n variables, and with LFP $\mathbf{0} < q^* < \mathbf{1}$, and given a rational number $r = a/b \in (0,1]$, where a and b are positive integers given in binary. Let $g = 32|P| + 4 + 6n + 56(\lceil\log(n)\rceil + \lceil\log(|P|)\rceil + \lceil\log(\log b))\rceil)$. Let $x^{(i)}$ denote the i'th Newton iterate starting at $x^{(0)} := 0$, applied to the PPS $x = P(x)$. Let $m := 2 + 3n + 28(\lceil\log(n)\rceil + \lceil\log(|P|)\rceil + \lceil\log(\log b))\rceil)$. Then for any $k \in \{1,\dots,n\}$,*

1. *$q^*_k > r$ if and only if $x^{(g)}_k > r$.*
2. *$q^*_k < r$ if and only if $x^{(g)}_k + 2 \cdot \frac{1}{2^{2^m}} < r$.*

Proof. Let $\gamma = 2^{-2(n+1)(\max\{|P|,\log(b)\}+2(n+1)\log(2n+2))5^n}$. Recall that Lemma 3.37 tells us that $|q^*_k - r| \geq \gamma$, for any k, unless $q^*_k = r$. We know $x^{(g)} \leq q^*$. Furthermore, g has been chosen so that, by Theorem 3.36, $\|q^* - x^{(g)}\|_\infty < \frac{1}{2^{2^m}} < \gamma/8$.

To establish (1.), in one direction we simply note that if $x_k^{(g)} > r$, then since $q_k^* > x_k^{(g)}$, we must have $q_k^* > r$. In the other direction, if $q_k^* > r$, then $q_k^* - r \geq \gamma$, but we know $q_k^* - x_k^{(g)} \leq \gamma/8$, so $x_k^{(g)} \geq r + \frac{7}{8}\gamma \geq r$.

To establish (2.), in one direction since $q_k^* - x_k^{(g)} < \frac{1}{2^{2^m}}$, we have $q_k^* < x_k^{(g)} + 2 \cdot \frac{1}{2^{2^m}}$, and thus if $x_k^{(g)} + 2 \cdot \frac{1}{2^{2^m}} < r$, then $q_k^* < r$. In the other direction, if $q_k^* < r$, then since $r - q_k^* \geq \gamma$, and since $q_k^* \geq x_k^{(g)}$, and since $2 \cdot \frac{1}{2^{2^m}} \geq \gamma/4$, we have $x_k^{(g)} + 2 \cdot \frac{1}{2^{2^m}} \geq \gamma/4 < r$. This completes the proof. □

Corollary 3.40. *Given a PPS, $x = P(x)$, with n variables, and with LFP $q^* \in [0,1]^n$, given a coordinate $k \in \{1,\ldots,n\}$, and given a rational number $r \in [0,1]$, there is an algorithm that determines which of the following cases holds: (A) $q_k^* < r$, or (B) $q_k^* = r$, or (C) $q_k^* > r$.*
The algorithm runs in time polynomial in $|P|$, the bit encoding size of the PPS, and size(r), the binary encoding size of r, in the unit-cost arithmetic RAM model of computation.

Thus, in particular, deciding whether $q_k^* \geq r$ is in $\mathbf{P}^{PosSLP}$. Furthermore, deciding whether $q_k^* > r$, or deciding whether $q_k^* < r$, are both P-time many-one (Karp) reducible to PosSLP.

Thus, since the problem of deciding whether $q_k^* > r$, and deciding whether $q_k^* < r$, are already known to be PosSLP-hard under many-one reductions (Theorem 5.3 of [EY09]), it follows that both these problems are P-time equivalent to PosSLP.

Proof. First, recall that deciding whether $q_k^* = 0$ and whether $q_k^* = 1$, can be carried out in P-time (see Propositions 2.12 and 3.1). Hence, we can detect and remove in P-time all variables x_i such that $q_i^* \in \{0,1\}$. Then we are left with a residual PPS, $x = P(x)$, with LFP q^* such that $\mathbf{0} < q^* < \mathbf{1}$.

Notice that each iteration of Newton's method, $x^{(j+1)} = \mathcal{N}(x^{(j)}) = x^{(j)} + (I - B(x^{(j)}))^{-1}(P(x^{(j)}) - x^{(j)})$, on a PPS, $x = P(x)$ with n variables, can be computed by performing a $n \times n$ matrix inversion and matrix-vector multiplication and summing of vectors. Thus, using Cramer's rule to express the matrix inverse as the ratio of matrix determinants, each iteration can be computed by an arithmetic circuit over basis $\{+,-,*,/\}$ with polynomially many gates (as a function of n), given the previous iteration vector $x^{(j)}$ as input. Thus, it can be performed by polynomially many arithmetic operations.

Now we apply Theorem 3.39. Since the g given in the statement of Theorem 3.39 is polynomial in $|P|$ and $size(r)$ (in fact, even in $\log(size(r))$) we can compute $x^{(g)}$ in polynomial time in the unit-cost arithmetic RAM model of computation. Likewise, since the m given in the statement of the Theorem is also polynomial in $|P|$ and $size(r)$, we can use repeated squaring to compute $\frac{1}{2^{2^m}}$ in time polynomial in $|P|$ and $size(r)$ (i.e., with polynomially many arithmetic operations). We can also add two numbers at unit-cost to obtain $x_k^{(g)} + 2 \cdot \frac{1}{2^{2^m}}$.

In order to determine whether $q_k^* > r$, we simply need to check whether $x_k^{(g)} > r$, and to determine whether $q_k^* < r$ we simply need to check whether $x_k^{(g)} + 2 \cdot \frac{1}{2^{2^m}} < r$. Finally, note that $q_k^* = r$ holds precisely when neither $q_k^* > r$ nor $q_k^* < r$ holds.

To conclude that these problems can be decided in $\mathbf{P}^{PosSLP}$, we simply note that it was established by Allender et. al. in [ABKPM09] that every discrete decision problem (with rational valued inputs) that can be decided in P-time in the unit-cost arithmetic RAM model of computation can be decided in $\mathbf{P}^{PosSLP}$.

Lastly, we conclude that deciding whether $q_k^* > r$, and deciding whether $q_k^* < r$, are actually P-time *many-one* (Karp) reducible to PosSLP. This holds for the following reasons. If $r \in \{0, 1\}$, we have already pointed out that deciding both $q_k^* > r$ and $q_k^* < r$ is solvable in (strongly) polynomial time ([EY09, EGK13]), thus there is nothing to prove in this case.

So, suppose $r \in (0, 1)$, and suppose that $\mathbf{0} < q^* < \mathbf{1}$. In this case, we have established that $q_k^* > r$ if and only if $x_k^{(g)} > r$.

As shown in [ABKPM09] (see also [EY09]), division gates in arithmetic circuits over $\{+, -, *, /\}$ can be removed by keeping track of numerators and denominators separately. Thus, overall (the numerator and denominator of) the rational coordinate $x_k^{(g)}$ of the vector $x^{(g)}$ can be computed by a polynomial-sized arithmetic circuit which can be constructed in P-time given $x = P(x)$. Obviously the rational number r can also have its numerator and denominator represented this way in P-time. Consequently, to decide whether $x_k^{(g)} > r$ (likewise, whether $x_k^{(g)} < r$), we simply need to compare the output value of two (P-time constructible) arithmetic circuits. But PosSLP is precisely this problem, so this yields a P-time many-one reduction from both these problems to PosSLP. □

Chapter 4

Branching Markov Decision Processes

In this chapter we extend the results of Chapter 3 to provide the first polynomial time algorithms for approximating the LFP of both maxPPSs and minPPSs, and thus the first polynomial time algorithm for computing (to within any desired additive error) the optimal value vector for BMDPs with the objective of maximizing or minimizing their extinction probability.

Unlike PPSs and MPSs, the min/maxPPSs that define the Bellman equations for BMDPs are no longer differentiable functions (they are only piecewise differentiable). It is not even clear a priori how one could solve them using Newton's method.

We suggest using a *generalized Newton's method* (GNM), that extends Newton's method in a natural way to the setting of max/minPPSs, where each iteration requires the computation of the least (greatest) fixed point solution of a max- (min-) linear system of equations, both of which we show can be solved using linear programming. Just like in Chapter 3, our approach also makes crucial use of prior P-time algorithms (in this case, from [EY05]) for *qualitative* analysis of max/min BMDPs, which allow us to remove variables x_i where the LFP is $q_i^* = 1$ or where $q_i^* = 0$. The algorithms we provide have the nice feature that they are relatively simple, although the analysis of their correctness and time complexity is rather involved.

We furthermore show that we can compute ε-optimal (pure) strategies (policies) for both maxPPSs and minPPSs, for *any* given desired $\varepsilon > 0$, in time polynomial in both the encoding size of the max/minPPS and in $\log(1/\varepsilon)$. This result is at first glance rather surprising, because there are only a bounded number of distinct pure policies for a max/minPPS, and computing an optimal policy is PosSLP-hard.

Finally, we consider *Branching simple stochastic games* (BSSGs), which are two-player turn-based stochastic games, where one player wants to maximize, and the other wants to minimize, the extinction probability (see [EY05]). The *value* of these games (which are determined) is characterized by the LFP solution of associated min-maxPPSs which combine both min and max operators (see [EY05]). We observe that our results easily imply a FNP upper bound for ε-approximating the *value* of BSSGs and computing ε-optimal strategies for them.

Related work

We have already mentioned some of the important related results. BMDPs and related processes have been studied previously in both the operations research (e.g. [Pli76, RW82, DR05]) and computer science literature (e.g. [EY05, EGKS08, BBFK06]), but no efficient algorithms were known for the (approximate) computation of the relevant optimal probabilities and policies; the best known upper bound was PSPACE [EY05].

In [EY05], Etessami and Yannakakis introduced Recursive Markov Decision Processes (RMDPs), a recursive extension of MDPs. They showed that for general RMDPs, the problem of computing the optimal termination probabilities, even within any non-trivial approximation, is undecidable. However, they showed for the important class of 1-exit RMDPs (1-RMDP), the optimal probabilities can be expressed by min (or max) PPSs, and in fact the problems of computing (approximately) the LFP of a min/maxPPS and the termination probabilities of a max/min 1-RMDP, or BMDP, are all polynomially equivalent. They furthermore showed in [EY05] that there are always pure, memoryless optimal policies for both maximizing and minimizing 1-RMDPs (and for the more general turn-based stochastic games).

In [EWY08], 1-RMDPs with a different objective were studied, namely optimizing the total expected reward in a setting with positive rewards. In that setting, things are much simpler: the Bellman equations turn out to be max/min-linear, the optimal values are rational, and they can be computed *exactly* in P-time using linear programming.

A work that is more closely related to the results in this chapter is [EGKS08] by Esparza, Gawlitza, Kiefer, and Seidl. They studied more general monotone min-maxMPSs, i.e., systems of monotone polynomial equations that include both min and max operators, and they presented two different iterative analogs of Newton's methods for approximating the LFP of a min-maxMPS, $x = P(x)$. Their methods are related to ours, but differ in key respects. Both of their methods use certain piece-wise linear functions to approximate the min-maxMPS in each iteration, which is also what one does to solve each iteration of our generalized Newton's method. However, the precise nature of their piece-wise linearizations, as well as how they solve them, differ in important ways from ours, even when they are applied in the specific context of maxPPSs or minPPSs. They show, working in the unit-cost *exact* arithmetic model, that using their methods one can compute j "valid bits" of the LFP (i.e., compute the LFP within relative error at most 2^{-j}) in $k_P + c_P \cdot j$ iterations, where k_P and c_P are terms that depend in *some* way on the input system, $x = P(x)$. However, they give no constructive upper bounds on k_P, and their upper bounds on c_P are exponential in the number n of variables of $x = P(x)$. Note that MPSs are more difficult: even without the min and max operators, we know that it is PosSLP-hard to approximate their LFP within any nontrivial constant additive error $c < 1/2$, even for pure MPSs that arise from Recursive Markov Chains [EY09].

Another subclass of RMDPs, called *one-counter MDPs* (a controlled extension of one-counter Markov chains and Quasi-Birth-Death processes [EWY10]) has been studied, and the approximation of their optimal termination probabilities was recently shown to be computable, but only in *exponential time* ([BBEK11]). This subclass is incomparable with 1-RMDPs and BMDPs, and does not have min/maxPPSs as Bellman equations.

4.1 Generalizing Newton's method using LP

If we knew an optimal policy τ for a max/minPPS, $x = P(x)$, then we would be able to solve the problem of computing the LFP for a max/minPPS by using the algorithm from Chapter 3 to approximate the LFP of the PPS $x = P_\tau(x)$, q^*_τ, because we know $q^*_\tau = q^*$. Unfortunately, we do not know which policy is optimal. In fact, as we will see, it is probably impossible to identify an optimal policy in polynomial time. There are exponentially many policies, so it would be inefficient to run this algorithm using every policy.

To prove the polynomial-time upper bound in Chapter 3, an inductive step of the following form was used:

Lemma 4.1 (Combining Lemma 3.9 and Theorem 3.14)**.** *Let $x = P(x)$ be a PPS in SNF with $\mathbf{0} < q^* < \mathbf{1}$. For any $0 \leq x \leq q^*$ and $\lambda > 0$, the operator $\mathcal{N}(x)$ is defined, $\mathcal{N}(x) \leq q^*$, and if $q^* - x \leq \lambda(\mathbf{1} - q^*)$ then $q^* - \mathcal{N}(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$.*

Our goal will be to find an iteration operator, $I(x)$, for max/minPPSs, that has similar properties to the Newton operator for PPSs, i.e., that can be computed efficiently for a given x and for which we can prove a property similar to Lemma 4.1, i.e., such that if $q^* - x \leq \lambda(\mathbf{1} - q^*)$, then $q^* - I(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$. Once we do so, we will be able to adapt and extend results from Chapter 3 to get a polynomial time algorithm for the problem of approximating the LFP q^* of a max/minPPS.

If a max/minPPS, $x = P(x)$, has no equations of form $\mathtt{Q}$, then it amounts to precisely the Bellman equations for an ordinary finite-state Markov Decision Process with the objective of maximizing/minimizing reachability probabilities. It is well known that we can compute the exact (rational) optimal values for such finite-state MDPs, and thus the exact LFP, q^*, for such a max(min)-linear systems, using linear programming (see, e.g., [Put94, CY98]).

Computing the LFP of max/minPPSs is clearly a generalization of this finite-state MDP problem to the infinite-state setting of branching and recursive MDPs. If we have no equations of form $\mathtt{M}$, we have a PPS, which we can solve in P-time using Newton's method, as shown in Chapter 3. An iteration of Newton's method works by approximating the system of equations by a linear system. For a maxPPS(or minPPS), we will define an analogous "approximate" system of equations that we have to solve in each iteration of **"Generalized Newton's Method"** (GNM) which has both linear equations and equations involving the max (or min) function. We will show that we can solve the equations that arise from each iteration of GNM using linear programming. We will then show that a polynomial (in fact, linear) number of iterations are enough to approximate the desired LFP solution, and that it suffices to carry out the computations with polynomial precision.

The rest of this section is organized as follows. In 4.1.1 we define a linearization of a max/minPPS and prove some basic properties. In 4.1.2 we define the operator for an iteration of the Generalized Newton's method and show that it can be computed by Linear Programming. In 4.1.3 we analyze the operator for maxPPS and in Section 4.1.4 for minPPS. Finally in 4.1.5 we put everything together and show that the algorithm approximates the LFP within any desired precision in polynomial time in the Turing model.

4.1.1 Linearizations of max/minPPSs and their properties

We begin by expressing the max/min linear equations that should be solved by one iteration of what will eventually become the "Generalized Newton's Method" (GNM), applied at a point y. Recall that we assume w.l.o.g. throughout that max/minPPS and

PPS are in SNF.

Definition 4.2. *For a max/minPPS, $x = P(x)$, with n variables, the* **linearization of** $P(x)$ **at a point y** $\in \mathbb{R}^n$*, is a system of max/min linear functions denoted by $P^y(x)$, which has the following form:*

if $P_i(x)$ has form `L` *or* `M`*, then $P_i^y(x) = P_i(x)$, and*

if $P_i(x)$ has form `Q`*, i.e., $P_i(x) = x_j x_k$ for some j,k, then*

$$P_i^y(x) = y_j x_k + x_j y_k - y_j y_k$$

We can consider the linearization of a PPS, $x = P_\sigma(x)$, obtained as the result of fixing a policy, σ, for a max/minPPS, $x = P(x)$.

Definition 4.3. $P_\sigma^y(x) := (P_\sigma)^y(x)$.

Note than the linearization $P^y(x)$ only changes equations of form `Q`, and using a policy σ only changes equations of form `M`, so these operations are independent in terms of the effects they have on the underlying equations, and thus $P_\sigma^y(x) \equiv (P_\sigma)^y(x) = (P^y)_\sigma(x)$.

Lemma 4.4. *Let $x = P(x)$ be any PPS. For any $y \in \mathbb{R}^n$, let $(P^y)'(x)$ denote the Jacobian matrix of $P^y(x)$. Then for any $x \in \mathbb{R}^n$, we have $(P^y)'(x) = B(y)$.*

Proof. We need to show that the Jacobian $(P^y)'(x)$ of $P^y(x)$, evaluated anywhere, is equal to $B(y)$. If $x_i = P_i(x)$ is not of form `Q`, then, for any $x \in \mathbb{R}^n$, $P_i(x) = P_i^y(x)$. So for any x_j, $\frac{\partial P_i^y(x)}{\partial x_j} = \frac{\partial P_i(x)}{\partial x_j}$. Otherwise, $x_i = P_i(x)$ has form `Q`, that is $P_i(x) = x_j x_k$ for some variables x_j,x_k. Then $P_i^y(x) = y_j x_k + x_j y_k - y_j y_k$. In this case $\frac{\partial P_i^y(x)}{\partial x_j} = y_k$ and $\frac{\partial P_i^y(x)}{\partial x_k} = y_j$. But when $x = y$, $\frac{\partial P_i(x)}{\partial x_j} = y_k$ and $\frac{\partial P_i(x)}{\partial x_k} = y_j$. Furthermore, clearly for any x_l, with $l \neq j$ and $l \neq k$, $\frac{\partial P_i(x)}{\partial x_l} = 0$ and $\frac{\partial P_i^y(x)}{\partial x_l} = 0$. We have thus established that $(P^y)'(x) = B(y)$ for any $x \in \mathbb{R}^n$. □

Lemma 4.5. *If $x = P(x)$ is any PPS, then for any $x, y \in \mathbb{R}^n$, $P^y(x) = P(y) + B(y)(x - y)$.*

Proof. Firstly, note that $P^y(x) = P^y(y) + (P^y)'(x)(x - y)$, since the functions $P_i^y(x)$ are all linear in x. Next, observe that $P_i(y) = P_i^y(y)$, for all i, and thus that $P(y) = P^y(y)$. Thus, to show that $P^y(x) = P^y(y) + B(y)(x - y) = P(y) + B(y)(x - y)$, all we need to show is that the Jacobian $(P^y)'(x)$ of $P^y(x)$, evaluated anywhere, is equal to $B(y)$. But this was established in Lemma 4.4. □

An iteration of Newton's method on $x = P_\sigma(x)$ at a point y solves a system of linear equations that can be expressed in terms of $P^y_\sigma(x)$. The next lemma establishes this basic fact in part *(i)*. In part *(ii)* it provides us with conditions under which we are guaranteed to be doing "at least as well" as one such Newton iteration.

Lemma 4.6. *Suppose that the matrix inverse $(I - B_\sigma(y))^{-1}$ exists and is non-negative, for some policy σ, and some $y \in \mathbb{R}^n$. Then*

(i) $\mathcal{N}_\sigma(y) \equiv y + (I - B_\sigma(y))^{-1}(P_\sigma(y) - y)$ *is defined, and is equal to the unique point $a \in \mathbb{R}^n$ such that $P^y_\sigma(a) = a$.*

(ii) *For any vector $x \in \mathbb{R}^n$:*
If $P^y_\sigma(x) \geq x$, then $x \leq \mathcal{N}_\sigma(y)$.
If $P^y_\sigma(x) \leq x$, then $x \geq \mathcal{N}_\sigma(y)$.

Proof. (i): We define:

$$a = y + (I - B_\sigma(y))^{-1}(P_\sigma(y) - y) \equiv \mathcal{N}_\sigma(y)$$

Then we can re-arrange this expression, reversibly, yielding:

$$\begin{aligned} a = y + (I - B_\sigma(y))^{-1}(P_\sigma(y) - y) \quad &\Leftrightarrow \quad P_\sigma(y) - y - (I - B_\sigma(y))(a - y) = 0 \\ &\Leftrightarrow \quad P_\sigma(y) + B_\sigma(y)(a - y) = a \\ &\Leftrightarrow \quad P^y_\sigma(a) = a \qquad \text{(by Lemma 4.5)} \end{aligned}$$

Uniqueness follows from the reversibility of these transformations.

(ii): Firstly, we shall observe that the result of applying Newton's method to solve $x = P^y_\sigma(x)$ with any initial point x gives us $\mathcal{N}_\sigma(y) = a$ in a single iteration. Recalling from Lemma 4.4 that the following equality holds between the Jacobians: $(P^y)'(x) = B_\sigma(y)$, one iteration of Newton's method applied to $x = P^y_\sigma(x)$ can be equivalently defined as:

$$\begin{aligned} x + (I - B_\sigma(y))^{-1}(P^y_\sigma(x) - x) &= x + (I - B_\sigma(y))^{-1}(P_\sigma(y) + B_\sigma(y)(x - y) - x) \\ &= (I - B_\sigma(y))^{-1}(x - B_\sigma(y)x + P_\sigma(y) + B_\sigma(y)(x - y) - x) \\ &= (I - B_\sigma(y))^{-1}(P_\sigma(y) - B_\sigma(y)y) \\ &= (I - B_\sigma(y))^{-1}((I - B_\sigma(y))y + P_\sigma(y) - y) \\ &= y + (I - B_\sigma(y))^{-1}(P_\sigma(y) - y) \\ &= \mathcal{N}_\sigma(y). \end{aligned}$$

We thus have $\mathcal{N}_\sigma(y) = x + (I - B_\sigma(y))^{-1}(P^y_\sigma(x) - x)$. By assumption, $(I - B_\sigma(y))^{-1}$ is a non-negative matrix. So if $P^y_\sigma(x) - x \geq 0$ then $\mathcal{N}_\sigma(y) \geq x$, whereas if $P^y_\sigma(x) - x \leq 0$ then $\mathcal{N}_\sigma(y) \leq x$. □

4.1.2 The iteration operator of Generalized Newton's Method

We shall now define distinct iteration operators for a maxPPS and a minPPS, both of which we shall refer to with the overloaded notation $I(x)$. (We shall also establish in the next two subsections that the operators are well-defined in their respective settings.) These operators will serve as the basis for a *Generalized Newton's Method* to be applied to maxPPSs and minPPSs, respectively.

Definition 4.7. *For a maxPPS, $x = P(x)$, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, and for a real vector y such that $0 \leq y \leq q^*$, we define the operator $I(y)$ to be the* unique *optimal solution, $a \in \mathbb{R}^n$, to the following mathematical program:*

$$\text{Minimize: } \sum_i a_i \ ; \quad \text{Subject to: } \quad P^y(a) \leq a$$

For a minPPS, $x = P(x)$, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, and for a real vector y such that $0 \leq y \leq q^*$, we define the operator $I(y)$ to be the unique *optimal solution $a \in \mathbb{R}^n$ to the following mathematical program:*

$$\text{Maximize: } \sum_i a_i \ ; \quad \text{Subject to: } \quad P^y(a) \geq a$$

A priori, it is not even clear if the above "definitions" of $I(x)$ for maxPPSs and minPPSs are well-defined. We now make the following central claim, which we shall prove separately for maxPPSs and minPPSs in the following two subsections:

Proposition 4.8. *Let $x = P(x)$ be a max/minPPS, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$. For any $0 \leq x \leq q^*$:*

1. *$I(x)$ is well-defined, and $I(x) \leq q^*$, and:*
2. *For any $\lambda > 0$, if $q^* - x \leq \lambda(\mathbf{1} - q^*)$ then $q^* - I(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$.*

The next proposition observes that linear programming can be used to compute an iteration of the operator, $I(x)$, for both maxPPSs and minPPSs.

Proposition 4.9. *Given a max/minPPS, $x = P(x)$, with LFP q^*, and given a rational vector y, $0 \leq y \leq q^*$, the constrained optimization problem (i.e., mathematical program) "defining" $I(y)$ can be described by a LP whose encoding size is polynomial (in* *fact, linear) in both $|P|$ and the encoding size of the rational vector y. Thus, we can compute the (unique) optimal solution $I(y)$ to such an LP (assuming it exists, and is unique) in P-time.*

Proof. For a maxPPS (minPPS), the definition of $I(x)$ asks us to maximize (minimize) a linear objective, $\sum_i a_i$, subject to the constraints $P^y(a) \leq a$ ($P^y(a) \geq a$, respectively). All of these constraints are linear, except the constraints of form M. For a maxPPS, if $(P^y(a))_i$ is of form M, then the corresponding constraint is an inequality of the form $\max\{a_j, a_k\} \leq a_i$. Such an inequality is equivalent to, and can be replaced by, the two linear inequalities: $a_j \leq a_i$ and $a_k \leq a_i$. Likewise, for a minPPS, if $(P^y(a))_i$ is of form M, then the corresponding constraint is an inequality of the form $\min\{a_j, a_k\} \geq a_i$. Again, such an inequality is equivalent to, and can be replaced by, two linear inequalities: $a_j \geq a_i$ and $a_k \geq a_i$.

Thus, for a rational vector y whose encoding length is $\texttt{size}(y)$, the operator $I(y)$ can be formulated (for both maxPPSs and minPPSs) as a problem of computing the unique optimal solution to a linear program whose encoding size is polynomial (in fact, linear) in $|P|$ and in $\texttt{size}(y)$. □

4.1.3 An iteration of Generalized Newton's Method (GNM) for maxPPSs

For a maxPPS, $x = P(x)$, we know by Theorem 2.6 that there exists an optimal policy, τ, such that $q^* = q^*_\tau \geq q^*_\sigma$ for any policy σ. The next lemma implies part (i) of Proposition 4.8 for maxPPS:

Lemma 4.10. *If $x = P(x)$ is a maxPPS, with LFP solution $\mathbf{0} < q^* < \mathbf{1}$, and y is a real vector with $0 \leq y \leq q^*$, then $x = P^y(x)$ has a least fixed point solution, denoted μP^y, with $\mu P^y \leq q^*$. Furthermore, the operator $I(y)$ is well-defined, $I(y) = \mu P^y \leq q^*$, and for any optimal policy τ, $I(y) = \mu P^y \geq \mathcal{N}_\tau(y)$.*

Proof. Recall that (by Proposition 4.9) the following can be written as an LP that "defines" $I(y)$:

$$\textit{Minimize:} \sum_i a_i \ ; \qquad \textit{Subject to:} \quad P^y(a) \leq a \tag{4.1}$$

Firstly, we show that the LP constraints $P^y(a) \leq a$ in the definition of $I(y)$ are *feasible*. We do so by showing that actually $P^y(q^*) \leq q^*$. At any coordinate i, if $P_i(x)$ has form M or L, then $P^y_i(q^*) = P_i(q^*) = q^*_i$. Otherwise, $P_i(x)$ has form Q, i.e.,

$P_i(x) = x_j x_k$, and then

$$\begin{aligned} P_i^y(q^*) &= q_j^* y_k + y_j q_k^* - y_j y_k \\ &= q_j^* q_k^* - (q_j^* - y_j)(q_k^* - y_k) \\ &\leq q_i^* \qquad\qquad \text{(since } y \leq q^*\text{)} \end{aligned}$$

Next we show that the LP (4.1) defining $I(y)$ is *bounded.* Recall that, by Theorem 2.6, there is always an optimal policy for any maxPPS, $x = P(x)$.

Claim 4.11. *Let $x = P(x)$ be any maxPPS, with $\mathbf{0} < q^* < \mathbf{1}$, and let τ be any optimal policy for $x = P(x)$. For any y such that $0 \leq y \leq q^*$, we have that $\mathcal{N}_\tau(y)$ is defined, and for any vector a, if $P^y(a) \leq a$ then $\mathcal{N}_\tau(y) \leq a$. In particular, $\mathcal{N}_\tau(y) \leq q^*$.*

Proof. Recall, from our definition of an optimal policy, that $q^* = q_\tau^*$ is also the least non-negative solution to $x = P_\tau(x)$. So we can apply Theorem 3.6 using $x = P_\tau(x)$ and $y \leq q^*$ to deduce that $(I - B_\tau(y))^{-1}$ exists and is non-negative. Thus $\mathcal{N}_\tau(y)$ is defined. Now, by applying Lemma 4.6 (ii), to show that $a \geq \mathcal{N}_\tau(y)$ all we need to show is that $P_\tau^y(a) \leq a$. But recalling that $x = P(x)$ is a maxPPS, by the definition of $P^y(x)$ and $P_\tau^y(x)$, we have that $P_\tau^y(a) \leq P^y(a) \leq a$. We have just shown before this Claim that $P^y(q^*) \leq q^*$, and thus $\mathcal{N}_\tau(y) \leq q^*$. □

Thus the LP (4.1) defining $I(y)$ is both feasible and bounded, hence it has an optimal solution. To show that $I(y)$ is well-defined, all that remains is to show that this optimal solution is unique. In the process, we will also show that $I(y)$ defines precisely the *least fixed point* solution of $x = P^y(x)$, which we denote by μP^y.

Firstly, we claim that for any optimal solution b to the LP (4.1), it must be the case that $P^y(b) = b$. Suppose not. Then there exists i such that $P^y(b)_i < b_i$, then we can define a new vector b', such that $b'_i = P^y(b)_i$ and $b'_j = b_j$ for all $j \neq i$. By monotonicity of $P^y(x)$, it is clear that $P^y(b') \leq b'$, and thus that b' is a feasible solution to the LP (4.1). But $\sum_i b'_i < \sum_i b_i$, contradicting the assumption that b is an optimal solution to the LP (4.1).

Secondly, we claim that there is a unique optimal solution. Suppose not: suppose b and c are two distinct optimal solution to the LP (4.1). Define a new vector d by $d_i = \min\{b_i, c_i\}$, for all i. Clearly, $d \leq b$ and $d \leq c$. Thus by the monotonicity of $P^y(x)$, for all i $P^y(d)_i \leq P^y(b)_i = b_i$, and likewise $P^y(d)_i \leq P^y(c)_i = c_i$. Thus $P^y(d) \leq d$, and d is a feasible solution to the LP. But since b and c are distinct, and yet $\sum_i b_i = \sum_i c_i$, we have that $\sum_i d_i < \sum_i b_i = \sum_i c_i$, contradicting the optimality of both b and c.

We have thus established that $I(y)$ defines the unique *least fixed point* solution of $x = P^y(x)$, which we denote also by μP^y. Since q^* is also a solution of the LP, we have $\mu P^y \leq q^*$.

Finally, by Claim 4.11, it must be the case that $I(y) = \mu P^y \geq \mathcal{N}_\tau(y)$, where τ is any optimal policy for $x = P(x)$. □

We next establish part (ii) of Proposition 4.8 for maxPPS.

Lemma 4.12. *Let $x = P(x)$ be a maxPPS with $\mathbf{0} < q^* < \mathbf{1}$. For any $0 \leq x \leq q^*$ and $\lambda > 0$, we have $I(x) \leq q^*$, and furthermore if:*

$$q^* - x \leq \lambda(\mathbf{1} - q^*)$$

then

$$q^* - I(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$$

Proof. Let τ be an optimal policy (which exists by Theorem 2.6). The least fixed point solution of the PPS $x = P_\tau(x)$ is q^*. From our assumptions, Lemma 4.1 gives that $q^* - \mathcal{N}_\tau(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$. But by Lemma 4.10 $\mathcal{N}_\tau(x) \leq I(x) \leq q^*$. The claim follows. □

Proposition 4.8 for maxPPSs follows from Lemmas 4.10 and 4.12. In subsection 4.1.5 we will combine this result with methods from Chapter 3 to obtain a P-time algorithm for approximating the LFP of a maxPPS, in the standard Turing model of computation.

4.1.4 An iteration of GNM for minPPSs

Our proof of the minPPS version of Lemma 4.12 will be somewhat different, because it turns out we can not use the same argument based on LPs to prove that $I(y)$ is well-defined. Fortunately, in the case of minPPSs, we can show that $(I - B_\sigma(y))^{-1}$ exists and is non-negative for *any* policies σ, at those points y that are of interest. And we can use this to show that there is *some* policy, σ, such that $I(y)$ is equivalent to an iteration of Newton's method at y after fixing the policy σ. We shall establish the existence of such a policy using a policy improvement argument, instead of just using the LP, as we did for maxPPSs. (Note that the policy improvement algorithm may not be an efficient (P-time) way to compute it, and we do not claim it is. We only use policy improvement as an argument in the proof of existence of a suitable policy σ.)

Lemma 4.13. *For a minPPS, $x = P(x)$, and for any policy σ, the LFP of, $x = P_\sigma(x)$, denoted q^*_σ, satisfies $q^* \leq q^*_\sigma$.*

Proof. By Theorem 2.6, there is an optimal policy τ with $q^*_\tau = q^*$. But we defined an optimal policy for a minPPS as one with $q^*_\tau \leq q^*_\upsilon$ for any policies υ. So $q^* = q^*_\tau \leq q^*_\sigma$. □

Lemma 4.13 allows us to use Theorem 3.6 with any policy, not just with optimal policies:

Lemma 4.14. *For a minPPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < 1$, for any $0 \leq y \leq q^*$ and any policy σ, $(I - B_\sigma(y))^{-1}$ exists and is non-negative. Thus also $\mathcal{N}_\sigma(y)$ is defined.*

To prove this, we need a slight extension of Theorem 3.6. Recall that for a square matrix A, $\rho(A)$ denotes its spectral radius.

Lemma 4.15. *Given a PPS, $x = P(x)$, with LFP $q^* > 0$, if $0 \leq y \leq q^*$, and $y < 1$, then $(I - B(y))^{-1}$ exists and is non-negative.*

Recall that a PPS, $x = P(x)$, is called *strongly connected*, if its variable dependency graph H is strongly connected.

Lemma 4.16. *(Lemma 6.5 of [EY09])*[1] *Let $x = P(x)$ be a strongly connected PPS, in n variables, with LFP $q^* > \mathbf{0}$. For any vector $0 \leq y < q^*$, $\rho(B(y)) < 1$, and thus $(I - B(y))^{-1}$ exists and is nonnegative.*

Proof of Lemma 4.15. Consider a PPS, $x = P(x)$, with LFP $q^* > \mathbf{0}$, and a vector $0 \leq y \leq q^*$, such that $y < 1$. Note that all we need to establish is that $\rho(B(y)) < 1$, because it then follows by standard facts (see, e.g., [HJ85]) that $(I - B(y))^{-1}$ exists and is equal to $\sum_{i=0}^{\infty}(B(y))^i \geq 0$.

Let us first show that if $x = P(x)$ is strongly connected, then $\rho(B(y)) < 1$. To see this, note that if $x = P(x)$ is strongly connected, then every variable depends on every other, and thus if there exists any $i \in \{1, \ldots, n\}$ such that $q^*_i < 1$, then it must be the case that for all $j \in \{1, \ldots, n\}$, we have $q^*_j < 1$. Thus, either $q^* = 1$, or else $0 < q^* < \mathbf{1}$. If $q^* = 1$, then since $y < 1$, we have $y < q^*$, and thus, by Lemma 4.16, we have $\rho(B(y)) < 1$. If, on the other hand, $\mathbf{0} < q^* < \mathbf{1}$, then since $0 \leq y \leq q^*$, by Theorem 3.6, we have $\rho(B(y)) < 1$.

[1] Lemma 6.5 of [EY09] is actually a more general result, relating to strongly connected MPSs that arise from more general Recursive Markov Chains.

Next, consider an arbitrary PPS, $x = P(x)$, that is not necessarily strongly connected. Recall the variable dependency graph H of $x = P(x)$. We can partition the variables into sets $S_1,\dots,S_k$ which form the SCCs of H. Consider the DAG, D, of SCCs, whose nodes are the sets S_i, and for which there is an edge from S_i to S_j if and only if in the dependency graph H there is a node $i' \in S_i$ with an edge to a node in $j' \in S_j$.

Consider the matrix $B(y)$. Our aim is to show that $\rho(B(y)) < 1$. Since we assume $q^* > \mathbf{0}$, $0 \leq y \leq q^*$, and $y < 1$, it clearly suffices to show that $\rho(B(y)) < 1$ holds in the case where we additionally insist that $y > 0$, because then for any other z such that $0 \leq z \leq y$, we would have $\rho(B(z)) \leq \rho(B(y)) < 1$.

So, assuming also that $y > 0$, consider the $n \times n$-matrix $B(y)$. To keep notation clean, we let $A := B(y))$. For the $n \times n$ matrix A, we can consider its underlying *dependency* graph, $H = (\{1,\dots,n\}, E_H)$, whose nodes are $\{1,\dots,n\}$, and where there is an edge from i to j if and only if $A_{i,j} > 0$. Notice however that, since $y > 0$, this graph is precisely the same graph as the dependency graph H of $x = P(x)$, and thus it has the same SCCs, and the same DAG of SCCs, D. Let us sort the SCCs, so that we can assume $S_1,\dots,S_k$ are topologically sorted with respect to the partial ordering defined by the DAG D. In other words, for any variable indices $i \in S_a$ and $j' \in S_b$ if $(i, j) \in E_H$, then $a \leq b$.

Let $S \subseteq \{1,\dots,n\}$ be any non-empty subset of indices, and let $A[S]$ denote the principal submatrix of A defined by indices in S. It is a well known fact that $0 \leq \rho(A[S]) \leq \rho(A)$. (See, e.g, Corollary 8.1.20 of [HJ85].)

Since $A \geq 0$, $\rho(A)$ is an eigenvalue of A, and has an associated non-negative eigenvector $v \geq 0$, $v \neq 0$ (again see, e.g., Chapter 8 of [HJ85]). In other words,

$$Av = \rho(A)v$$

Firstly, if $\rho(A) = 0$, then we are of course trivially done. So we can assume w.l.o.g. that $\rho(A) > 0$. Now, if $v_i > 0$, then for every j such that $(j,i) \in E_H$, we have $(Av)_j > 0$, and thus since $(Av)_j = \rho(A)v_j$, we have $v_j > 0$. Hence, repeating this argument, if $v_i > 0$ then for every j that has a path to i in the dependency graph H, we have $v_j > 0$.

Since $v \neq 0$, it must be the case that there is exists some SCC, S_c, of H such that for every variable index $i \in S_c$, $v_i > 0$, and furthermore, such that c is the maximum index for such an SCC in the topologically sorted list $S_1,\dots,S_k$, i.e., such that for all $d > c$, and for all $j \in S_d$, we have $v_j = 0$.

First, let us note that it must be the case that S_c is a *non-trivial* SCC. Specifically, let us call an SCC, S_r of H *trivial* if $S_r = \{i\}$ consists of only a single variable index, i, and furthermore, such that $\mathbf{0} = (A)_i = (B(y))_i$, i.e., that row i of the matrix A is all zero. This can not be the case for S_c, because for any variable $i \in S_c$, we have $v_i > 0$, and thus $(Av)_i = \rho(A)v_i > 0$.

Let us consider the principal submatrix $A[S_c]$ of A. We claim that $\rho(A[S_c]) = \rho(A)$. To see why this is the case, note that $Av = \rho(A)v$, and for every $i \in S_c$, we have $(Av)_i = \sum_j a_{i,j}v_j = \rho(A)v_i$. But $v_j = 0$ for every $j \in S_d$ such that $d > c$, and furthermore $a_{i,j} = 0$ for every $j \in S_{d'}$ such that $d' < c$.

Thus, if we let v_{S_c} denote the subvector of v corresponding to the indices in S_c, then we have just established that $A[S_c]v_{S_c} = \rho(A)v_{S_c}$, and thus that $\rho(A[S_c]) \geq \rho(A)$. But since $A[S_c]$ is a principal submatrix of A, we also know easily (see, e.g, Corollary 8.1.20 of [HJ85]), that $\rho(A[S_c]) \leq \rho(A)$, so $\rho(A[S_c]) = \rho(A)$.

We are almost done. Given the original PPS, $x = P(x)$, for any subset $S \subseteq \{1,\ldots,n\}$ of variable indices, let $x_S = P_S(x_S, x_{D_S})$ denote the subsystem of $x = P(x)$ associated with the vector x_S of variables in set S, where x_{D_S} denotes the variables not in S.

Now, note that $x_{S_c} = P_{S_c}(x_{S_c}, y_{D_{S_c}})$ is itself a PPS. Furthermore, it is a *strongly connected* PPS, precisely because S_c is a strongly connected component of the dependency graph H, and because $y > 0$. Moreover, the Jacobian matrix of $P_{S_c}(x_{S_c}, y_{D_{S_c}}))$, evaluated at y_{S_c}, which we denote by $B_{S_c}(y)$, is precisely the principal submatrix $A[S_c]$ of A. Since $x_{S_c} = P_{S_c}(x_{S_c}, y_{D_{S_c}})$ is a strongly connected PPS, we have already argued that it must be the case that $\rho(B_{S_c}(y)) < 1$. Thus since $B_{S_c}(y) = A[S_c]$, we have $\rho(A[S_c]) = \rho(A) < 1$. This completes the proof. □

Proof of Lemma 4.14. $0 \leq y \leq q^* \leq q^*_\sigma \leq 1$. Note also that $y < 1$, and that $q^*_\sigma \geq q^* > \mathbf{0}$. This is all we need for Lemma 4.15 to apply. □

Lemma 4.17. *Given a minPPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < 1$, and a vector y with $0 \leq y \leq q^*$, there is a policy σ such that $P^y(\mathcal{N}_\sigma(y)) = \mathcal{N}_\sigma(y)$.*

Proof. We use a policy (strategy) improvement "algorithm" to prove this. Start with any policy σ_1. At step i, suppose we have a policy σ_i.

For notational simplicity, in the following we use the abbreviation: $z = \mathcal{N}_{\sigma_i}(y)$. By Lemma 4.6, $P^y_{\sigma_i}(z) = z$. So we have $P^y(z) \leq z$. If $P^y(z) = z$, then *stop*: we are done.

Otherwise, to construct the next strategy σ_{i+1}, take the smallest j such that $(P^y(z))_j < z_j$. Note that $P_j(x)$ has form `M`, because otherwise $(P(x))_j = (P_{\sigma_i}(x))_j$. Thus, there is some variable x_k with $P_j(x) = \min\{x_k, x_{\sigma_i(j)}\}$ and $z_k < z_{\sigma_i(j)}$. Define σ_{i+1} to be:

$$\sigma_{i+1}(l) = \begin{cases} \sigma_i(l) & \text{if } l \neq j \\ k & \text{if } l = j \end{cases}$$

Then $(P^y_{\sigma_{i+1}}(z))_j < z_j$, but for every other coordinate $l \neq j$, $(P^y_{\sigma_{i+1}}(z))_l = (P^y_{\sigma_i}(z))_l = z_l$. Thus

$$P^y_{\sigma_{i+1}}(z) \leq z \tag{4.2}$$

By Lemma 4.14, $\mathcal{N}_{\sigma_{i+1}}(y)$ is defined. Moreover, the inequality (4.2), together with Lemma 4.6 (ii), yields that $\mathcal{N}_{\sigma_{i+1}}(y) \leq z$. But $\mathcal{N}_{\sigma_{i+1}}(y) \neq z$ because $P^y_{\sigma_{i+1}}(z) \neq z$ whereas, by Lemma 4.6 (i), we have $P^y_{\sigma_{i+1}}(\mathcal{N}_{\sigma_{i+1}}(y)) = \mathcal{N}_{\sigma_{i+1}}(y)$.

Thus this algorithm gives us a sequence of policies $\sigma_1, \sigma_2, \ldots$ with $\mathcal{N}_{\sigma_1}(y) \geq \mathcal{N}_{\sigma_2}(y) \geq \mathcal{N}_{\sigma_3}(y) \geq \ldots$, where furthermore each step must strictly decrease at least one coordinate of $\mathcal{N}_{\sigma_i}(y)$. It follows that $\sigma_i \neq \sigma_j$, unless $i = j$. There are only finitely many policies. So the sequence must be finite, and the algorithm terminates. But it only terminates when we reach a σ_i with $P^y(\mathcal{N}_{\sigma_i}(y)) = \mathcal{N}_{\sigma_i}(y)$. □

We note that the analogous policy improvement algorithm might fail to work for maxPPSs, as we might reach a policy σ_i where $(I - P_{\sigma_i}(x))^{-1}$ does not exist, or has a negative entry.

The next Lemma shows that this policy improvement algorithm always produces a coordinate-wise minimal Newton iterate over all policies.

Lemma 4.18. *For a minPPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < \mathbf{1}$, if $0 \leq y \leq q^*$ and σ is a policy such that $P^y(\mathcal{N}_\sigma(y)) = \mathcal{N}_\sigma(y)$, then:*

(i) For any policy σ', $\mathcal{N}_{\sigma'}(y) \geq \mathcal{N}_\sigma(y)$.

(ii) For any $x \in \mathbb{R}^n$ with $P^y(x) \geq x$, we have $x \leq \mathcal{N}_\sigma(y)$.

(iii) For any $x \in \mathbb{R}^n$ with $P^y(x) \leq x$, we have $x \geq \mathcal{N}_\sigma(y)$.

(iv) $\mathcal{N}_\sigma(y)$ is the unique fixed point of $x = P^y(x)$.

(v) $\mathcal{N}_\sigma(y) \leq q^*$.

Proof. Note firstly that by Lemma 4.14, for any policy σ, $(I - B_\sigma(y))^{-1}$ exists and is non-negative, and $\mathcal{N}_\sigma(y)$ is defined.

(i) Consider $P^y_{\sigma'}(\mathcal{N}_\sigma(y))$. Note that $P^y_{\sigma'}(\mathcal{N}_\sigma(y)) \geq P^y(\mathcal{N}_\sigma(y)) = \mathcal{N}_\sigma(y)$ by assumption. Thus, by Lemma 4.6 (ii), $\mathcal{N}_\sigma(y) \leq \mathcal{N}_{\sigma'}(y)$.

(ii) $P^y_\sigma(x) \geq P^y(x) \geq x$, so by Lemma 4.6 (ii), $x \leq \mathcal{N}_\sigma(y)$.

(iii) If $P^y(x) \leq x$, then there a policy σ' with $P^y_{\sigma'}(x) \leq x$, and by Lemma 4.6 (ii), $x \geq \mathcal{N}_{\sigma'}(y)$. So using part (i) of this Lemma, $x \geq \mathcal{N}_{\sigma'}(y) \geq \mathcal{N}_\sigma(y)$.

(iv) By assumption, $\mathcal{N}_\sigma(y)$ is a fixed point of $x = P^y(x)$. We just need uniqueness. If $P^y(q) = q$, then by parts (ii) and (iii) of this Lemma, $q \leq \mathcal{N}_\sigma(y)$ and $q \geq \mathcal{N}_\sigma(y)$, i.e., $q = \mathcal{N}_\sigma(y)$.

(v) Consider an optimal policy τ, for the minPPS, $x = P(x)$. Then $q^*_\tau = q^*$ and by part of this Lemma, $\mathcal{N}_\sigma(y) \leq \mathcal{N}_\tau(y)$. All we need to show is that $q^*_\tau \leq q^*_\tau$. The following gives us that:

Lemma 4.19. *Given a PPS, $x = P(x)$, with LFP $q^* > \mathbf{0}$, if $0 \leq y \leq q^*$, and if $(I - B(y))^{-1}$ exists and is non-negative (in which case clearly $\mathcal{N}(y)$ is defined), then $\mathcal{N}(y) \leq q^*$ holds.*[2]

Proof. In Lemma 3.4, it was established that when $(I - B(y))$ is non-singular, i.e., $(I - B(y))^{-1}$ is defined, and thus $\mathcal{N}(y)$ is defined, then

$$q^* - \mathcal{N}(y) = (I - B(y))^{-1} \frac{B(q^*) - B(y)}{2} (q^* - y) \tag{4.3}$$

Now, since all polynomials in $P(x)$ have non-negative coefficients, it follows that the Jacobian $B(x)$ is monotone in x, and thus since $y \leq q^*$, we have that $B(q^*) \geq P'(y)$. Thus $(B(q^*) - B(y)) \geq 0$, and by assumption $(q^* - y) \geq 0$. Thus, by the assumption that $(I - B(y))^{-1} \geq 0$, we have by equation (4.3) that $q^* - \mathcal{N}(y) \geq 0$, i.e., that $q^* \geq \mathcal{N}(y)$. □

□

We can now return to using linear programming, which we can do in polynomial time. Recall the LP that "defines" $I(y)$, for a minPPS:

$$\textit{Maximize: } \sum_i a_i \; ; \qquad \textit{Subject to: } \quad P^y(a) \geq a \tag{4.4}$$

Lemma 4.20. *For a minPPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < \mathbf{1}$, and for $0 \leq y \leq q^*$, there is a unique optimal solution, which we call $I(y)$, to the LP (4.4), and furthermore $I(y) = \mathcal{N}_\sigma(y)$ for some policy σ, and $P^y(I(y)) = I(y)$.*

[2] Note that the Lemma does not claim that $\mathcal{N}(y) \geq 0$ holds. Indeed, it may not.

Proof. By Lemma 4.17, there is a σ such that $P^y(\mathcal{N}_\sigma(y)) = \mathcal{N}_\sigma(y)$. So $\mathcal{N}_\sigma(y)$ is a feasible solution of $P^y(a) \geq a$. Let a by any solution of $P^y(a) \geq a$. By Lemma 4.18 (ii), $a \leq \mathcal{N}_\sigma(y)$. Consequently $\sum_{i=1}^n a_i \leq \sum_{i=1}^n (\mathcal{N}_\sigma(y))_i$ with equality only if $a = \mathcal{N}_\sigma(y)$. So $\mathcal{N}_\sigma(y)$ is the unique optimal solution of the LP (4.4). □

In the maxPPS case, we had an iteration that was at least as good as iterating with the optimal policy. Here we have an iteration that is at least as bad! Nevertheless, we shall see that it is good enough. In the maxPPS case, the analog of Lemma 4.1, Lemma 4.12, thus followed from Lemma 4.1. Here we need a stronger result than Lemma 4.1.

Lemma 4.21. *If $x = P(x)$ is a PPS and we are given $x, y \in \mathbb{R}^n$ with $0 \leq x \leq y \leq P(y) \leq 1$, and if the following conditions hold:*

$$\lambda > 0 \quad \textit{and} \quad y - x \leq \lambda(\mathbf{1} - y) \quad \textit{and} \quad (I - B(x))^{-1} \textit{ exists and is non-negative,} \tag{4.5}$$

then $y - \mathcal{N}(x) \leq \frac{\lambda}{2}(\mathbf{1} - y)$.

(Note that we cannot conclude that $y - \mathcal{N}(x) \geq 0$.)

Proof. Firstly, we show that $B(y)(\mathbf{1} - y) \leq (\mathbf{1} - y)$. Clearly, for any PPS, $P(\mathbf{1}) \leq 1$. Note that since by assumption $y \leq P(y)$, we have $(\mathbf{1} - y) \geq (\mathbf{1} - P(y)) \geq (P(\mathbf{1}) - P(y))$. Then by Lemma 3.3 3.3:

$$\begin{aligned} (\mathbf{1} - y) \geq P(\mathbf{1}) - P(y) &= B(\frac{\mathbf{1} + y}{2})(\mathbf{1} - y) \qquad (4.6) \\ &\geq B(y)(\mathbf{1} - y) \qquad (4.7) \end{aligned}$$

Again by Lemma 3.3: $P(y) - P(x) = \frac{1}{2}(B(x) + B(y))(y - x)$, and thus:

$$P(x) = P(y) - \frac{1}{2}(B(x) + B(y))(y - x) \tag{4.8}$$

Thus:

$$
\begin{aligned}
y-\mathcal{N}(x) &= y-x-(I-B(x))^{-1}(P(x)-x)\\
&= y-x-(I-B(x))^{-1}(P(y)-x-\frac{1}{2}(B(x)+B(y))(y-x)) \quad \text{(by (4.8))}\\
&\leq y-x-(I-B(x))^{-1}(y-x-\frac{1}{2}(B(x)+B(y))(y-x))\\
&= (y-x)-(I-B(x))^{-1}((y-x)-\frac{1}{2}(B(x)+B(y))(y-x))\\
&= (I-(I-B(x))^{-1}(I-\frac{1}{2}(B(x)+B(y))))(y-x)\\
&= ((I-B(x))^{-1}(I-B(x))-(I-B(x))^{-1}(I-\frac{1}{2}(B(x)+B(y))))(y-x)\\
&= (I-B(x))^{-1}(I-B(x)-(I-\frac{1}{2}(B(x)+B(y))))(y-x)\\
&= (I-B(x))^{-1}(-B(x)+\frac{1}{2}(B(x)+B(y)))(y-x)\\
&= (I-B(x))^{-1}\frac{1}{2}(B(y)-B(x))(y-x)\\
&\leq \frac{\lambda}{2}(I-B(x))^{-1}(B(y)-B(x))(\mathbf{1}-y) \quad \text{(by (4.5), and because } (B(y)-B(x))\geq 0)\\
&\leq \frac{\lambda}{2}(I-B(x))^{-1}(I-B(x))(\mathbf{1}-y) \quad \text{(because by (4.7), } B(y)(1-y)\leq(1-y))\\
&= \frac{\lambda}{2}(\mathbf{1}-y)
\end{aligned}
$$

□

Lemma 4.22. *Let $x = P(x)$ be a minPPS, with LFP $\mathbf{0} < q^* < \mathbf{1}$. For any $0 \leq x \leq q^*$ and $\lambda > 0$, $I(x) \leq q^*$, and if:*

$$q^* - x \leq \lambda(\mathbf{1} - q^*)$$

then

$$q^* - I(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$$

Proof. By Lemma 4.17, there is a policy σ with $I(x) = \mathcal{N}_\sigma(x)$. We then apply Lemma 4.21 to $x = P_\sigma(x)$, x, and q^* instead of y. Observe that $P_\sigma(q^*) \geq P(q^*) = q^*$ and that $(I - B_\sigma(x))^{-1}$ exists and is non-negative. Thus the conditions of Lemma 4.21 hold, and we can conclude that $q^* - \mathcal{N}_\sigma(x) \leq \frac{\lambda}{2}(\mathbf{1} - q^*)$. Lastly, Lemma 4.18 (v) and Lemma 4.20 yield that $I(x) = \mathcal{N}_\sigma(x) \leq q^*$. □

Proposition 4.8 for minPPS follows from Lemmas 4.20 and 4.22.

4.1.5 A polynomial-time algorithm (in the Turing model) for max/minPPSs

In Chapter 3 we gave a polynomial time algorithm, in the standard Turing model of computation, for approximating the LFP of a PPS, $x = P(x)$, using Newton's method. Here we use the same methods, with our new *Generalized Newton's Method* (GNM), $I(x)$, to obtain polynomial-time algorithms (again, in the standard Turing model), for approximating the LFP of maxPPSs and minPPSs. The proof in Chapter 3 uses induction based on the "halving lemma", Lemma 4.1. We of course now have suitable "halving lemmas" for maxPPSs and minPPSs, namely, Lemmas 4.12 and 4.22. In Chapter 3, Theorem 3.14 was used for the base case of the induction. We can now easily derive an analogous Lemma for the setting of max/minPPSs:

Lemma 4.23. *If* $\mathbf{0} < q^* < \mathbf{1}$ *is the LFP of a max/minPPS,* $x = P(x)$*, in* n *variables, then for all* $i \in \{1,\dots,n\}$*:*

$$1 - q_i^* \geq 2^{-4|P|}$$

In other words, $0 < q_i^* \leq 1 - 2^{-4|P|}$*, for all* $i \in \{1,\dots,n\}$.

Proof. Let τ be any optimal policy for $x = P(x)$. We know it exists, by Theorem 2.6. Theorem 3.14 gives that $1 - q_i^* \geq 2^{-4|P_\tau|}$. All we need is to note is that $|P| \geq |P_\tau|$, which clearly holds using any sensible encoding for P and P_τ, in the sense that we should need no more bits needed to encode $x_i = x_j$ than to encode $x_i = \max\{x_j, x_k\}$ or $x_i = \min\{x_j, x_k\}$. □

Now we can give a polynomial time algorithm, in the Turing model of computation, for approximating the LFP, q^*, for a max/minPPS, to within any desired precision, by carrying out iterations of GNM using the same rounding technique, with the same rounding parameter, and using the same number of iterations, as in Chapter 3. Specifically, we use the following algorithm with rounding parameter h:

Start with $x^{(0)} := 0$;
For each $k \geq 0$ compute $x^{(k+1)}$ from $x^{(k)}$ as follows:

1. Calculate $I(x^{(k)})$ by solving the following LP:
   Minimize: $\sum_i x_i$; *Subject to:* $P^{x^{(k)}}(x) \leq x$, if $x = P(x)$ is a maxPPS,
   or:
   Maximize: $\sum_i x_i$; *Subject to:* $P^{x^{(k)}}(x) \geq x$, if $x = P(x)$ is a minPPS.

2. For each coordinate $i = 1, 2, \dots, n$, set $x_i^{(k+1)}$ to be the maximum (non-negative) multiple of 2^{-h} which is $\leq \max\{0, I(x^{(k)})_i\}$. (In other words, we round $I(x^{(k)})$ down to the nearest 2^{-h} and ensure it is non-negative.)

Theorem 4.24. *Given any max/minPPS, $x = P(x)$, with LFP $\mathbf{0} < q^* < \mathbf{1}$, if we use the above algorithm with rounding parameter $h = j + 2 + 4|P|$, then the iterations are all defined, and for every $k \geq 0$ we have $0 \leq x^{(k)} \leq q^*$, and furthermore after $h = j+2+4|P|$ iterations we have:*

$$\|q^* - x^{(j+2+4|P|)}\|_\infty \leq 2^{-j}$$

The proof is very similar to the proof of Theorem 3.20. We prove this using a few lemmas.

Lemma 4.25. *If we run the rounded-down-GNM starting with $x^{(0)} := \mathbf{0}$ on a max/minPPS, $x = P(x)$, with LFP q^*, $\mathbf{0} < q^* < \mathbf{1}$, then for all $k \geq 0$, $x^{(k)}$ is well-defined and $0 \leq x^{(k)} \leq q^*$.*

Proof. The base case $x^{(0)} = 0$ is immediate for both.

For the induction step, suppose the claim holds for k and thus $0 \leq x^{(k)} \leq q^*$. From Proposition 4.8, $I(x^{(k)})$ is well-defined and $I(x^{(k)}) \leq q^*$. Furthermore, since $x^{(k+1)}$ is obtained from $I(x^{(k)})$ by rounding down all coordinates, except setting to 0 any that are negative, and since obviously $q^* > \mathbf{0}$, we have that $0 \leq x^{(k+1)} \leq q^*$.

□

Lemma 4.26. *For a max/minPPS, $x = P(x)$, with LFP q^*, such that $\mathbf{0} < q^* < \mathbf{1}$, if we apply rounded-down-GNM with parameter h, starting at $x^{(0)} := \mathbf{0}$, then for all $j' \geq 0$, we have:*

$$\|q^* - x^{(j'+1)}\|_\infty \leq 2^{-j'} + 2^{-h+1+4|P|}$$

Proof. Since $x^{(0)} := 0$:

$$q^* - x^{(0)} = q^* \leq \mathbf{1} \leq \frac{1}{(\mathbf{1} - q^*)_{\min}}(\mathbf{1} - q^*) \tag{4.9}$$

For any $k \geq 0$, if $q^* - x^{(k)} \leq \lambda(\mathbf{1} - q^*)$, then by Proposition 4.8(which was proved separately for maxPPSs and minPPSs, in Lemmas 4.12 and 4.22, respectively), we have:

$$q^* - I(x^{(k)}) \leq (\frac{\lambda}{2})(\mathbf{1} - q^*) \tag{4.10}$$

Observe that after every iteration $k > 0$, in every coordinate i we have:

$$x_i^{(k)} \geq I(x^{(k-1)})_i - 2^{-h} \tag{4.11}$$

This holds simply because we are rounding down $I(x^{(k-1)})_i$ by at most 2^{-h}, unless it is negative in which case $x_i^{(k)} = 0 > I(x^{(k-1)})_i$. Combining the two inequalities (4.10) and (4.11) yields the following inequality:

$$q^* - x^{(k+1)} \leq (\frac{\lambda}{2})(\mathbf{1} - q^*) + 2^{-h}\mathbf{1} \leq (\frac{\lambda}{2} + \frac{2^{-h}}{(\mathbf{1}-q^*)_{\min}})(\mathbf{1}-q^*)$$

Taking inequality (4.9) as the base case (with $\lambda = \frac{1}{(\mathbf{1}-q^*)_{\min}}$), by induction on k, for all $k \geq 0$:

$$q^* - x^{(k+1)} \leq (2^{-k} + \sum_{i=0}^{k} 2^{-(h+i)})\frac{1}{(\mathbf{1}-q^*)_{\min}}(\mathbf{1}-q^*)$$

But $\sum_{i=0}^{k} 2^{-(h+i)} \leq 2^{-h+1}$ and $\frac{\|\mathbf{1}-q^*\|_\infty}{(\mathbf{1}-q^*)_{\min}} \leq \frac{1}{(\mathbf{1}-q^*)_{\min}} \leq 2^{4|P|}$, by Lemma 4.23. Thus:

$$q^* - x^{(k+1)} \leq (2^{-k} + 2^{-h+1})2^{4|P|}\mathbf{1}$$

Clearly, we have $q^* - x^{(k)} \geq 0$ for all k. Thus we have shown that for all $k \geq 0$:

$$\|q^* - x^{(k+1)}\|_\infty \leq (2^{-k} + 2^{-h+1})2^{4|P|} = 2^{-k} + 2^{-h+1+4|P|}.$$

□

Proof of Theorem 4.24. In Lemma 4.26 let $j' := j + 4|P| + 1$ and $h := j + 2 + 4|P|$. We have: $\|q^* - x^{(j+2+4|P|)}\|_\infty \leq 2^{-(j+1+4|P|)} + 2^{-(j+1)} \leq 2^{-(j+1)} + 2^{-(j+1)} = 2^{-j}$. □

Corollary 4.27. *Given any max/minPPS, $x = P(x)$, with LFP q^*, and given any integer $j > 0$, there is an algorithm that computes a rational vector v with $\|q^* - v\|_\infty \leq 2^{-j}$, in time polynomial in $|P|$ and j.*

Proof. First, we use the algorithms given in [EY05] (Theorems 11 and 13), to detect those variables x_i with $q_i^* = 0$ or $q_i^* = 1$ in time polynomial in $|P|$. Then we can remove these from the max/minPPS by substituting their known values into the equations for other variables. This gives us a max/minPPS with LFP $0 < q'^* < 1$ and does not increase $|P|$. Now we can use the iterated GNM, with rounding down, as outlined earlier in this section. In each iteration of GNM we solve an LP. Each LP has at most $n \leq |P|$ variables, at most $2n$ equations and the numerators and denominators of each rational coefficient are no larger than $2^{j+2+4|P|}$, so it can be solved in time polynomial in $|P|$ and j using standard algorithms. We need only $j + 2 + 4|P|$ iterations involving one LP each. Putting back the removed 0 and 1 values into the resulting vector gives us the full result q^*. This can all be done in polynomial time. □

4.2 Computing an ε-optimal policy in P-time

First let us note that we can not hope to compute an optimal policy in P-time, without a major breakthrough:

Theorem 4.28. *Computing an optimal policy for a max/minPPS is PosSLP-hard.*

Proof. Recall from Chapter 2[cn] that the termination probability vector q^* of an SCFG (equivalently, of a 1-exit RMC) can be equivalently viewed as the LFP of a purely probabilistic PPS, and vice-versa.

It was shown in [EY09] (Theorems 5.1 and 5.3), that given a PPS (equivalently, an SCFG or 1-exit Recursive Markov Chain), and given a rational probability p, it is PosSLP-hard to decide whether the LFP $q^*_1 > p$, for a given rational p, as well as to decide whether $q^*_1 < p$. (In fact, these hardness results hold already even if $p = 1/2$.)

The fact that computing an optimal policy for max/minPPS is PosSLP-hard follows easily from this: For the case of maxPPSs (minPPS, respectively), given a PPS, x=P(x), and given p, we simply add a new variable x_0 to the PPS, and a corresponding equation:

$$x_0 = \max\{p, x_1\} \quad (= \min\{p, x_1\}) \tag{4.12}$$

It is clear that $q^*_i > p$ ($q^*_i < p$, respectively) for the original PPS, if and only if in any optimal policy σ, for the augmented maxPPS (minPPS, respectively), the policy picks x_1 rather than p on the RHS of equation 4.12. So, if we could compute an optimal policy for a maxPPS (minPPS), we would be able to decide whether $q^*_i > p$ (whether $q^*_i < p$, respectively). □

Since we can not hope to compute an optimal policy for max/minPPSs in P-time without a major breakthrough, we will instead seek to find a policy σ such that $\|q^*_\sigma - q^*\|_\infty \leq \varepsilon$ for a given desired $\varepsilon > 0$, in time $poly(|P|, \log(1/\varepsilon))$. We have an algorithm for approximating q^*. Can we use a sufficiently close approximation, q, to q^* to find such an ε-optimal strategy? Once we have an approximation q, it seems natural to consider policies σ such that $P_\sigma(q) = P(q)$. For minPPSs, this means choosing the variable that has the lowest *approximate* value q_i and for maxPPS choosing the variable that has the highest *approximate* value. It turns out that this works as long as we can establish good enough upper bounds on the norm of $(I - B_\sigma(x))^{-1}$ for certain values of x. Recall that for a square matrix A, $\rho(A)$ denotes its spectral radius. For a vector x, the l_∞ norm is $\|x\|_\infty := \max_i |x_i|$, and its associated matrix norm $\|A\|_\infty$ is the maximum absolute-value row sum of A, i.e., $\|A\|_\infty := \max_i \sum_j |A_{i,j}|$.

Theorem 4.29. *For a max/minPPS, $x = P(x)$, given $0 \leq q \leq q^*$, such that $q < 1$, and a policy σ such that $P(q) = P_\sigma(q)$, and such that $\rho(B_\sigma(\frac{1}{2}(q^* + q^*_\sigma))) < 1$, and thus $(I - B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)))^{-1}$ exists and is non-negative, then*

$$\|q^*_\sigma - q^*\|_\infty \leq (2\|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q^*)))^{-1}\|_\infty + 1)\|q^* - q\|_\infty$$

Proof. We know that q is close to q^*. We just have to show that q is close to q^*_σ as well. We have to exploit some results about PPSs established in Chapter 3.

Lemma 4.30. *If $x = P(x)$ is a PPS, with LFP q^*, such that $\mathbf{0} < q^* \leq 1$, and $0 \leq y \leq q^*$, such that $y < 1$, then:*

$$q^* - y = (I - B(\frac{1}{2}(q^* + y)))^{-1}(P(y) - y)$$

Proof. Lemma 3.3 tells us that for any PPS, $x = P(x)$, (assumed to be in SNF form), and any pair of vectors $a, b \in \mathbb{R}^n$, we have $P(a) - P(b) = B((a+b)/2)(a-b)$. Applying this to $a = q^*$ and $b = y$, we have that

$$q^* - P(y) = B((1/2)(q^* + y))(q^* - y)$$

Subtracting both sides from $q^* - y$, we have that:

$$P(y) - y = (I - B((1/2)(q^* + y)))(q^* - y) \tag{4.13}$$

Now, by Lemma 4.15, we know that for any $z \leq q^*$, such that $z < 1$, $(I - B(z))^{-1}$ exists and is non-negative. But since $y \leq q^*$, clearly also $(1/2)(q^* + y) \leq q^*$, and since $y < 1$, and $q^* \leq 1$, then clearly $(1/2)(q^* + y) < 1$. Thus $(I - B((1/2)(q^* + y))^{-1}$ exists and is non-negative. Multiplying both sides of equation (4.13) by $(I - B((1/2)(q^* + y))^{-1}$, we obtain:

$$q^* - y = (I - B(1/2(q^* + y))^{-1}(P(y) - y)$$

as required. □

By assumption, σ was chosen such that $P(q) = P_\sigma(q)$. Note also that since $0 \leq q \leq q^*$, we have $0 \leq B_\sigma(\frac{1}{2}(q + q^*_\sigma)) \leq B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)$, and thus $0 \leq \rho(B_\sigma(\frac{1}{2}(q + q^*_\sigma))) \leq \rho(B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)) < 1$. Thus $(I - (B_\sigma(\frac{1}{2}(q + q^*_\sigma)))^{-1}$ also exists and is non-negative. Using this, and applying Lemma 4.30 to the PPS $x = P_\sigma(x)$, where we set $y := q$, and taking norms, we obtain the following inequality:

$$\|q^*_\sigma - q\|_\infty \leq \|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q)))^{-1}\|_\infty \|P(q) - q\|_\infty \tag{4.14}$$

To find a bound on $\|P(q) - q\|_\infty$, we need the following:

Lemma 4.31. *If $x = P(x)$ is a max/minPPS, and if $0 \leq y \leq q^*$, then $||P(y) - y||_\infty \leq 2||q^* - y||_\infty$.*

Proof. Suppose that $x = P(x)$ is a PPS. By Lemma 3.3, we have that $q^* - P(y) = B(\frac{1}{2}(y+q^*))(q^* - y)$. Since $\frac{1}{2}(y+q^*) \leq 1$, $||B(\frac{1}{2}(y+q^*))||_\infty \leq 2$: If the ith row has $x_i = P_i(x)$ of type L then $\sum_{j=1}^{n} |p_{i,j}| \leq 1$ and if $x_i = P_i(x)$ has type Q, then $\sum_{j=1}^{n} |\frac{\partial P_i(x)}{\partial x_j}(\frac{1}{2}(y+q^*))| = \frac{1}{2}(y_j + q^*_j) + \frac{1}{2}(y_k + q^*_k) \leq 2$. So we have that $||q^* - P(y)||_\infty \leq ||B(\frac{1}{2}(y+q^*))||_\infty ||q^* - y||_\infty \leq 2||q^* - y||_\infty$. As well as $y \leq q^*$, we know that $P(y) \leq q^*$ since $P(x)$ is monotone. If $(P(y))_i \leq y_i$, then $y_i - P(y)_i \leq q^*_i - P(y)_i \leq ||q^* - P(y)||_\infty \leq 2||q^* - y||_\infty$. If $P_i(y) \geq y_i$, $P_i(y) - y_i \leq q^*_i - y_i \leq ||q^* - y||_\infty$. So $||P(y) - y||_\infty \leq 2||q^* - y||_\infty$ as required.

If $x = P(x)$ is a max/minPPS, then it has some optimal policy, τ, and from the above, $||P_\tau(y) - y||_\infty \leq 2||q^* - y||_\infty$. It thus only remains to show that $|P_i(y) - y_i| \leq 2||q^* - y||_\infty$ when $x_i = P_i(x)$ is of form `M` (because the other equations don't change in $x = P_\tau(x)$).

If $P_i(y) \geq y_i$, then this is follows easily: as before we have that $P_i(y) - y_i \leq q^*_i - y_i \leq ||q^* - y||_\infty$. Suppose that instead we have $P_i(y) \leq y_i$. Then we consider the two cases (min and max) separately:

Suppose $x = P(x)$ is a minPPS, and that $P_i(x) = \min\{x_j, x_k\}$. Since $q^* = P(q^*)$, we have:

$$0 \leq y_i - P_i(y) \leq q^*_i - P_i(y) = \min\{q^*_j, q^*_k\} - P_i(y) \tag{4.15}$$

We can assume, w.l.o.g., that $P_i(y) \equiv \min\{y_j, y_k\} = y_j$. (The case where $P_i(y) = y_k$ is entirely analogous.) Then, by (4.15), we have:

$$0 \leq y_i - P(y)_i \leq \min\{q^*_j, q^*_k\} - y_j \leq q^*_j - y_j \leq ||q^* - y||_\infty$$

Suppose now that $x = P(x)$ is a maxPPS, and that $P_i(x) \equiv \max\{x_j, x_k\}$. Again, we are already assuming that $P_i(y) \leq y_i$. Since $q^* = P(q^*)$, we have:

$$0 \leq y_i - P_i(y) \leq q^*_i - P_i(y) = P_i(q^*) - \max\{y_j, y_k\} \tag{4.16}$$

We can assume, w.l.o.g., that $P_i(q^*) \equiv \max\{q^*_j, q^*_k\} = q^*_j$. (Again, the case when $P_i(q^*) = q^*_k$ is entirely analogous.) Then, by (4.16), we have:

$$0 \leq y_i - P_i(y) \leq q^*_j - \max\{y_j, y_k\} \leq q^*_j - y_j \leq ||q^* - y||_\infty$$

This completes the proof of the Lemma for all max/minPPSs. □

Now, we can show the result:

$$\begin{aligned}
\|q^* - q^*_\sigma\|_\infty &\leq \|q^* - q\|_\infty + \|q^*_\sigma - q\|_\infty \\
&\leq \|q^* - q\|_\infty + \|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q)))^{-1}\|_\infty \|P_\sigma(q) - q\|_\infty \\
&= \|q^* - q\|_\infty + \|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q)))^{-1}\|_\infty \|P(q) - q\|_\infty \\
&\leq \|q^* - q\|_\infty + \|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q)))^{-1}\|_\infty 2\|q^* - q\|_\infty \\
&= (2\|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q)))^{-1}\|_\infty + 1)\|q^* - q\|_\infty \\
&\leq (2\|(I - B_\sigma(\frac{1}{2}(q^*_\sigma + q^*)))^{-1}\|_\infty + 1)\|q^* - q\|_\infty
\end{aligned}$$

The last inequality follows because $q \leq q^*$, and

$$0 \leq (I - B_\sigma(q^*_\sigma + q))^{-1} = \sum_{i=0}^{\infty}(B_\sigma(q^*_\sigma + q))^i \leq \sum_{i=0}^{\infty}(B_\sigma(q^*_\sigma + q^*))^i = (I - B_\sigma(q^*_\sigma + q^*))^{-1}.$$

□

Finding these bounds is different for maxPPSs and minPPSs . Although we assume that $\mathbf{0} < q^* < \mathbf{1}$, for an arbitrary policy σ, it need not be true that $\mathbf{0} < q^*_\sigma < \mathbf{1}$. But the following obviously does hold:

Proposition 4.32. *Given a max/minPPS, $x = P(x)$, with LFP q^* such that $\mathbf{0} < q^* < \mathbf{1}$, for any policy σ:*
(i) If $x = P(x)$ is a maxPPS then $q^*_\sigma < 1$.
(ii) If $x = P(x)$ is a minPPS, then $q^*_\sigma > 0$.

Proof. This is trivial: if $x = P(x)$ is a maxPPS, then clearly $q^*_\sigma \leq q^* < \mathbf{1}$, because σ can be no better than an optimal strategy. Likewise, if $x = P(x)$ is a minPPS, then $\mathbf{0} < q^* \leq q^*_\sigma$, for the same reason. □

For maxPPSs, we may have that some coordinate of q^*_σ is equal to 0 and for minPPSs we may have that some coordinate of q^*_σ is equal to 1, even when $\mathbf{0} < q^* < \mathbf{1}$. This is the source of the different complications.

We use Theorem 3.24 from Chapter 3, which we restate here, to obtain the norm bounds we need for Theorem 4.29:

Theorem 3.24. *If $x = P(x)$ is a PPS with LFP $q^* > \mathbf{0}$ then*

(i) *If $q^* < \mathbf{1}$ and $0 \leq y < 1$, then $(I - B(\frac{1}{2}(y + q^*)))^{-1}$ exists and is non-negative, and*

$$\|(I - B(\frac{1}{2}(y + q^*)))^{-1}\|_\infty \leq 2^{10|P|} max\,\{2(1-y)^{-1}_{\min}, 2^{|P|}\}$$

(ii) *If $q^* = 1$ and $x = P(x)$ is strongly connected (i.e. every variable depends on every other) and $0 \leq y < 1 = q^*$, then $(I - B(y))^{-1}$ exists and is non-negative, and*

$$\|(I - B(y))^{-1}\|_\infty \leq 2^{4|P|} \frac{1}{(1-y)_{\min}}$$

We first focus on minPPSs, for which we shall show that if y is a close approximation to q^*, then any policy σ with $P(y) = P_\sigma(y)$ is ε-optimal. The maxPPS case will not be so simple: the analogous statement is false for maxPPSs.

Theorem 4.33. *If $x = P(x)$ is a minPPS, with LFP $\mathbf{0} < q^* < \mathbf{1}$, and $0 \leq \varepsilon \leq 1$, and $0 \leq y \leq q^*$, such that $\|q^* - y\|_\infty \leq 2^{-14|P|-3}\varepsilon$, then for any policy σ with $P_\sigma(y) = P(y)$, $\|q^* - q^*_\sigma\|_\infty \leq \varepsilon$.*

Proof. By Proposition 4.32, $q^*_\sigma \geq q^*$, and so $q^*_\sigma > 0$. Suppose for now that $q^*_\sigma < 1$ (we will show this later). Then applying Theorem 3.24 (i), for the case where we set $y := q^*$ and the PPS is $x = P_\sigma(x)$, yields that

$$\|(I - B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)))^{-1}\|_\infty \leq 2^{10|P_\sigma|}\max\{\frac{2}{(\mathbf{1} - q^*)_{\min}}, 2^{|P|}\}$$

Note that $|P_\sigma| \leq |P|$. Since for any minPPS, $x = P(x)$, there is an optimal strategy τ, and $x = P_\tau(x)$ is a PPS with the same LFP, $q^*_\tau = q^*$, as $x = P(x)$, and furthermore since $|P_\tau| \leq |P|$, it follows from Theorem 3.14 that $(\mathbf{1} - q^*)_{\min} \geq 2^{-4|P|}$. Thus

$$\|(I - B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)))^{-1}\|_\infty \leq 2^{14|P|+1}$$

Theorem 4.29 now gives that

$$\|q^* - q^*_\sigma\|_\infty \leq (2^{14|P|+2} + 1)\|q^* - y\|_\infty \leq \varepsilon$$

Thus, under the assumption that $q^*_\sigma < 1$, we are done.

To complete the proof, we now show that $q^*_\sigma < 1$. Suppose, for a contradiction, that for some i, $(q^*_\sigma)_i = 1$. Then by results in [EY09], $x = P_\sigma(x)$ has a bottom strongly connected component S with $q^*_S = 1$. If x_i is in S then only variables in S appear in $(P_\sigma)_i(x)$, so we write $x_S = P_S(x)$ for the PPS which is formed by such equations. We also have that $B_S(1)$ is irreducible and that the least fixed point solution of $x_S = P_S(x_S)$ is $q^*_S = 1$. Take y_S to be the subvector of y with coordinates in S. Now if we apply Theorem 3.24 (ii), by taking the y in its statement to be $\frac{1}{2}(y_S + 1)$, it gives that

$$\|(I - B_S(\frac{1}{2}(y_S + 1)))^{-1}\|_\infty \leq 2^{4|P_S|}\frac{1}{\frac{1}{2}(1 - y_S)_{\min}}$$

But $|P_S| \leq |P|$ and $(1-y_S)_{\min} \geq (\mathbf{1}-q^*)_{\min} \geq 2^{-4|P|}$. Thus

$$\|(I-B_S(\frac{1}{2}(y_S+1)))^{-1}\|_\infty \leq 2^{8|P|+1}$$

Lemma 4.30 gives that

$$1-y_S = (I-B_S(\frac{1}{2}(1+y_S)))^{-1}(P_S(y_S)-y_S)$$

Taking norms and re-arranging gives:

$$\|P_S(y_S)-y_S)\|_\infty \geq \frac{\|1-y_S\|_\infty}{\|(I-B_S(\frac{1}{2}(y_S+1)))^{-1}\|_\infty} \geq \frac{2^{-4|P|}}{2^{8|P|+1}} \geq 2^{-12|P|-1}$$

However $\|P_S(y_S)-y_S)\|_\infty \leq \|P_\sigma(y)-y\|_\infty$ and $P_\sigma(y) = P(y)$. We deduce that $\|P(y)-y\|_\infty \geq 2^{-12|P|-1}$. Lemma 4.31 states that $\|P(y)-y\|_\infty \leq 2\|q^*-y\|_\infty$. We thus have $\|q^*-y\|_\infty \geq 2^{-12|P|-2}$. This contradicts our assumption that $\|q^*-y\|_\infty \leq 2^{-14|P|-3}\varepsilon$ for some $\varepsilon \leq 1$. □

Now we proceed to the harder case of maxPPSs. The main theorem in this case is the following.

Theorem 4.34. *If $x = P(x)$ is a maxPPS with $\mathbf{0} < q^* < \mathbf{1}$ and given $0 \leq \varepsilon \leq 1$ and a vector y, with $0 \leq y \leq q^*$, such that $\|q^*-y\|_\infty \leq 2^{-14|P|-2}\varepsilon$, there exists a policy σ such that $\|q^*-q^*_\sigma\|_\infty \leq \varepsilon$, and furthermore, such a policy can be computed in P-time, given $x = P(x)$ and y.*

We need a policy σ for which we can apply Theorem 3.24, and for which we can get good bounds on $\|P_\sigma(y)-y\|_\infty$. Firstly we show that such policies exist. In fact, any optimal policy will do: for an optimal policy τ, $q^*_\tau > 0$ and Lemma 4.31 applied to $x = P_\tau(x)$ gives that $\|P_\tau(y)-y\|_\infty \leq 2^{-14|P|-1}\varepsilon$. Unfortunately the optimal policy might be hard to find (Theorem 4.28). We can however, given a policy σ and the PPS $x = P_\sigma(x)$, easily detect in polynomial time whether $q^*_\sigma > 0$ (see, e.g., Theorem 2.2 of [EY09], and also [ABE$^+$05]). We shall also make use of the following easy fact:

Lemma 4.35. *If $x = P(x)$ is a PPS with n variables, and with LFP q^*, then for any variable index $i \in \{1,\ldots,n\}$ the following are equivalent*
(i) $q^*_i > 0$.
(ii) there is a $k > 0$ such that $(P^k(0))_i > 0$.
(iii) $(P^n(0))_i > 0$.

Proof. (i) $\implies$ (ii): From [EY09], $P^k(0) \to q^*$ as $k \to \infty$. It follows that if $(P^k(0))_i = 0$ for all k, then $q^*_i = 0$.
(ii) $\implies$ (iii): Firstly, if there is a $1 \leq k < n$ with $(P^k(0))_i > 0$ then $(P^n(0))_i > 0$. $P(0) \geq 0$ and so by monotonicity and an easy induction $P^{l+1}(0) \geq P^l(0)$ for all $l > 0$. Another induction gives that $P^m(0) \geq P^l(0)$ when $m \geq l > 0$. As $k < n$, $(P^n(0))_i \geq (P^k(0))_i > 0$.

Whether $P_i(x) > 0$ depends only on whether each $x_j > 0$ or not and not on the value of x_j. So, for any k, whether $(P^{k+1}(0))_i > 0$ depends only on the set $S_k = \{x_j \text{ such that } (P^k(0))_j > 0\}$. From before $P^{k+1}(0) \geq P^k(0)$, so $S_{k+1} \supseteq S_k$. If ever we have that $S_{k+1} = S_k$, then for any j, $(P^{k+2}(0))_j > 0$ whenever $(P^{k+1}(0))_j > 0$ so $S_{k+2} = S_{k+1} = S_k$. $S_{k+1} \supset S_k$ can only occur for n values of k as there are only n variables to add. Consequently $S_{n+1} = S_n$ and so $S_m = S_n$ whenever $m > n$. So if we have a $k > n$ with $(P^k(0))_i > 0$, then $(P^n(0))_i > 0$

(iii) $\implies$ (i): By monotonicity and an easy induction, $q^* \geq P^k(0)$ for all $k > 0$. In particular $q^* \geq P^n(0)$. So $q^*_i \geq (P^n(0))_i > 0$. □

Given the maxPPS, $x = P(x)$, with $\mathbf{0} < q^* < \mathbf{1}$, and given a vector y that satisfies the conditions of Theorem 4.34, we shall use the following algorithm to obtain the policy we need:

1. Initialize the policy σ to any policy such that $P_\sigma(y) = P(y)$.

2. Calculate for which variables x_i in $x = P_\sigma(x)$ we have $(q^*_\sigma)_i = 0$. Let S_0 denote this set of variables. (We can do this in P-time; see, e.g., Theorem 2.2 of [EY09].)

3. If for all i we have $(q^*_\sigma)_i > 0$, i.e., if $S_0 = \emptyset$, then terminate and output the policy σ.

4. Otherwise, look for a variable x_i, where $P_i(x)$ is of form `M`, with $P_i(x) = \max\{x_j, x_k\}$, and where $(q^*_\sigma)_i = 0$ but one of x_j, x_k, say x_j, has $(q^*_\sigma)_j > 0$ and where furthermore $\|y_i - y_j\| \leq 2^{-14|P|-1}\varepsilon$. (We shall establish that such a pair x_i and x_j will always exist when we are at this step of the algorithm.)

   Let σ' be the policy that chooses x_j at x_i but is otherwise identical to σ. Set $\sigma := \sigma'$ and return to step 2.

Lemma 4.36. *The steps of the above algorithm are always well-defined, and the algorithm always terminates with a policy σ such that $q^*_\sigma > 0$ and $\|P_\sigma(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon$.*

Proof. Firstly, to show that the steps of the algorithm are always well-defined, we need to show that if there exists an x_i with $(q^*_\sigma)_i = 0$, then step 4 will find some variable to switch to. Suppose there is such an x_i. Let τ be an optimal policy. $(q^*_\tau)_i = q^*_i > 0$. So by Lemma 4.35, $(P^n_\tau)_i > 0$. For any variable x_j with $(P_\tau(0))_j > 0$, the equation $x_j = P_j(x)$ must have form `L` and not `M` so $(P_\sigma(0))_j > 0$ and so $(q^*_\sigma)_j > 0$. There must be a least k, $k_{\min}$ with $1 < k_{\min} \leq n$, such that there is a variable x_j with $(P^k_\tau(0))_j > 0$ but $(q^*_\sigma)_j = 0$. Let $x_{i'}$ be a variable such that $(P^{k_{\min}}_\tau(0))_{i'} > 0$ but $(q^*_\sigma)_{i'} = 0$.

Suppose that $x_{i'} = P_{i'}(x)$ has form `Q`, then $P_{i'}(x) = x_j x_l$ for some variables x_j, x_l. We have $0 < (P^{k_{\min}}_\tau(0))_{i'} = (P^{k_{\min}-1}_\tau(0))_j (P^{k_{\min}-1}_\tau(0))_l$. So $(P^{k_{\min}-1}_\tau(0))_j > 0$ and $(P^{k_{\min}-1}_\tau(0))_l > 0$. The minimality of $k_{\min}$ now gives us that $(q^*_\sigma)_j > 0$ and $(q^*_\sigma)_l > 0$. So $(q^*_\sigma)_{i'} = (q^*_\sigma)_j (q^*_\sigma)_l > 0$. This is a contradiction. Thus, $x_{i'} = P_{i'}(x)$ does not have form `Q`.

Similarly, $x_{i'} = P_{i'}(x)$ does not have form `L`. So $x_{i'} = P_{i'}(x)$ has form `M`. There are variables x_j, x_l with $P_{i'}(x) = \max\,\{x_j, x_l\}$. Suppose, w.l.o.g. that $(P_\tau(x))_{i'} = x_j$. We have $P^{k_{\min}}_\tau(0))_{i'} > 0$ and so $(P^{k_{\min}-1}(0))_j > 0$. By minimality of $k_{\min}$, we have that $(q^*_\sigma)_j > 0$. We have that $(q^*_\sigma)_{i'} = 0$ and so $(P_\sigma(x))_{i'} = x_l$.

Lemma 4.31 applied to the system $x = P_\tau(x)$ gives that $\|P_\tau(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon$. So $|y_{i'} - y_j| = |y_{i'} - (P_\tau(y))_{i'}| \leq 2^{-14|P|-1}\varepsilon$. Thus, step 4 could use i' and change the policy σ at i' (i.e., switch $\sigma(i')$) from x_l to x_j.

Next, we need to show that the algorithm terminates:

Claim 4.37. *If step 4 switches the variable x_i with $P_i(x) = max\ \{x_j, x_k\}$ from $(P_\sigma(x))_i = x_k$ to $(P_{\sigma'}(x))_i = x_j$, then*
(i) $q^*_{\sigma'} \geq q^*_\sigma$,
(ii) $(q^*_{\sigma'})_i > 0$,
(iii) The set of variables x_l with $(q^*_{\sigma'})_l > 0$ is a strict superset of the set of variables x_l with $(q^*_\sigma)_l > 0$.

Proof. Recall that step 4 will only switch if $(q^*_\sigma)_i = 0$ and $(q^*_\sigma)_j > 0$.

(i) We show that, for any $t > 0$, $P^t_{\sigma'}(0) \geq P^t_\sigma(0)$.

The base case $t = 1$, is clear, because the only indices i where $P_i(0) \neq 0$ are when $P_i(0)$ has form `L`, in which case $P_i(0) = (P_{\sigma'}(0))_i = (P_\sigma(0))_i$.

For the inductive case: note firstly that $P_\sigma(x)$ and $P_{\sigma'}(x)$ only differ on the ith coordinate. $(q^*_\sigma)_i = 0$, so for any t, $(P^t_\sigma(0))_i = 0$. Suppose that $P^t_{\sigma'}(0) \geq P^t_\sigma(0)$. Then by monotonicity $P^{t+1}_{\sigma'}(0) \geq P_{\sigma'}(P^t_\sigma(0))$. But $(P_{\sigma'}(P^t_\sigma(0)))_r = (P^{t+1}_\sigma(0))_r$ when

$r \neq i$. Furthermore, $(P_{\sigma'}(P^t_\sigma(0)))_i \geq 0 = (P^{t+1}_\sigma(0))_i$. So $P_{\sigma'}(P^k_\sigma(0)) \geq P^{k+1}_\sigma(0)$. We thus have that $P^{t+1}_{\sigma'}(0) \geq P^{t+1}_\sigma(0)$.

We know that as $t \to \infty$, $P^t_{\sigma'}(0) \to q^*_{\sigma'}$ and $P^t_\sigma(0) \to q^*_\sigma$. So $q^*_{\sigma'} \geq q^*_\sigma$.

(ii) We have $(q^*_{\sigma'})_i = (q^*_{\sigma'})_j$. By (i) $(q^*_{\sigma'})_j \geq (q^*_\sigma)_j$. We chose x_j such that $(q^*_\sigma)_j > 0$. So $(q^*_{\sigma'})_i > 0$.

(iii) If $(q^*_\sigma)_l > 0$, then by (i) $(q^*_{\sigma'})_l > 0$. Also $(q^*_\sigma)_i = 0$ and by (ii) $(q^*_{\sigma'})_i > 0$.

□

Thus, if at some stage of the algorithm we do not yet have $q^*_\sigma > 0$, then step 4 always gives us a new σ' with more coordinates having $(q^*_{\sigma'})_i > 0$. Furthermore, note that if $\|P_\sigma(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon$ then $\|P_{\sigma'}(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon$. Our starting policy has $\|P_\sigma(y) - y\|_\infty = \|P(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon$. The algorithm terminates and gives a σ with $q^*_\sigma > 0$ and $\|P_\sigma(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon$. □

We can now complete the proof of the Theorem:

Proof of Theorem 4.34. Using the algorithm, we find a σ with $\|y - P_\sigma(y)\|_\infty \leq 2^{-14|P|-1}\varepsilon$ and $q^*_\sigma > 0$. By Proposition 4.32, $q^*_\sigma < 1$. Applying Theorem 3.24 (i) to the PPS $x = P_\sigma(x)$ and point $y := q^*$ (not to be confused with the y in the statement of Theorem 4.34), gives that

$$\|(I - B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)))^{-1}\|_\infty \leq 2^{10|P_\sigma|}\max\{\frac{2}{(\mathbf{1} - q^*)_{\min}}, 2^{|P|}\}$$

We have $|P_\sigma| \leq |P|$. Also, from the fact there always exists an optimal policy, and from Theorem 3.14, it follows that we have $(\mathbf{1} - q^*)_{\min} \geq 2^{-4|P|}$. So

$$\|(I - B_\sigma(\frac{1}{2}(q^* + q^*_\sigma)))^{-1}\|_\infty \leq 2^{14|P|+1} \tag{4.17}$$

We can not use Theorem 4.29 as stated because we need not have $P(y) = P_\sigma(y)$. We do however have

$$\|P_\sigma(y) - y\|_\infty \leq 2^{-14|P|-1}\varepsilon \tag{4.18}$$

Applying Lemma 4.30, and taking norms, we get the inequality

$$\|q^*_\sigma - y\|_\infty \leq \|(I - B(\frac{1}{2}(q^*_\sigma + y)))^{-1}\|_\infty \|P(y) - y\|_\infty \tag{4.19}$$

Combining (4.17), (4.18) and (4.19) yields:

$$\|q^*_\sigma - y\|_\infty \leq \frac{1}{2}\varepsilon$$

so $\|q^*_\sigma - q^*\|_\infty \leq \|q^*_\sigma - y\|_\infty + \|q^* - y\|_\infty \leq \frac{1}{2}\varepsilon + 2^{-14|P|-2}\varepsilon \leq \varepsilon$. □

Theorem 4.38. *Given a max/minPPS, $x = P(x)$, and given $\varepsilon > 0$, we can compute an ε-optimal policy for $x = P(x)$ in time poly($|P|$, log $(1/\varepsilon)$)*

Proof. First we use the algorithms from [EY05] to detect variables x_i with $q^*_i = 0$ or $q^*_i = 1$ in time polynomial in $|P|$. Then we can remove these from the max/minPPS by substituting the known values into the equations for other variables. This gives us an max/minPPS with least fixed point $0 < q'^* < 1$ and does not increase $|P|$. To use either Theorem 4.34 or Theorem 4.33, it suffices to have a y with $y < q^*$ with $q^* - y \leq 2^{-14|P|-3}\varepsilon$. Theorem 4.24 says that we can find such a y in time polynomial in $|P|$ and $14|P| - \log(\varepsilon)$, which is polynomial in $|P|$ and $\log(1/\varepsilon)$ as required. Now depending on whether we have a maxPPS or minPPS, Theorem 4.34 or Theorem 4.33 show that from this y, we can find an ε-optimal policy for the max/minPPS with $0 < q'^* < 1$ in time polynomial in $|P|$ and $\log(1/\varepsilon)$. All that is left to show is that we can extend this policy to the variables x_i where $q^*_i = 0$ or $q^*_i = 1$ while still remaining ε-optimal.

We next show how this can be done.

For a minPPS, if $q^*_i = 1$ then for any policy σ, $(q^*_\sigma)_i = 1$ so the choice made at such variables x_i is irrelevant. Similarly, for maxPPSs, when $q^*_i = 0$, any choice at x_i is optimal.

For a minPPS with $q^*_i = 0$, if $P_i(x)$ has form `M`, we can choose any variable x_j with $q^*_j = 0$. There is such a variable: if $P_i(x) = \min\{x_j, x_k\}$ and $q^*_i = 0$ then either $q^*_j = 0$ or $q^*_k = 0$. Let σ be a policy such that for each variable x_i with $q^*_i = 0$, $(q^*)_{\sigma(i)} = 0$. We need to show that $(q^*_\sigma)_i = 0$ for all such variables. Suppose that, for some $k \geq 0$, $(P^k_\sigma(0))_i = 0$ for all x_i such that $q^*_i = 0$. Then $P(P^k_\sigma(0))_i = 0$ for all x_i with $q^*_i = 0$.

To see why this is so, note that whether or not $P_i(z) = 0$ depends only on which coordinates of z are 0, and furthermore if $P_i(z) = 0$ when the set of 0 coordinates of z is S, then for any vector z' where the 0 coordinates of z' are $S' \supseteq S$, we have $P_i(z') = 0$. Since the coordinate S that are 0 in q^* are a subset of the coordinates S' that are 0 in $P^k_\sigma(0)$, and we have $P_i(q^*) = q^*_i = 0$, we thus have $P(P^k_\sigma(0))_i = 0$.

If $P_i(x) = \min\{x_j, x_k\}$ and $q^*_i = 0$ then either $q^*_j = 0$ or $q^*_k = 0$. Suppose w.l.o.g. that $(P_\sigma(x))_i = x_j$. Then $q^*_j = 0$, so by assumption $(P^k_\sigma(0))_j = 0$ and so $(P_\sigma(P^k_\sigma(0)))_i = 0$. We now have enough for $(P^{k+1}_\sigma(0))_i = 0$ for each variable x_i with $q^*_i = 0$. $P^0_\sigma(0) = 0$, so by induction for all $k \geq 0$, $(P^k_\sigma(0))_i = 0$ for all x_i with $q^*_i = 0$. From this, for each variable x_i with $q^*_i = 0$, $(q^*_\sigma)_i = 0$.

The case of a maxPPS that have variables with $q^*_i = 1$ is not so simple. The P-time algorithm given in [EY05] to detect vertices with $q^*_i = 1$, produces a partial randomised policy for such vertices (Lemma 12 in [EY05]). A randomised policy is a map $\rho : M \rightarrow$

$[0,1]$, that turns a maxPPS $x = P(x)$ into a PPS $x = P_\rho(x)$ by replacing equations of form $\mathtt{M}$, $P_i(x) = \max\ \{x_j, x_k\}$, with equations of form $\mathtt{L}$ $P_i(x) = \rho(i)x_j + (1-\rho(i))x_k$. We would prefer a non-randomised (pure) policy σ with $(q^*_\sigma)_i = 1$ for all variables x_i with $q^*_i = 1$. Theorem 2.6 (which quotes Theorem 2 of [EY05]) guarantees the existence of such a σ.

We can construct such a pure optimal partial policy. We start with $P_{(0)}(x) = P(x)$. Given an x_i with $(P_{(l)}(x))_i = \max\ \{x_j, x_k\}$ and $(q^*_{(l)})_i = 1$, we try setting $(P_{(l+1)}(x))_i = x_j$ and see if this gives $(q^*_{(l+1)})_i = 1$. If it does then set $(P_{(l+1)}(x)))_i = x_j$. If it does not then set $(P_{(l+1)}(x)))_i = x_k$. We can argue inductively that the LFP $q^*_{(l)}$ of $x = P_{(l)}(x)$ is equal to the LFP q^* of $x = P(x)$ for all l. The basis, $l = 0$, is clear. For the induction step. we know from Theorem 2.6 that there is an optimal policy σ for the maxPPS $x = P_{(l)}(x)$. If σ does not have $\sigma(i) = j$ then $\sigma(i) = k$. So if setting $(P_{(l+1)}(x))_i = x_j$ would not give $(q^*_{(l+1)})_i = 1$ then $(P_{(l+1)}(x))_i = x_k$ does give $(q^*_{(l+1)})_i = 1$. We have that $(q^*_{(l+1)})_r = (q^*_{(l)})_r$ when $r \neq i$ so $q^*_{(l+1)} = q^*_{(l)}$. When there are no x_i with $(P_{(l)}(x))_i = \max\ \{x_j, x_k\}$ and $(q^*_{(l)})_i = 1$, we have found a pure partial optimal policy for x_i with $q^*_i = 1$. This requires no more than n calls to the polynomial time algorithm given in [EY05] for determining for a maxPPS, $x = P(x)$ those coordinates i such that $q^*_i = 1$. □

4.3 Approximating the value of BSSGs in FNP

In this section we briefly note that, as an easy corollary of our results for BMDPs, we can obtain a TFNP (total NP search problem) upper bound for computing (approximately), the *value* of *Branching simple stochastic games* (BSSG), where the objective of the two players is to maximize, and minimize, the extinction probability. For relevant definitions and background results about these games see [EY05]. It suffices for our purposes here to point out that, as shown in [EY05], the value of these games (which are determined) is characterized by the LFP solution of associated minmaxPPSs, $x = P(x)$, where both min and max operators can occur in the equations for different variables. Furthermore, both players have optimal policies (i.e. optimal pure, memoryless strategies) in these games (see [EY05]).

Corollary 4.39. *Given a max-minPPS, $x = P(x)$, and given a rational $\varepsilon > 0$, the problem of approximating the LFP q^* of $x = P(x)$, i.e., computing a vector v such that $\|q^* - v\|_\infty \leq \varepsilon$, is in TFNP, as is the problem of computing ε-optimal policies for both* *players. (And thus also, the problem of approximating the value, and computing* ε*-optimal strategies, for BSSGs is in FNP.)*

Proof. Given $x = P(x)$, whose LFP, q^*, we wish to compute, first guess pure policies σ and τ for the max and min players, respectively. Then, fix σ as max's strategy, and for the resulting minPPS (with LFP q^*_σ) use our algorithm to compute in P-time an approximate value vector v_σ, such that $\|v_\sigma - q^*_\sigma\|_\infty \leq \varepsilon/4$. Next, fix τ as min's strategy, and for the resulting maxPPS (with LFP q^*_τ), use our algorithm to compute in P-time an approximate value vector v_τ, such that $\|v_\tau - q^*_\tau\|_\infty \leq \varepsilon/4$. Finally, check whether $\|v_\sigma - v_\tau\|_\infty \leq \varepsilon/4$. If not, then reject this "guess". If so, then output σ and τ as ε-optimal policies for max and min, respectively, and output $v := v_\sigma$ (or $v := v_\tau$) as an ε-approximation of the LFP, q^*. This procedure is correct because if q^* is the LFP of the min-maxPPS, $x = P(x)$, then $q^*_\sigma \leq q^* \leq q^*_\tau$, and thus:

$$\begin{aligned}
\|q^* - v_\sigma\|_\infty &\leq \|q^* - q^*_\sigma\|_\infty + \|q^*_\sigma - v_\sigma\|_\infty \\
&\leq \|q^*_\tau - q^*_\sigma\|_\infty + \|q^*_\sigma - v_\sigma\|_\infty \\
&\leq \|q^*_\tau - v_\tau\|_\infty + \|v_\tau - v_\sigma\|_\infty + \|v_\sigma - q^*_\sigma\|_\infty + \|q^*_\sigma - v_\sigma\|_\infty \\
&\leq \varepsilon
\end{aligned}$$

And likewise for v_τ. □

It is worth noting that the problem of approximating the value of a BSSG game, to within a desired $\varepsilon > 0$, when ε is given as part of the input, is already at least as hard as computing the *exact* value of Condon's finite-state simple stochastic games (SSGs) [Con92], and thus one can not hope for a P-time upper bound without a breakthrough. In fact, it was shown in [EY05] that even the *qualitative* problem of deciding whether the value $q^*_i = 1$ for a given BSSG (or max-minPPS), which was shown there to be in NP∩coNP, is already at least as hard as Condon's *quantitative* decision problem for finite-state simple stochastic games. (Whereas for finite-state SSGs the qualitative problem of deciding whether the value is 1 is in P-time.)

Chapter 5

General Monotone Polynomial Systems and Probabilistic One-Counter Automata

In this chapter we will established general worst-case upper bounds on the convergence behaviour of Newton's method for computing the least fixed point solution of monotone polynomial systems of equations (MPSs), as a function of both the encoding size of the system of equation, and $\log(1/\varepsilon)$, where $\varepsilon > 0$ is the desired additive error. Our bounds are essentially optimal in several important parameters of the problem. We shall also use these results in order to establish, in particular, a polynomial time algorithm (in the Turing model of computation) for approximating termination probabilities, and model checking probabilities, for probabilistic one-counter automata (p1CA).

As mentioned in the Introduction (Section 1.3), Etessami and Yannakakis [EY09] gave a *Decomposed Newton's method* for MPSs, and showed that it converges monotonically to the LFP solution. Subsequently, Esparza, Kiefer, and Luttenberger [EKL10] obtained upper bounds on the convergence rate of Newton's method for strongly connected MPSs. We shall use several results and techniques from [EY09, EKL10] and from the work in Chapter 3, and we shall develop substantial additional machinery, in order to obtain our upper bounds for general MPSs in this chapter.

In Chapters 3 and 6 we use an undecomposed version of Newton's method. In this chapter, we describe a *Rounded-down Decomposed Newton's Method* (**R-DNM**), a variant of the algorithm from [EY09], but with rounding, which applies to all MPSs. We will then analyze this algorithm to obtain bounds for both general MPSs, and better bounds for the specific class of MPSs that arise in the context of Probabilistic One-Counter Automata (p1CAs).

We now outline the results of this chapter in more detail.

Suppose that the given (cleaned) MPS $x = P(x)$ has a LFP $q^* > \mathbf{0}$. The decomposition into strongly connected components yields a DAG of SCCs with depth d, and we wish to compute the LFP with (absolute) error at most ε. Let $q^*_{\min}$ and $q^*_{\max}$ be the minimum and maximum coordinate of q^*. Then the rounded decomposed Newton method will converge to a vector $\tilde{q}$ within ε of the LFP, i.e., such that $\|q^* - \tilde{q}\|_\infty \leq \varepsilon$ in time polynomial in the size $|P|$ of the input, $\log(1/\varepsilon)$, $\log(1/q^*_{\min})$, $\log(q^*_{\max})$, and 2^d (the depth d in the exponent can be replaced by the maximum number of *nonlinear* SCCs in any path of the DAG of SCCs). We also obtain bounds on $q^*_{\min}$ and $q^*_{\max}$ in terms of $|P|$ and the number of variables n, so the overall time needed is polynomial in $|P|$, 2^n and $\log(1/\varepsilon)$. We provide actually concrete expressions on the number of iterations and the number of bits needed. As we shall explain, the bounds are essentially optimal in terms of several parameters. The analysis is quite involved and builds on the previous work.

We apply our results then to probabilistic 1-counter automata (p1CAs). Using our analysis for the rounded decomposed Newton method and properties of p1CAs from [EWY10], we show that termination probabilities of a p1CA M (and QBDs) can be computed to desired precision ε in polynomial time in the size $|M|$ of the p1CA and $\log(1/\varepsilon)$ (the bits of precision) in the standard Turing model of computation, thus solving the open problem of [EWY10].

Furthermore, combining with the results of [BKK11] and [EY12], we show that one can do quantitative model checking of ω-regular properties for p1CAs in polynomial time in the standard Turing model, i.e., we can compute to desired precision ε the probability that a run of a given p1CA M satisfies an ω-regular property in time polynomial in $|M|$ and $\log(1/\varepsilon)$ (and exponential in the size of the property if it is given for example as a non-deterministic Büchi automaton or polynomial if it is given as a deterministic Rabin automaton).

The rest of this chapter is organized as follows. In Section 5.1 we give some preliminary definitions and results, and we describe rounded-down decomposed Newton's method. In Section 5.2 we consider strongly-connected MPS, and in Section 5.3 general MPS. Section 5.4 analyzes p1CAs. Section 5.5 proves some upper and lower bounds on the values of the LFP solution of an MPS, which are stated (but not proved) in section 5.1.

5.1 Preliminaries

We now describe how ***Rounded-down Decomposed Newton's Method (R-DNM)*** works when applied to an MPS, $x = P(x)$, with real-valued LFP $q^* \geq 0$. Firstly, we use Proposition 2.12 to remove 0 variables, and thus we can assume we are given a cleaned MPS, $x = P(x)$, with real-valued LFP $q^* > \mathbf{0}$.

Let H_P be the DAG of SCC's of the dependency graph G_P. We work bottom-up in H_P, starting at bottom SCCs. For each SCC, S, suppose its corresponding equations are $x_S = P_S(x_S, x_{D(S)})$, where $D(S)$ denotes the union of the variables in "lower" SCCs, below S, on which S depends. In other words, a variable $x_j \in D(S)$ if and only if there is some variable $x_i \in S$ such that there is directed path in G_p from x_i to x_j. If the system $x_S = P_S(x_S, q^*_{D(S)})$ is a linear system (in x_S), we call S a *linear SCC*, otherwise S is a *nonlinear SCC*. Assume we have already calculated (using R-DNM) an approximation $\tilde{q}_{D(S)}$ to the LFP solution $q^*_{D(S)}$ for these lower SCCs. We plug in $\tilde{q}_{D(S)}$ into the equations for S, obtaining the equation system $x_S = P_S(x_S, \tilde{q}_{D(S)})$. We denote the actual LFP solution of this new equation system by q'_S. (Note that q'_S is not necessarily equal to q^*_S, because $\tilde{q}_{D(S)}$ is only an approximation of $q^*_{D(S)}$.)

If S is a nonlinear SCC, we apply a chosen number g of iterations of R-NM on the system $x_S = P_S(x_S, \tilde{q}_{D(S)})$ to obtain an approximation $\tilde{q}_S$ of q'_S; if S is linear then we just apply 1 iteration of R-NM, i.e., we solve the linear system and round down the solution. We of course want to make sure our approximations are such that $\|q^*_S - \tilde{q}_S\|_\infty \leq \varepsilon$, for all SCCs S, and for the desired additive error $\varepsilon > 0$. We shall establish upper bounds on the number of iterations g, and on the rounding parameter h, needed in R-DNM for this to hold, as a function of various parameters: the input size $|P|$ and the number n of variables; the *nonlinear depth* f of P, which is defined as the maximum, over all paths of the DAG H_P of SCCs, of the number of nonlinear SCCs on the path; and the maximum and minimum coordinates of the LFP.

Bounds on the size of LFPs for an MPS. For a positive vector $v > 0$, we use $v_{\min} = \min_i v_i$ to denote its minimum coordinate, and we use $v_{\max} = \max_i v_i$ to denote its maximum coordinate. Slightly overloading notation, for an MPS, $x = P(x)$, we shall use $c_{\min}$ to denote the minimum value of all positive monomial coefficients and all positive constant terms in $P(x)$. Note that $c_{\min}$ also serves as a lower bound for all positive constants and coefficients for entries of the Jacobian matrix $B(x)$, since $B(x)_{ij} = \frac{\partial P_i(x)}{\partial x_j}$.

We prove the following Theorem in 5.5, establishing bounds on the maximum and minimum coordinates of the LFP q^* of an MPS $x = P(x)$.

Theorem 5.1. *If $x = P(x)$ is a quadratic MPS in n variables, with LFP $q^* > \mathbf{0}$, and where $P(x)$ has rational coefficients and total encoding size $|P|$ bits, then*

1. $q^*_{\min} \geq 2^{-|P|(2^n-1)}$, and

2. $q^*_{\max} \leq 2^{2(n+1)(|P|+2(n+1)\log(2n+2))\cdot 5^n}$.

How good are our upper bounds? In section 5.5.1 we discuss how good our upper bounds on R-DNM are, and in what senses they are optimal, in light of the convergence rate of Newton's method on known bad examples ([EKL10]), and considerations relating to the size of $q^*_{\min}$ and $q^*_{\max}$. In this way, our upper bounds can be seen to be essentially optimal in several parameters, including the depth of SCCs in the dependency graph of the MPS, and in terms of $\log\frac{1}{\varepsilon}$.

5.2 Strongly Connected Monotone Polynomial Systems

Theorem 5.2. *Let $P(x,y)$ be an n-vector of monotone polynomials with degree ≤ 2 in variables which are coordinates of the n-vector x and the m-vector y, where $n \geq 1$ and $m \geq 1$.*

Given non-negative m-vectors y_1 and y_2 such that $0 < y_1 \leq \mathbf{1}$ and $0 \leq y_2 \leq y_1$, let $P_1(x) \equiv P(x,y_1)$ and $P_2(x) \equiv P(x,y_2)$. Suppose that $x = P_1(x)$ is a strongly-connected MPS with LFP solution $\mathbf{0} < q^*_1 \leq \mathbf{1}$.
Let $\alpha = \min\{1, c_{\min}\}\min\{y_{\min}, \frac{1}{2}q^*_{\min}\}$, where $c_{\min}$ is the smallest non-zero constant or coefficient of any monomial in $P(x,y)$, where $y_{\min}$ is the minimum coordinate of y_1, and finally where $q^*_{\min}$ is the minimum coordinate of q^*_1. Then:

1. *The LFP solution of the MPS $x = P_2(x)$ is q^*_2 with $0 \leq q^*_2 \leq q^*_1$, and*

$$\|q^*_1 - q^*_2\|_\infty \leq \sqrt{4n\alpha^{-(3n+1)}\|P(\mathbf{1},\mathbf{1})\|_\infty\|y_1 - y_2\|_\infty}$$

Furthermore, if $x = P_1(x)$ is a linear system, then:

$$\|q^*_1 - q^*_2\|_\infty \leq 2n\alpha^{-(n+2)}\|P(\mathbf{1},\mathbf{1})\|_\infty\|y_1 - y_2\|_\infty \tag{5.1}$$

2. *Moreover, for every $0 < \varepsilon < 1$, if we use $g \geq h - 1$ iterations of rounded down Newton's method with parameter*

$$h \geq \lceil 2 + n\log\frac{1}{\alpha} + \log\frac{1}{\varepsilon}\rceil$$

applied to the MPS, $x = P_2(x)$, starting at $x^{[0]} := \mathbf{0}$, to approximate q^*_2, then the iterations are all defined, and $\|q^*_2 - x^{[g]}\|_\infty \leq \varepsilon$.

Before proving Theorem 5.2, We recall some Lemmas from Chapter 3 which apply to general quadratic MPSs and not just the PPSs analysed there.

Lemma 3.3. *Let $x = P(x)$ be a quadratic MPS, with n variables, and let $a, b \in \mathbb{R}^n$. Then:*

$$P(a) - P(b) = B(\frac{a+b}{2})(a-b) = \frac{B(a)+B(b)}{2}(a-b)$$

Lemma 3.4. *Let $x = P(x)$ be a quadratic MPS. Let $z \in \mathbb{R}^n$ be any vector such that $(I - B(z))$ is non-singular, and thus $\mathcal{N}_P(z)$ is defined. Then:*

$$q^* - \mathcal{N}_P(z) = (I - B(z))^{-1} \frac{B(q^*) - B(z)}{2}(q^* - z)$$

Lemma 3.9. *Let $x = P(x)$ be a quadratic MPS with LFP, $q^* \geq 0$. Let $B(x)$ denote the Jacobian matrix of $P(x)$. For any positive vector $\boldsymbol{d} \in \mathbb{R}^n_{>0}$ that satisfies $B(q^*)\boldsymbol{d} \leq \boldsymbol{d}$, any positive real value $\lambda > 0$, and any nonnegative vector $z \in \mathbb{R}^n_{\geq 0}$, if $q^* - z \leq \lambda \boldsymbol{d}$, and $(I - B(z))^{-1}$ exists and is nonnegative, then $q^* - \mathcal{N}_P(z) \leq \frac{\lambda}{2}\boldsymbol{d}$.*

We also need to recall a number of basic facts from matrix analysis and Perron-Frobenius theory. For a square matrix A, let $\rho(A)$ denote the spectral radius of A. Recall that a nonnegative square matrix A is called *irreducible* if its underlying directed graph is strongly connected, where the adjacency matrix of its underlying directed graph is obtained by setting the positive entries of the matrix A to 1.

Lemma 5.3. *(see, e.g., [HJ85], Theorem 8.4.4) If A is an irreducible nonnegative square matrix, then there is a positive eigenvector $v > 0$, such that $Av = \rho(A)v$. Such a vector v is called the* Perron vector *of A. It is unique up to rescaling by a positive factor.*

Lemma 5.4. *(see, e.g., [LT85], Theorem 15.4.1 and Exercise 1, page 540) If A is an irreducible nonnegative square matrix and $0 \leq B \leq A$, but $B \neq A$, then $\rho(B) < \rho(A)$.*

Lemma 5.5. *(see, e.g., [LT85], Theorem 15.2.2, page 531) If A is a square matrix with $\rho(A) < 1$, then $I - A$ is non-singular and $(I - A)^{-1} = \sum_{i=0}^{\infty} A^i$.*

Lemma 5.6. *(see, e.g., [LT85], Section 15.3 and Exercise 11) If A is an irreducible nonnegative square matrix, and $v > 0$ is a positive eigenvector associated with some eigenvalue r, i.e., such that $Av = rv$, then $r = \rho(A)$. Thus $v > 0$ is the Perron vector (which is unique up to scaling).*

We are now ready to prove the Theorem.

Proof of Theorem 5.2. We first establish **1.** Since $x = P_1(x)$ is a strongly connected system of equations, and $q_1^* > 0$, this implies that matrix $B_1(q_1^*)$ is non-negative and irreducible, where $B_1(x)$ is the Jacobian matrix of $P_1(x)$.

Thus, by Lemma 5.3, there is a positive Perron eigenvector $v > 0$ of $B_1(q_1^*)$, which satisfies $B_1(q_1^*)v = \rho(B_1(q_1^*))v$. We can always scale v such that $\|v\|_\infty = 1$.

We will observe that $B_1(q_1^*)v \leq v$, and that if we scale v so that $\|v\|_\infty = 1$ then the smallest coordinate of v, denoted $v_{\min}$, has $v_{\min} \geq \alpha^n$.

Lemma 5.7. *(This is a variant of Lemma 6.5 from [EY09]) For any strongly-connected MPS, $x = P(x)$, with LFP $q^* > \mathbf{0}$, and Jacobian $B(x)$, we have $\rho(B(q^*)) \leq 1$, and for all vectors y with $0 \leq y < q^*$, $\rho(B(y)) < 1$.*

Proof. We will only show here that $\rho(B(q^*)) \leq 1$ if $x = P(x)$ is strongly connected, but in fact this holds for any MPS, $x = P(x)$, with LFP $q^* > \mathbf{0}$. We do so because we will only use the strongly-connected case.

If we have $0 \leq z \leq y$ and $z \leq P(z)$, then Lemma 6.4 of [EY09] shows that for any $d \geq 1$, $B^d(z)(y-z) \leq P^d(y) - P^d(z)$. Let $x^i = P^i(\mathbf{0})$, for all $i \geq 1$. Recall that $\lim_{i\to\infty} x^i = q^*$. Also note that, because $x = P(x)$ is strongly connected, $x^i < q^*$ for all i.

Then for all $i, d \geq 1$, $B^d(x^i)(q^* - x^i) \leq P^d(q^*) - P^d(x^i) = q^* - x^{i+d}$. But since $\lim_{d\to\infty} x^{i+d} = q^*$, we see that the right hand side goes to 0. But since $(q^* - x^i) > 0$ for all i, it must be the case that $B^d(x^i) \to 0$, as d goes to infinity. But this is a necessary and sufficient condition for $\rho(B(x^i)) < 1$. Now notice that for any vector y such that $0 \leq y < q^*$, there is some i such that $y \leq x^i$. Thus, by monotonicity of $\rho(B(x))$ in $x \geq 0$, we must have $\rho(B(y)) < 1$.

Thus, also, since $\lim_{i\to\infty} x^i = q^*$, and by continuity of the spectral radius function, we get that $\rho(B(q^*)) \leq 1$. □

Corollary 5.8. $\rho(B_1(q_1^*)) \leq 1$, *and thus if v is a Perron vector of $B_1(q_1^*)$ then $B_1(q_1^*)v = \rho(B_1(q_1^*))v \leq v$.*

The following basic Lemma, applied to $B_1(q_1^*)$ and its normalized Perron vector v, yields the desired result about v:

Lemma 5.9. *If A is a irreducible, non-negative $n \times n$ matrix with minimum non-zero entry $a_{\min}$, and $u \geq 0$ is a non-zero vector in $\mathbb{R}^n$ with $Au \leq u$, then $a_{\min} \leq 1$ and if the minimum and maximum coordinates of u are denoted $u_{\min}$ and $u_{\max}$, respectively, then we have $\frac{u_{\min}}{u_{\max}} \geq a_{\min}^n$. In particular $u > 0$.*

Proof. Let i,j be some coordinates with $u_i = u_{\min}$ and $u_j = u_{\max}$. Because A is irreducible and non-negative, there is a power $0 \le k \le n$ with $(A^k)_{ij} > 0$. By matrix multiplication, for any $k \ge 1$, $(A^k)_{ij} = \sum \prod_l A_{i_l,i_{l+1}}$, where the sum is taken over all length $k+1$ sequences of indices $i_1,\dots,i_{k+1}$, with $i_1 = i$ and $i_{k+1} = j$, and with $i_l \in \{1,\dots,n\}$ for all l ranging from 1 to k. At least one of these products is non-zero and thus it is at least $a^k_{\min}$. That is $(A^k)_{ij} \ge a^k_{\min}$. Since $Au \le u$, and A is non-negative, a simple induction gives that $A^k u \le u$. And since u is non-zero, $u_{\max} = u_j > 0$, so $0 < A^k_{ij} u_j \le u_i$. Since $u_i = u_{\min}$, this means $u > 0$. Also, $1 \ge \frac{u_{\min}}{u_{\max}} = \frac{u_i}{u_j} \ge A^k_{ij} \ge a^k_{\min}$. Note that since $1 \ge a^k_{\min}$, this implies $a_{\min} \le 1$. We know that $1 \le k \le n$, so $a^k_{\min} \ge a^n_{\min}$. □

Applying Lemma 5.9 to $A = B_1(q_1^*)$ and v the Perron vector of $B_1(q_1^*)$, normalized so that $v_{\max} = 1$, and observing that the smallest non-zero entry of $B_1(q_1^*)$ is at least α, we get:

Corollary 5.10. *If v is the Perron vector of $B_1(q_1^*)$, normalized so that $v_{\max} = 1$, then $\frac{v_{\min}}{v_{\max}} = v_{\min} \ge \alpha^n$.*

Next, to show that $0 \le q_2^* \le q_1^*$, we consider $P_1^k(0) = P_1(P_1(\dots P_1(0)\dots))$, i.e., the k'th iterate of P_1 applied to the vector $\mathbf{0}$, and $P_2^k(0)$. We know that for any MPS, $x = P(x)$ with LFP $q^* \in \mathbb{R}^n_{\ge 0}$, we have $\lim_{k\to\infty} P^k(0) = q^*$ ([EY09]). Thanks to the monotonicity of P, for any $x \ge 0$, we have $P_1(x) \ge P_2(x)$. By the monotonicity of P_1 and an easy induction, $P_1^k(0) \ge P_2^k(0)$. So $q_1^* \ge q_2^*$.

Next we want to obtain the bounds (5.2) and (5.1) on $\|q_1^* - q_2^*\|_\infty$. If $q_1^* = q_2^*$, then we are trivially done so we assume that $q_2^* \ne q_1^*$. Because $x = P_1(x)$ is at most quadratic, we can apply Lemma 3.3 to get:

$$B_1(\frac{1}{2}(q_1^* + q_2^*))(q_1^* - q_2^*) = P_1(q_1^*) - P_1(q_2^*) = q_1^* - P_1(q_2^*) \tag{5.2}$$

Multiplying both sides of equation (5.2) by -1, and then adding $(q_1^* - q_2^*)$ to both sides, we get:

$$\begin{aligned}(I - B_1(\frac{1}{2}(q_1^* + q_2^*)))(q_1^* - q_2^*) &= (q_1^* - q_2^*) - (q_1^* - P_1(q_2^*)) \\ &= P_1(q_2^*) - q_2^* \\ &= P_1(q_2^*) - P_2(q_2^*) \qquad (5.3)\end{aligned}$$

Provided that $(I - B_1(\frac{1}{2}(q_1^* + q_2^*))$ is non-singular, we can multiply both sides of equation (5.3) by $(I - B_1(\frac{1}{2}(q_1^* + q_2^*))^{-1}$, to get

$$q_1^* - q_2^* = (I - B_1(\frac{1}{2}(q_1^* + q_2^*))^{-1}(P_1(q_2^*) - P_2(q_2^*)) \tag{5.4}$$

We will be taking the $\|.\|_\infty$ norm of equation (5.4) to obtain the bound we need for $\|q_2^* - q_1^*\|_\infty$. To do this we first need to bound $\|(I - B_1(\frac{1}{2}(q_1^* + q_2^*))^{-1}\|_\infty$, and in particular we need to show that $(I - B_1(\frac{1}{2}(q_1^* + q_2^*))^{-1}$ is nonsingular.

By (5.2) we have $q_1^* - P_1(q_2^*) = B_1(\frac{1}{2}(q_1^* + q_2^*))(q_1^* - q_2^*)$. Now $P_1(q_2^*) \geq P_2(q_2^*) = q_2^*$. Thus $q_1^* - q_2^* \geq q_1^* - P_1(q_2^*)$. So $B_1(\frac{1}{2}(q_1^* + q_2^*))(q_1^* - q_2^*) \leq (q_1^* - q_2^*)$. Since each polynomial in $P(x,y)$ has degree no more than 2, each entry of $B_1(x)$ is a polynomial of degree no more than 1 in both x and in the entries of y_1 when these are treated as variables. In other words, each entry of $B_1(x)$ can be expressed in the form $(\sum_i c_i x_i) + (\sum_j c'_j y_j) + c''$, where c_i, c'_j, and c'' are all non-negative coefficients and constants of $P(x,y)$ (possibly multiplied by 2 in the case where the term of $P(x,y)$ they originate from is of the form cx_r^2) for all indices i and j. So for any i,j $B_1(\frac{1}{2}q_1^*)_{ij} \geq \frac{1}{2}B_1(q_1^*)_{ij}$. Also, since $q_2^* \leq q_1^*$ and $q_1^* > 0$, we have $B_1(\frac{1}{2}(q_1^* + q_2^*)) \geq B_1(\frac{1}{2}q_1^*)$, and the matrices $B_1(\frac{1}{2}(q_1^* + q_2^*))$ and $B_1(\frac{1}{2}q_1^*)$ are both irreducible. Also, both these matrices have non-zero entries $\geq \alpha$, because the coefficients c_i, c'_j, and c'' are all $\geq c_{\min}$, and the entries of $\frac{1}{2}q_1^*$ and $\frac{1}{2}(q_1^* + q_2^*)$ are both $\geq \frac{1}{2}q_{\min}^*$. Now, Lemma 5.9, applied to matrix $A = B_1(\frac{1}{2}(q_1^* + q_2^*))$ and vector $u = (q_1^* - q_2^*)$, yields that

$$\frac{(q_1^* - q_2^*)_{\min}}{(q_1^* - q_2^*)_{\max}} \geq \alpha^n \tag{5.5}$$

In particular, we have thus also shown that if $q_2^* \neq q_1^*$ then:

$$q_2^* < q_1^* \tag{5.6}$$

Now suppose that $B_1(x)$ is not independent of x. Since $q_2^* < q_1^*$, there is some entry of $B_1(\frac{1}{2}(q_1^* + q_2^*))$, say $B_1(\frac{1}{2}(q_1^* + q_2^*))_{ij}$, which is strictly smaller than that of $B_1(q_1^*)_{ij}$. The entry $B_1(x)_{ij}$ must be of the form $(\sum_i c_i x_i) + (\sum_j c'_j y_j) + c''$, where for some k, $c_k > 0$ so that the term $c_k x_k$ depends on x_k. We must therefore have $B_1(q_1^*)_{ij} - (B(\frac{1}{2}(q_1^* + q_2^*))_{ij} \geq c_{\min}\frac{1}{2}(q_{\mathbf{1}}^* - q_2^*)_k$, for some indices i, j, k. From inequality (5.5) we know that $\frac{(q_{\mathbf{1}}^* - q_2^*)_k}{(q_{\mathbf{1}}^* - q_2^*)_{\max}} \geq \alpha^n$, for all indices k. Thus, since $(q_{\mathbf{1}}^* - q_2^*)_{\max} = \|q_1^* - q_2^*\|_\infty$, we have

$$\begin{aligned} B_1(q_1^*)_{ij} - B_1(\frac{1}{2}(q_1^* + q_2^*))_{ij} &\geq c_{\min}\frac{1}{2}(q_{\mathbf{1}}^* - q_2^*)_k \\ &\geq c_{\min}\frac{1}{2}\alpha^n \|q_1^* - q_2^*\|_\infty \\ &\geq \alpha^{n+1}\frac{1}{2}\|q_1^* - q_2^*\|_\infty \end{aligned} \tag{5.7}$$

Since $q_2^* < q_1^*$, $\frac{1}{2}(q_1^* + q_2^*) < q_1^*$. This combined with Lemma 5.7 together imply that $\rho(B_1(\frac{1}{2}(q_1^* + q_2^*))) < 1$, and thus that $(I - B_1(\frac{1}{2}(q_1^* + q_2^*)))^{-1}$ exists and that $(I -$

$B_1(\frac{1}{2}(q_1^*+q_2^*)))^{-1} = \sum_{i=0}^{\infty} B_1(\frac{1}{2}(q_1^*+q_2^*)))^i \geq 0$. Now we need the following result from [EWY10]:

Lemma 5.11. *(Lemma 18 from [EWY10]) Let $A \in \mathbb{R}_{\geq 0}^{n\times n}$ and $b \in \mathbb{R}_{\geq 0}^n$ such that: $(I-A)^{-1} = \sum_{k=0}^{\infty} A^k$, $(I-A)^{-1}b \leq \mathbf{1}$, and A is an irreducible nonnegative matrix whose smallest nonzero entry is $c > 0$, and $b \neq 0$ and $p > 0$ is the largest entry of b. Then $\|(I-A)^{-1}\|_\infty \leq \frac{n}{pc^n}$.*

We will take $A = B_1(\frac{1}{2}(q_1^*+q_2^*))$ and $b = (I - B_1(\frac{1}{2}(q_1^*+q_2^*)))v$ in this Lemma (recall that v is the normalized Perron vector of $B_1(q_1^*)$, such that $v_{\max} = 1$). We know that $(I-B_1(q_1^*))v \geq 0$. So

$$b \geq (B_1(q_1^*) - B_1(\frac{1}{2}(q_1^*+q_2^*)))v \geq \mathbf{0} \tag{5.8}$$

Inequality (5.7) gives us a lower bound for a single entry of the non-negative matrix $(B_1(q_1^*) - B_1(\frac{1}{2}(q_1^*+q_2^*)))$, namely the (i,j)'th entry. In $(B_1(q_1^*) - B_1(\frac{1}{2}(q_1^*+q_2^*)))v$ this (i,j)'th entry is multiplied by a coordinate of v, which is at least $v_{\min}$. Thus, combining inequalities (5.7) and (5.8), we have $\|b\|_\infty \geq \alpha^{n+1}\frac{1}{2}\|q_1^*-q_2^*\|_\infty v_{\min}$. From Corollary 5.10 we have that $v_{\min} \geq \alpha^n$. So $b \geq 0$ and $\|b\|_\infty \geq \alpha^{2n+1}\frac{1}{2}\|q_1^*-q_2^*\|_\infty$. Now, by definition, $(I-B_1(\frac{1}{2}(q_1^*+q_2^*)))^{-1}b = v \leq \mathbf{1}$. Since the smallest non-zero entry of $A = B_1(\frac{1}{2}(q_1^*+q_2^*))$ is at least α, and since $\|b\|_\infty \geq \alpha^{2n+1}\frac{1}{2}\|q_1^*-q_2^*\|_\infty$, Lemma 5.11 now gives that

$$\|(I-B_1(\frac{1}{2}(q_1^*+q_2^*)))^{-1}\|_\infty \leq \frac{2n}{\alpha^{3n+1}\|q_1^*-q_2^*\|_\infty} \tag{5.9}$$

Next suppose that $B_1(x)$ is independent of x (i.e., $P_1(x)$ consists of linear or constant polynomials in x). We can thus write it as B_1, a constant, irreducible Jacobian matrix of $P_1(x)$, where the MPS $x = P_1(x)$ has an LFP $q_1^* > 0$. It must therefore be the case that $\rho(B_1) < 1$, because we already know from Lemma 5.7 that for all z such that $0 \leq z < q_1^*$, we have $\rho(B_1(z)) < 1$, but $B_1(z)$ is independent of z, because B_1 is a constant matrix.

Let us apply Lemma 3.3, with $a = q_1^*$, $b = 0$, and $P_1(x)$ in place of $P(x)$. We get $(B_1)\cdot(q_1^*-0) = P_1(q_1^*) - P(0)$. Multiplying both sides of this equation by -1 and then adding q_1^* to both sides, we get $(I-B_1)q_1^* = P_1(0)$, and thus $q_1^* = (I-B_1)^{-1}P_1(0)$. Since $q_1^* > 0$, we must have that $P_1(0) \neq 0$. But $P_1(0) \geq 0$. Indeed, $\|P_1(0)\|_\infty \geq c_{\min}\min\{1, y_{\min}^2\} \geq \alpha^2$. The smallest non-zero entry of B_1 is at least $c_{\min}\cdot\min\{1, y_{\min}\} \geq \alpha$. We now apply Lemma 5.11 to $A := B_1$ and $b := P_1(0)$, where we note that $(I-B_1)^{-1}P_1(0) = q_1^* \leq \mathbf{1}$. Lemma 5.11 thus gives:

$$\|(I - B_1(\frac{1}{2}(q_1^* + q_2^*)))^{-1}\|_\infty \leq n\alpha^{-(n+2)} \tag{5.10}$$

Since $\|q_1^* - q_2^*\|_\infty \leq 1$ ($q_1^* \leq \mathbf{1}$ and $q_2^* \geq 0$), and $0 < \alpha \leq 1$, and since $n \geq 1$, the upper bound (5.9) for the nonlinear case is worse than the upper bound (5.10) for the linear case, so the upper bound (5.9) holds in all cases.

We have shown that $(I - B_1(\frac{1}{2}(q_1^* + q_2^*)))$ is non-singular, since $\rho(B_1(\frac{1}{2}(q_1^* + q_2^*))) < 1$. Equation (5.4) is thus valid, and taking norms of (5.4) yields:

$$\|q_1^* - q_2^*\|_\infty \leq \|(I - B_1(\frac{1}{2}(q_1^* + q_2^*)))^{-1}\|_\infty \|P_1(q_2^*) - P_2(q_2^*)\|_\infty \tag{5.11}$$

Inserting our upper bound (5.9) for $\|(I - B_1(\frac{1}{2}(q_1^* + q_2^*)))^{-1}\|_\infty$ gives:

$$\|q_1^* - q_2^*\|_\infty \leq \frac{2n}{\alpha^{3n+1}\|q_1^* - q_2^*\|_\infty} \|P_1(q_2^*) - P_2(q_2^*)\|_\infty$$

We now move the $\|q_1^* - q_2^*\|_\infty$ terms to the left and take square roots to obtain:

$$\|q_1^* - q_2^*\|_\infty \leq \sqrt{2n\alpha^{-(3n+1)}\|P_1(q_2^*) - P_2(q_2^*)\|_\infty} \tag{5.12}$$

Lemma 5.12. *If* $0 \leq x \leq 1$, *then* $\|P_1(x) - P_2(x)\|_\infty \leq 2\|P(\mathbf{1}, \mathbf{1})\|_\infty \|y_1 - y_2\|_\infty$.

Proof. Since each entry of $P(x, y)$ is a quadratic polynomial, for each $b \in \{1, 2\}$ and each $d \in \{1, \ldots, n\}$, the d'th coordinate, $(P_b(x))_d$, of $P_b(x) = P(x, y_b)$ has the form

$$\sum_{i,j} a_{d,i,j} x_i x_j + \sum_{i,j} c_{d,i,j} y_{b,i} y_{b,j} + \sum_{i,j} c'_{d,i,j} x_i y_{b,j} + \sum_k a'_{d,k} x_k + \sum_k c''_{d,k} y_{b,k} + c'''_d$$

where $y_{b,j}$ refers to the j'th coordinate of the m-vector y_b, and where all the coefficients $a_{d,i,j}$, $c_{d,i,j}$, $c'_{d,i,j}$, $c''_{d,k}$ and c'''_d, are non-negative. Also, recall $0 < y_1 \leq \mathbf{1}$ and $0 \leq y_2 \leq y_1$. Thus,

$\|P_1(x) - P_2(x)\|_\infty$

$$\begin{aligned}
&= \max_d \sum_{i,j} c_{d,i,j}(y_{1,i}y_{1,j} - y_{2,i}y_{2,j}) + \sum_{i,j} c'_{d,i,j} x_i (y_{1,j} - y_{2,j}) + \sum_k c''_{d,k}(y_{1,k} - y_{2,k}) \\
&\leq \max_d \sum_{i,j} c_{d,i,j}((y_{1,i} - y_{2,i}) + (y_{1,j} - y_{2,j})) + \sum_{i,j} c'_{d,i,j}(y_{1,j} - y_{2,j}) + \sum_k c''_{d,k}(y_{1,k} - y_{2,k}) \\
&\leq \max_d \sum_{i,j} 2 \cdot c_{d,i,j} \cdot \|y_1 - y_2\|_\infty + \sum_{i,j} c'_{d,i,j} \cdot \|y_1 - y_2\|_\infty + \sum_k c''_{d,k} \|y_1 - y_2\|_\infty \\
&= (\max_d \sum_{i,j} 2c_{d,i,j} + c'_{d,i,j} + c''_{d,k}) \cdot \|y_1 - y_2\|_\infty \\
&\leq 2\|P(1,1)\|_\infty \|y_1 - y_2\|_\infty
\end{aligned}$$

□

Combining (5.12) and Lemma 5.12, we have,

$$\|q_1^* - q_2^*\|_\infty \leq \sqrt{4n\alpha^{-(3n+1)}\|P(\mathbf{1},\mathbf{1})\|_\infty\|y_2 - y_1\|_\infty}$$

which completes the proof of the first inequality of part (**1.**) of Theorem 5.2.

We show the second inequality (5.1) of part (**1.**) in the next lemma.

Lemma 5.13. *If $B_1(x)$ is a constant matrix, i.e. $x = P_1(x)$ is linear,*

$$\|q_1^* - q_2^*\|_\infty \leq 2n\alpha^{-(n+2)}\|P(\mathbf{1},\mathbf{1})\|_\infty\|y_1 - y_2\|_\infty$$

Proof. If $q_1^* = q_2^*$, then the result is trivial. So we assume that $q_1^* \neq q_2^*$. In the proof of the first inequality, under the assumption that $B_1(x)$ is a constant, we obtained equation (5.10). So, under the assumptions of this Lemma, equation (5.10) is valid, and we can substitute the bound (5.10) into the equation (5.11) instead. This gives

$$\|q_1^* - q_2^*\|_\infty \leq n\alpha^{-(n+2)}\|P_1(q_2^*) - P_2(q_2^*)\|_\infty$$

Again, Lemma 5.12 gives a bound on $\|P_1(q_2^*) - P_2(q_2^*)\|_\infty$. Substituting this gives:

$$\|q_1^* - q_2^*\|_\infty \leq n\alpha^{-(n+2)}2\|P(\mathbf{1},\mathbf{1})\|_\infty\|y_1 - y_2\|_\infty$$

□

We will next establish part (**2.**) of Theorem 5.2. Let us first prove that, starting from $x^{[0]} := 0$, all the iterations of R-NM, applied to $x = P_2(x)$ are defined.

We firstly note that if $0 \leq x^{[k]} \leq q_2^*$ and $\rho(B_2(x^{[k]})) < 1$, then $\mathcal{N}_{P_2}(x^{[k]})$ is well-defined and $0 \leq x^{[k+1]} \leq q_2^*$. If $\rho(B_2(x^{[k]})) < 1$, then by Lemma 5.5, $(I - B_2(x^{[k]}))$ is non-singular and so $\mathcal{N}_{P_2}(x^{[k]})$ is well-defined. Lemma 5.5 also gives that $(I - B_2(x^{[k]}))^{-1} = \sum_{i=0}^{\infty} B_2(x^{[k]})^i \geq 0$. Lemma 3.4 yields that:

$$q_2^* - \mathcal{N}_{P_2}(x^{[k]}) = (I - B_2(x^{[k]}))^{-1}\frac{B_2(q_2^*) - B_2(x^{[k]})}{2}(q_2^* - x^{[k]})$$

Note that $(q_2^* - x^{[k]}) \geq 0$, thus that $B_2(q_2^*) - B_2(x^{[k]}) \geq 0$, and we have just shown that $(I - B_2(x^{[k]}))^{-1} \geq 0$. So all the terms on the right of the above equation are non-negative, and thus $q_2^* - \mathcal{N}_{P_2}(x^{[k]}) \geq 0$. That is $q_2^* \geq \mathcal{N}_{P_2}(x^{[k]})$. $x^{[k+1]}$ is defined by rounding down $\mathcal{N}_{P_2}(x^{[k]})$ and maintaining non-negativity, thus for all coordinates i, either $x_i^{[k+1]} = 0$, in which case trivially we have $x_i^{[k+1]} = 0 \leq (q_2^*)_i$, or else $0 \leq x_i^{[k+1]} \leq \mathcal{N}_{P_2}(x^{[k]})_i \leq (q_2^*)_i$. Thus $x^{[k+1]} \leq q_2^*$.

What is still missing is to show that $\rho(B_2(x^{[k+1]})) < 1$. If we can show this then by an easy induction, for all k, $\mathcal{N}_{P_2}(x^{[k]})$ is well-defined and $0 \leq x^{[k]} \leq q_2^*$. We will prove $\rho(B_2(x^{[k+1]})) < 1$ by considering separately the cases where $P_1(x)$ contains nonlinear or only linear polynomials.

Lemma 5.14. *If $x = P(x)$ is a strongly-connected quadratic MPS with n variables, with LFP $q^* > \mathbf{0}$, and there is some nonlinear quadratic term in some polynomial $P_i(x)$, then if $0 \leq z < q^*$, then $\mathcal{N}_P(z)$ is defined and $\mathcal{N}_P(z) < q^*$.*

Proof. Lemma 5.7 tells us that $\rho(B(q^*)) \leq 1$. Nonlinearity of $P(x)$ means that $B(x)$ does depend on x. That is, some entry of $B(x)$ contains a term of the form cx_i for some x_i with $c > 0$. So $B(z) \neq B(q^*)$, and $B(z) \leq B(q^*)$ since B is monotone. Since $x = P(x)$ is strongly-connected and $q^* > \mathbf{0}$, Lemma 5.7 yields that $\rho(B(z)) < 1$. By Lemma 5.5, $(I - B(z))$ is non-singular and so the Newton iterate $\mathcal{N}_P(z)$ is well-defined. Consider the equation given by Lemma 3.4:

$$q^* - \mathcal{N}_P(z) = (I - B(z))^{-1} \frac{B(q^*) - B(z)}{2} (q^* - z)$$

We know that $q^* - z > 0$, and thus $B(q^*) - B(z) \geq 0$. Since $\rho(B(z)) < 1$, by Lemma 5.5, $(I - B(z))^{-1} = \sum_{k=0}^{\infty} B(z)^k \geq 0$. This and Lemma 3.4 is already enough to yield that $q^* - \mathcal{N}_P(z) \geq 0$, and we just need to show that this is a strict inequality.

We first show that if $P_i(x)$ contains a term of degree 2, then $(\frac{B(q^*)-B(z)}{2}(q^* - z))_i > 0$. This term of degree 2 must be of the form cx_jx_k for some j,k. Then $B(x)_{i,j}$ has a term cx_k with $c > 0$ and so $(B(q^*) - B(z))_{i,j} \geq c(q^* - z)_k$. But then $(\frac{B(q^*)-B(z)}{2}(q^* - z))_i \geq c(q^* - z)_k(q^* - z)_j > 0$.

Now we will show that for all $i \in \{1, \ldots, n\}$, $(q - \mathcal{N}_P(z))_i > 0$. If $P_i(x)$ contains a term of degree 2, then we have just shown that $(\frac{B(q^*)-B(z)}{2}(q^* - z))_i > 0$. But $(I - B(z))^{-1} = \sum_{k=0}^{\infty} B(z)^k \geq I$. So $(q - \mathcal{N}_P(z))_i \geq (\frac{B(q^*)-B(z)}{2}(q^* - z))_i > 0$. If $P_i(x)$ does not contain a term of degree 2, there must be some other x_j with $P_j(x)$ containing a term of degree 2 and, since $x = P(x)$ is strongly-connected, x_i depends on x_j, possibly indirectly. That is, there is a sequence of variables $i_0, i_1, \ldots, i_l$ with $l < n$, $i_0 = i, i_l = j$, and for each $0 < m \leq l, x_{i_m}$ appears in a term of $P(x)_{i_{m-1}}$. Let k be the the least integer such that $P(x)_{i_k}$ contains a term of degree 2. Then if $0 < m \leq k$, x_{i_m} appears in a degree 1 term in $P(x)_{i_{m-1}}$, that is one of the form $c_m x_m$ with $c_m > 0$. So $B(x)_{i_{m-1},i_m}$ contains the constant term $c_m > 0$. So $B(z)_{i_{m-1},i_m} \geq c_m > 0$. So $B^k(z)_{i,i_k} \geq \prod_{m=0}^{k-1} B(z)_{i_m,i_{m+1}} \geq \prod_{m=0}^{k-1} c_m > 0$. Since $P(x)_{i_k}$ contains a term of degree 2, from above $(\frac{B(q^*)-B(z)}{2}(q^* - z))_{i_k} > 0$. So $(B^k(z)\frac{B(q^*)-B(z)}{2}(q^* - z))_i > 0$. But $q^* - \mathcal{N}_P(z) = (I -$

$B(z))^{-1}\frac{B(q^*)-B(z)}{2}(q^*-z) = (\sum_{m=0}^{\infty} B^m(z))\frac{B(q^*)-B(z)}{2}(q^*-z) \geq B^k(z)\frac{B(q^*)-B(z)}{2}(q^*-z)$. So $(q^* - \mathcal{N}_P(z))_i > 0$ for all i, as required. □

We will only actually need to apply Lemma 5.14 in the case when $q_2^* = q_1^*$ and $x = P_1(x)$ is nonlinear.

Suppose that $q_1^* = q_2^*$ and some polynomial in $P_1(x)$ is nonlinear in x. We claim that then $P_1(x) \equiv P_2(x)$. That is, for all those variables in y, say $(y)_j$, that actually appear in some polynomials in $P(x,y)$, it must be the case that $(y_1)_j = (y_2)_j$. Otherwise, if there is some variable $(y)_j$ with $(y_2)_j < (y_1)_j$ such that $(y)_j$ appears in the polynomial $(P(x,y))_i$, then $(P_2(q_1^*))_i = (P(q_1^*,y_2))_i < P(q_1^*,y_1))_i = (q_1^*)_i$, so q_1^* is not a fixed point of $P_2(x)$, contradicting that $q_1^* = q_2^*$. Thus if $x = P_1(x)$ is nonlinear and $q_1^* = q_2^*$ then $x = P_2(x)$ is also nonlinear and $q_2^* = q_1^* > 0$, so we can use Lemma 5.14, which shows that if $0 \leq x^{[k]} < q_2^*$, then $\mathcal{N}_{P_2}(x^{[k]}) < q_2^*$ and so $0 \leq x^{[k+1]} < q_2^* \leq q_1^*$. Since $x^{[k+1]} < q_1^*$, we have $\rho(B_1(x^{[k+1]})) < 1$. Since $B_2(x^{[k+1]}) \leq B_1(x^{[k+1]})$, we also have $\rho(B_2(x^{[k+1]})) < 1$.

This leaves us with two cases remaining to show that all Newton iterates exist: first, the case where $x = P_1(x)$ is linear or constant, and second, the case where $x = P_1(x)$ is nonlinear and $q_2^* \neq q_1^*$. Recall that it is sufficient to show that $\rho(B_2(x^{[k]})) < 1$ for all iterates in order to show that all R-NM iterates exist. It thus suffices to show that in these cases for any $0 \leq z \leq q_2^*$, $\rho(B_2(z)) < 1$.

For the first case, suppose that $x = P_1(x)$ is linear. Then $B_1(x)$ is a constant matrix. Thus $B_1(z) = B_1(0)$ for all $0 \leq z$. But Lemma 5.7 tells us that, since $\mathbf{0} < q_1^*$, $\rho(B_1(0)) < 1$. Thus $\rho(B_1(z)) < 1$ for all $0 \leq z \leq q_2^*$. Since $0 \leq B_2(z) \leq B_1(z)$, we have $\rho(B_2(z)) < 1$ for all $0 \leq z \leq q_2^*$.

For the second case, suppose that $q_2^* \neq q_1^*$ and that $x = P_1(x)$ is nonlinear, and thus $B_1(x)$ depends on x. Then we have previously argued that $q_2^* < q_1^*$ (see inequality (5.6)). But then $B_1(q_2^*) \neq B_1(q_1^*)$. For any $0 \leq z \leq q_2^*$, $B_2(z) \leq B_2(q_2^*) \leq B_1(q_2^*) \leq B_1(q_1^*)$ but because $B_1(q_2^*) \neq B_1(q_1^*)$, we have $B_2(z) \neq B_1(q_1^*)$. But $B_1(q^*)$ is irreducible, and Lemma 5.4 then tells us that $\rho(B_2(z)) < \rho(B_1(q_1^*))$. But we know, by Corollary 5.8, that $\rho(B_1(q_1^*) \leq 1$. So $\rho(B_2(z)) < 1$.

Thus the R-NM iterations applied to $x = P_2(x)$ are defined in all cases, and yield iterates $0 \leq x^{[k]} \leq q^*$, for all $k \geq 0$.

We can now prove the upper bound on the rate of convergence for R-NM applied to $x = P_2(x)$.

Lemma 5.15. *Suppose an MPS, $x = P(x)$, with n variables has LFP $0 \leq q^* \leq 1$, and for some n-vector $v > 0$ we have $B(q^*)v \leq v$. Suppose we perform $g \geq h-1$ iterations* *of R-NM with parameter $h \geq 2 + \lceil \log \frac{v_{\max}}{v_{\min} \cdot \varepsilon} \rceil$ on the MPS $x = P(x)$, and suppose that for all $k \geq 0$, every iteration $x^{[k]}$ is defined and $0 \leq x^{[k]} \leq q^*$. Then $\|q^* - x^{[g]}\|_\infty \leq \varepsilon$.*

Proof. By induction on k, we claim that $\forall k \geq 0$, $q^* - x^{[k]} \leq (2^{-k} + 2^{-h+1}) \frac{1}{v_{\min}} v$. Note that this would indeed yield the Lemma: for all k $0 \leq x^{[k]} \leq q^*$. and the claim would yield $q^* - x^{[g]} \leq (2^{-h+1} + 2^{-h+1}) \frac{1}{v_{\min}} v \leq 2^{-\log \frac{v_{\max}}{v_{\min} \cdot \varepsilon}} \frac{1}{v_{\min}} v = \varepsilon \frac{1}{v_{\max}} v \leq \varepsilon \mathbf{1}$.

It remains to prove by induction on $k \geq 0$ that $q^* - x^{[k]} \leq (2^{-k} + 2^{-h+1}) \frac{1}{v_{\min}} v$. This is true for $k = 0$, because $q^* \geq 0 = x^{[0]}$, and $q^* - x^{[0]} = q^* \leq 1 \leq \frac{1}{v_{\min}} v$.

Lemma 3.9 then gives that $q^* - \mathcal{N}_P(x^{[k]}) \leq (2^{-(k+1)} + 2^{-h}) \frac{1}{v_{\min}} v$. Now, by definition of $x^{[k+1]}$, $\mathcal{N}_P(x^{[k]}) - x^{[k+1]} \leq 2^{-h} \mathbf{1} \leq 2^{-h} \frac{1}{v_{\min}} v$. So $q^* - x^{[k+1]} \leq (2^{-(k+1)} + 2^{-h+1}) \frac{1}{v_{\min}} v$ as required. Thus $q^* - x^{[h-1]} \leq 2^{-h+2} \frac{1}{v_{\min}} v \leq \frac{\varepsilon}{v_{\max}} v \leq \varepsilon \mathbf{1}$. □

To use Lemma 5.15 to get a bound on using R-NM on $x = P_2(x)$ to compute q_2^*, note that because $0 \leq B_2(q_2^*) \leq B_1(q_1^*)$, the Perron vector $v > 0$ of $B_1(q_1^*)$, which satisfies $B_1(q_1^*) v \leq v$, must also satisfy $B_2(q_2^*) v \leq v$.

Thus, we just need to perform $g \geq h - 1$ iterations of R-NM on $x = P_2(x)$, with parameter $h \geq 2 + \log \frac{v_{\max}}{v_{\min} \varepsilon} \geq 2 + \log \alpha^{-n} \varepsilon^{-1}$ in order to obtain that $\|q_2^* - x^{[h-1]}\|_\infty \leq \varepsilon$. This completes the proof of Theorem 5.2. □

Corollary 5.16. *Let $x = P(x)$ be a strongly connected MPS with n variables, and with LFP q^* where $\mathbf{0} < q^* \leq 1$. Let $\alpha = \min\{1, c_{\min}\} \frac{1}{2} q^*_{\min}$, where $c_{\min}$ is the smallest non-zero constant or coefficient of any monomial in $P(x)$.*

Then for all $0 < \varepsilon < 1$, if we use $g \geq h - 1$ iterations of R-NM with parameter $h \geq \lceil 2 + n \log \frac{1}{\alpha} + \log \frac{1}{\varepsilon} \rceil$ applied to the MPS, $x = P(x)$, starting at $x^{[0]} := \mathbf{0}$, then the iterations are all defined, and $\|q^* - x^{[g]}\|_\infty \leq \varepsilon$.

Proof. This is just a trivial application of Theorem 5.2, part **2.**, where we define y to be a dummy variable of dimension $m = 1$, and we define $y_1 = y_2 = y_{\min} = 1$, and where we define the n-vector of monotone polynomials $P(x, y)$, by replacing all constant terms $c > 0$ in every polynomial in $P(x)$ by cy. In this case, note that $P_1(x) = P_2(x) = P(x)$, and that since $y_{\min} = 1$, the α defined in the statement of this corollary is the same α as in Theorem 5.2. □

5.3 General Monotone Polynomial Systems

In this section, we use the rounded-down decomposed Newton's method (R-DNM), to compute the LFP q^* of general MPSs. First we consider the case where $\mathbf{0} < q^* \leq 1$:

Theorem 5.17. *For all* ε*, where* $0 < \varepsilon < 1$*, if* $x = P(x)$ *is an MPS with LFP solution* $\mathbf{0} < q^* \leq 1$*, with* $q^*_{\min} = \min_i q^*_i$*, and the minimum non-zero coefficient or constant in* $P(x)$ *is* $c_{\min}$*, then rounded down decomposed Newton's method (R-DNM) with parameter*

$$h \geq \left\lceil 3 + 2^f \cdot (\, \log(\frac{1}{\varepsilon}) + d \cdot (\log(\alpha^{-(4n+1)}) + \log(16n) + \log(\|P(\mathbf{1})\|_\infty))\,)\right\rceil$$

using $g \geq h - 1$ *iterations for every nonlinear SCC (and 1 iteration for linear SCC), gives an approximation* $\tilde{q}$ *to* q^* *with* $\tilde{q} \leq q^*$ *and such that* $\|q^* - \tilde{q}\|_\infty \leq \varepsilon$*.*

Here d *denotes the maximum depth of SCCs in the DAG* H_P *of SCCs of the MPS* $x = P(x)$*,* f *is the nonlinear depth, and* $\alpha = \min\{1, c_{\min}\} \cdot \frac{1}{2} q^*_{\min}$*.*

Before proving the theorem, let us note that we can obtain worst-case expressions for the needed number of iterations $g = h - 1$, and the needed rounding parameter h, in terms of only $f \leq d \leq n \leq |P|$, and ε, by noting that $\log(\|P(\mathbf{1})\|_\infty) \leq |P|$, and by appealing to Theorem 5.1 to remove references to $q^*_{\min}$ in the bounds. Noting that $c_{\min} \geq 2^{-|P|}$, these tell us that $\min\{1, c_{\min}\}\frac{1}{2}q^*_{\min} \geq 2^{-|P|2^n - 1}$. Substituting, we obtain that any:

$$g \geq \left\lceil 2 + 2^f \cdot (\, \log(\frac{1}{\varepsilon}) + d \cdot (|P|2^n(4n+1) + (4n+1) + \log(16n) + |P|)\,)\right\rceil \tag{5.13}$$

iterations suffice in the worst case, with rounding parameter $h = g + 1$. Thus, for $i = \log(1/\varepsilon)$ bits of precision, $g = k_P + c_P \cdot i$ iterations suffice, where $c_P = 2^f$ and $k_P = O(2^f 2^n nd|P|)$, with tame constants in the big-O.

Proof of Theorem 5.17. For every SCC S, its *height* h_S (resp. *nonlinear height* f_S) is the maximum over all paths of the DAG H_P of SCCs starting at S, of the number of SCCs (resp. nonlinear SCCs) on the path. We show by induction on the height h_S of each SCC S that $\|q^*_S - \tilde{q}_S\|_\infty \leq \beta^{h_S}\delta^{2^{-f_S}}$ where $\beta = 16n\alpha^{-(3n+1)}\|P(\mathbf{1})\|_\infty$ and $\delta = (\frac{\varepsilon}{\beta^d})^{2^f}$. Note that since $n \geq 1$, $\varepsilon < 1$, and $\alpha \leq c_{\min}$, we have $\beta \geq 1$ and $\delta \leq 1$, and thus also $\delta \leq \sqrt{\delta}$.

Let us first check that this would imply the theorem. For all SCCs, S, we have $1 \leq h_S \leq d$ and $0 \leq f_S \leq f$, and thus $\|q^*_S - \tilde{q}_S\|_\infty \leq \beta^{h_S}\delta^{2^{-f_S}} \leq \beta^d\delta^{2^{-f}} = \beta^d(\frac{\varepsilon}{\beta^d}) = \varepsilon$.

We note that h is related to δ by the following:

$$h \geq 2 + n\log\frac{1}{\alpha} + \log\frac{2}{\delta} \tag{5.14}$$

This is because

$$\log\frac{2}{\delta} = 1 + \log\frac{1}{\delta} = 1 + 2^f(\log\frac{1}{\varepsilon} + d\log\beta) = 1 + 2^f(\log(\frac{1}{\varepsilon}) + d\log(16n\alpha^{-3n+1}\|P(\mathbf{1})\|_\infty))$$

. Note that this inequality holds also for any subsystem of $x = P(x)$ induced by a SCC S and its successors $D(S)$ because the parameters n and $1/\alpha$ for a subsystem are no larger than those for the whole system.

We now prove by induction on h_S that $\|q^*_S - \tilde{q}_S\|_\infty \leq \beta^{h_S}\delta 2^{-f_S}$.

In the base case, $h_S = 1$, we have a strongly connected MPS $x_S = P_S(x)$. If S is linear, we solve the linear system exactly and then round down to a multiple of 2^{-h}. Then $f_S = 0$, and we have to show $\|q^*_S - \tilde{q}_S\|_\infty \leq \beta^{h_S}\delta 2^{-f_S} = \beta\delta$. But $\|q^*_S - \tilde{q}_S\|_\infty \leq 2^{-h} \leq \frac{\delta}{2} \leq \beta\delta$.

For the base case where S in nonlinear, equation 5.14 and Corollary 5.16 imply that $\|q^*_S - \tilde{q}_S\|_\infty \leq \frac{\delta}{2}$, which implies the claim since $\delta \leq 1$ and $\beta \geq 1$, hence $\frac{\delta}{2} \leq \beta^{h_S}\delta 2^{-f_S} = \beta^1\delta 2^{-1}$.

Inductively, consider an SCC S with $h_S > 1$. Then S depends only on SCCs with height at most $h_S - 1$. If S is linear, it depends on SCCs of nonlinear depth at most $f_{D(S)} = f_S$, whereas if S is nonlinear, it depends on SCCs of nonlinear depth at most $f_{D(S)} = f_S - 1$. We can assume by inductive hypothesis that $\|q^*_{D(S)} - \tilde{q}_{D(S)}\|_\infty \leq \beta^{h_S-1}\delta 2^{-f_{D(S)}}$. Take q'_S to be the LFP of $x_S = P_S(x_S, \tilde{q}_{D(S)})$.

Suppose $x_S = P_S(x_S, q^*_{D(S)})$ is linear in x_S. Then Theorem 5.2 with $y_1 := q^*_{D(S)}$ and $y_2 := \tilde{q}_{D(S)}$, yields

$$\|q^*_S - q'_S\|_\infty \leq 2n_S\alpha^{-(n_S+2)}\|P(\mathbf{1},\mathbf{1})\|_\infty\|q^*_{D(S)} - \tilde{q}_{D(S)}\|_\infty$$

But $2n_S\alpha^{-(n_S+2)}\|P(\mathbf{1},\mathbf{1})\|_\infty \leq \frac{\beta}{2}$, so $\|q^*_S - q'_S\|_\infty \leq \frac{\beta}{2}\|q^*_{D(S)} - \tilde{q}_{D(S)}\|_\infty \leq \frac{\beta}{2}\beta^{h_S-1}\delta 2^{-f_S} = \frac{1}{2}\beta^{h_S}\delta 2^{-f_S}$. Since $\|q'_S - \tilde{q}_S\|_\infty \leq 2^{-h} \leq \frac{\delta}{2} \leq \frac{1}{2}\beta^{h_S}\delta 2^{-f_S}$, it follows that $\|q^*_S - \tilde{q}_S\|_\infty \leq \beta^{h_S}\delta 2^{-f_S}$.

Suppose that $x_S = P_S(x_S, q^*_{D(S)})$ is nonlinear in x_S. Theorem 5.2, with $y_1 := q^*_{D(S)}$ and $y_2 := \tilde{q}_{D(S)}$, yields that

$$\|q^*_S - q'_S\|_\infty \leq \sqrt{4n\alpha^{-(3n+1)}\|P(\mathbf{1})\|_\infty\|q^*_{D(S)} - (\tilde{q})_{D(S)}\|_\infty} \tag{5.15}$$

Note that the α from Theorem 5.2 is indeed the same or better (i.e., bigger) than the α in this Theorem, because $y_{\min} = (q^*_{D(S)})_{\min} \geq q^*_{\min}$ and $(q^*_S)_{\min} \geq q^*_{\min}$. Rewriting (5.15) in terms of β, we have $\|q^*_S - q'_S\|_\infty \leq \sqrt{\frac{1}{4}\beta\|q^*_{D(S)} - (\tilde{q})_{D(S)}\|_\infty}$. By inductive assumption, $\|q^*_{D(S)} - \tilde{q}_{D(S)}\|_\infty \leq \beta^{h_S-1}\delta 2^{-f_S+1}$, and thus $\|q^*_S - q'_S\|_\infty \leq \sqrt{\frac{1}{4}\beta^{h_S}\delta 2^{1-f_S}} \leq \frac{1}{2}\beta^{h_S}\delta 2^{-f_S}$. Thus to show that the inductive hypothesis holds also for SCC S, it suffices to show that for the approximation $\tilde{q}_S$ we have $\|q'_S - \tilde{q}_S\|_\infty \leq \frac{1}{2}\beta^{h_S}\delta 2^{-f_S}$. But $\beta \geq 1$, $h_S \geq 1$, $2^{-f_S} \leq 1$ and $\delta \leq 1$, so $\frac{1}{2}\delta \leq \frac{1}{2}\beta^{h_S}\delta 2^{-f_S}$, so it suffices to show that $\|q'_S - \tilde{q}_S\|_\infty \leq \frac{1}{2}\delta$.

Part 2 of Theorem 5.2 tells us that we will have $\|q'_S - \tilde{q}_S\|_\infty \leq \frac{1}{2}\delta$ if $g \geq h-1$ and $h \geq 2 + n\log\frac{1}{\alpha} + \log\frac{2}{\delta}$. But we have already established this in equation (5.14), hence the claim follows. □

Next, we want to generalize Theorem 5.17 to arbitrary MPSs that have an LFP, $q^* > \mathbf{0}$, without the restriction that $\mathbf{0} < q^* \leq 1$. The next Lemma allows us to establish this by a suitable "rescaling" of any MPS which has an LFP $q^* > \mathbf{0}$. If $x = P(x)$ is an MPS and $c > 0$, we can consider the MPS $x = \frac{1}{c}P(cx)$.

Lemma 5.18. *Let $x = P(x)$ be a MPS with LFP solution q^*, and with Jacobian $B(x)$, and recall that for $z \geq 0$, $\mathcal{N}_P(z) := z + (I - B(z))^{-1}(P(z) - z)$ denotes the Newton operator applied at z on $x = P(x)$. Then:*

(i) The LFP solution of $x = \frac{1}{c}P(cx)$ is $\frac{1}{c}q^*$.

(ii) The Jacobian of $\frac{1}{c}P(cx)$ is $B(cx)$.

(iii) A Newton iteration of the "rescaled" MPS, $x = \frac{1}{c}P(cx)$, applied to the vector z is given by $\frac{1}{c}\mathcal{N}_P(cz)$.

Proof. From [EY09], we know that the value iteration sequence $P(0), P(P(0)), P(P(P(0)))\ldots P^k(0)$ converges to q^*. Now note that for the MPS $x = \frac{1}{c}P(cx)$, the value iteration sequence is $\frac{1}{c}P(0)$, $\frac{1}{c}P(c\frac{1}{c}P(0)) = \frac{1}{c}P(P(0))$, $\frac{1}{c}P(P(P(0)))\ldots$ which thus converges to $\frac{1}{c}q^*$. This establishes (i).

For (ii), note that, by the chain rule in multivariate calculus (see, e.g., [Apo74] Section 12.10), the Jacobian of $P(cx)$ is $cB(cx)$. Now (iii) follows because:

$$z + (I - B(cz))^{-1}(\frac{1}{c}P(cz) - z) = \frac{1}{c}(cz + (I - B(cz))^{-1}(P(cz) - cz)) = \frac{1}{c}\mathcal{N}_P(cz).$$

□

We use Lemma 5.18 to generalise Theorem 5.17 to MPSs with LFP q^*, where q^* does not satisfy $q^* \leq 1$.

Theorem 5.19. *If $x = Q(x)$ is an MPS with n variables, with LFP solution $q^* > \mathbf{0}$, if $c'_{\min}$ is the least positive coefficient of any monomial in $Q(x)$, then R-DNM with rounding parameter h', and using g' iterations per nonlinear SCC (and one for linear), gives an approximation $\tilde{q}$ such that $\|q^* - \tilde{q}\|_\infty \leq \varepsilon'$, where*

$$g' = 2 + \lceil\, 2^f \cdot (\log(\frac{1}{\varepsilon'}) + d \cdot (2u + \log(\alpha'^{-(4n+1)}) + \log(16n) + \log(\|Q(\mathbf{1})\|_\infty))\ )\ \rceil$$

and $h' = g' + 1 - u$, where $u = \max\{0, \lceil\log q^*_{\max}\rceil\}$, d is the maximum depth of SCCs in the DAG H_Q of SCCs of $x = Q(x)$, f is the nonlinear depth, and $\alpha' = 2^{-2u}\min\{1, c'_{\min}\}\min\{1, \frac{1}{2}q^*_{\min}\}$.

Proof. If $q^*_{\max} \leq 1$, then Theorem 5.17 gives this immediately. So we assume that $q^*_{\max} > 1$. u is chosen so that $2^u \geq q^*_{\max}$. We rescale and use Lemma 5.18 with scaling parameter $c = 2^u$. This yields the "rescaled" MPS $x = 2^{-u}Q(2^u x)$, which has LFP $p^* = 2^{-u}q^* \leq \mathbf{1}$.

So we can apply Theorem 5.17 to this rescaled MPS $x = P(x)$, where $P(x) \equiv 2^{-u}Q(2^u x)$, and letting $\varepsilon := 2^{-u}\varepsilon'$. Then Theorem 5.17 gives us the needed number of iterations g and the rounding parameter $h = g+1$, needed to obtain an approximation $\tilde{p}$ of the LFP $p^* = 2^{-u}q^*$, such that $\|\tilde{p} - p^*\|_\infty \leq \varepsilon$.

In the bounds specified for Theorem 5.17 for g and h, in place of $q^*_{\min}$ we get $p^*_{\min} = 2^{-u}q^*_{\min}$, and in place of $c_{\min}$ we get $2^{-u}c'_{\min}$. Thus α becomes the α' we have specified in the statement of this theorem. Furthermore, the $\|P(\mathbf{1})\|_\infty$ appearing in Theorem 5.17 is now $\|2^{-u}Q(2^u\mathbf{1})\|_\infty$, but it is easy to verify that for a quadratic MPS, $\|2^{-u}Q(2^u\mathbf{1})\|_\infty \leq 2^u\|Q(\mathbf{1})\|_\infty$.

Theorem 5.17 tells us that if we use R-DNM on $x = P(x)$ for g iterations per nonlinear SCC and a precision of $h = g+1$ bits, we will obtain an approximation $\tilde{p}$ to the LFP p^* of $x = P(x)$ with $\|\tilde{p} - p^*\|_\infty \leq \varepsilon$ provided that $h \geq \lceil 3 + 2^f \cdot (\log(\frac{1}{\varepsilon}) + d \cdot (\log(\alpha^{-(4n+1)}) + \log(16n) + \log(\|P(\mathbf{1})\|_\infty)))\rceil$. This condition is satisfied if we take $g = g'$ and $h = g'+1$ because:

$$\begin{aligned}
& \lceil 3 + 2^f \cdot (\log(\frac{1}{\varepsilon}) + d \cdot (\log(\alpha^{-(4n+1)}) + \log(16n) + \log(\|P(\mathbf{1})\|_\infty)))\rceil \\
\leq \;& 3 + 2^f(\log(\frac{1}{2^{-u}\varepsilon'}) + d(\log(\alpha'^{-(4n+1)}) + \log(16n) + \log(2^u\|Q(\mathbf{1})\|_\infty)))\rceil \\
= \;& 3 + 2^f(u + \log(\frac{1}{\varepsilon'}) + d(\log(\alpha'^{-(4n+1)}) + \log(16n) + u + \log(\|Q(\mathbf{1})\|_\infty)))\rceil \\
\leq \;& g' + 1 = h
\end{aligned}$$

Thus, applying R-DNM on $x = P(x)$ with parameters $g = g'$ and $h = g'+1$ yields an approximation $\tilde{p}$ to the LFP p^* of $x = P(x)$ with $\|\tilde{p} - p^*\|_\infty \leq \varepsilon$ or, in terms of the original MPS, $\|\tilde{p} - 2^{-u}q^*\|_\infty \leq 2^{-u}\varepsilon'$.

To obtain Theorem 5.19, we now show that if we apply R-DNM to $x = Q(x)$ with LFP q^*, using rounding parameter h' and using g' iterations per nonlinear SCC (where h' and g' were specified in the statement of the Theorem), we will obtain an approximation $\tilde{q}$ to q^* that satisfies $\tilde{q} = 2^u\tilde{p}$. This would then give us that $\|q^* - \tilde{q}\|_\infty = \|2^u p^* - 2^u\tilde{p}\|_\infty = 2^u\|p^* - \tilde{p}\|_\infty \leq 2^u\varepsilon = \varepsilon'$, which is what we want to prove.

Since we are using the decomposed Newton's method, we will show that $\tilde{q}_S = 2^u\tilde{p}_S$ for every SCC S by induction on the depth of the SCC S. Suppose that for the variables

$D(S)$ that S depends on (if any), we have that $\tilde{q}_{D(S)} = 2^u \tilde{p}_{D(S)}$. If we call the kth iterate of R-NM applied to $x_S = P_S(x_S, \tilde{p}_{D(S)})$ with parameter h, $x^{[k]}$ and the kth iterate of R-NM applied to $x_S = Q_S(x_S, \tilde{q}_{D(S)})$ with parameter h', $x'^{[k]}$, then we aim to show by induction on k that $x'^{[k]} = 2^u x^{[k]}$.

The base case is $x'^{[0]} = 0 = 2^u x^{[0]}$. By abuse of notation, we will call the Newton iterate of $x_S = P_S(x_S, \tilde{p}_{D(S)})$, $\mathcal{N}_P(x_S)$ and that of $x_S = Q_S(x_S, \tilde{q}_{D(S)})$, $\mathcal{N}_Q(x_S)$. Note that because we assume that $\tilde{q}_{D(S)} = 2^u \tilde{p}_{D(S)}$, $x_S = P_S(x_S, \tilde{p}_{D(S)})$ is the result of scaling $x_S = Q_S(x_S, \tilde{q}_{D(S)})$ using $c = 2^u$. So Lemma 5.18 (iii) yields that $\mathcal{N}_P(x_S) = 2^{-u}\mathcal{N}_Q(2^u x_S)$. If $x'^{[k]} = 2^u x^{[k]}$, then $\mathcal{N}_Q(x'^{[k]}) = 2^u \mathcal{N}_P(x^{[k]})$.

If $(\mathcal{N}_P(x^{[k]}))_i \leq 0$, we would set $x_i^{[k+1]} := 0$. If so, $\mathcal{N}_Q(x'^{[k]})_i = 2^u \mathcal{N}_P(x^{[k]})_i \leq 0$, so we would set $x_i'^{[k+1]} := 0$.

If $(\mathcal{N}_P(x^{[k]}))_i > 0$, we set $x_i^{[k+1]}$ to be the result of rounding $(\mathcal{N}_P(x^{[k]}))_i$ down to a multiple of 2^h. But then $\mathcal{N}_Q(x'^{[k]}) = 2^u \mathcal{N}_P(x^{[k]}) > 0$ and we would set $x_i'^{[k+1]}$ to be the result of rounding $(\mathcal{N}_Q(x'^{[k]}))_i$ down to a multiple of $2^{-h'}$. Note that $h' = h - u$. So the result of rounding $2^u(\mathcal{N}_P(x^{[k]}))_i$ down to a multiple of $2^{-h'}$ is just 2^u times the result of rounding $(\mathcal{N}_P(x^{[k]}))_i$ down to a multiple of 2^{-h}. So $x'^{[k+1]} = 2^u x^{[k+1]}$.

This completes the induction showing that $x'^{[k]} = 2^u x^{[k]}$ for all $k \geq 0$. Note that $g = g'$. So $\tilde{q}_S = x'^{[g']} = 2^u x^{[g]} = 2^u \tilde{p}_S$. This in turn completes the induction on the SCCs, showing that $\tilde{q} = 2^u \tilde{p}$, which completes the proof. □

We can again obtain worst-case expressions for the needed number of iterations g', and the needed rounding parameter h', in terms of only $f \leq d \leq n \leq |Q|$, and ε', by noting that $\log(\|Q(\mathbf{1})\|_\infty) \leq |Q|$ and by appealing to Theorem 5.1 to remove references to $q^*_{\min}$ and $q^*_{\max}$ in the bounds. Substituting and simplifying we get that to guarantee additive error at most ε', i.e. for $i = \log(1/\varepsilon')$ bits of precision, it suffices in the worst-case to apply $g' = k_Q + c_Q \cdot i$ iterations of R-DNM with rounding parameter $h' = g' + 1$ (which is more accurate rounding than $h' = g' + 1 - u$), where $c_Q = 2^f$, and $k_Q = O(2^f 5^n n^2 d(|Q| + n\log n))$ (and we can calculate precise, tame, constants for the big-O expression).

Corollary 5.20. *If $x = P(x)$ is an MPS with LFP solution q^* with $\mathbf{0} < q^*_{\min} \leq q^*_i \leq q^*_{\max}$ for all i, with the least coefficient of any monomial in $P(x)$, $c_{\min}$, with f the nonlinear depth of the DAG of SCCs of $x = P(x)$ and with encoding size $|P|$ bits, we can compute an approximation $\tilde{q}$ to q^* with $\|q^* - \tilde{q}\|_\infty \leq \varepsilon$, for any given $0 < \varepsilon \leq 1$, in time polynomial in $|P|$,2^f, $\log\frac{1}{\varepsilon}$,$\log\frac{1}{q^*_{\min}}$ and $\log q^*_{\max}$.*

Proof. After preprocessing to remove all variables x_i with $q^*_i = 0$, which takes P-time in $|P|$, we use R-DNM as specified in Theorem 5.19. Calculating a Newton iterate at z is just a matter of solving a matrix equation and if the coordinates of z are multiples of 2^{-h} this can be done in time polynomial in $|P|$ and h. Theorem 5.19 tells us that the number of iterations and h are polynomial in 2^f, $\log\frac{1}{\varepsilon}$, $\log\frac{1}{q^*_{\min}}$, $\log q^*_{\max}$, n, $\log\frac{1}{c_{\min}}$ and $\log\|P(\mathbf{1})\|_\infty$. The last three of these are bounded by $|P|$. Together, these give the corollary. □

5.4 MPSs and Probabilistic 1-Counter Automata

A **probabilistic 1-counter automaton** (p1CA), M, is a 3-tuple $M = (V, \delta, \delta_0)$ where V is a finite set of *control states* and $\delta \subseteq V \times \mathbb{R}_{>0} \times \{-1,0,1\} \times V$ and $\delta_0 \subseteq V \times \mathbb{R}_{>0} \times \{0,1\} \times V$ are *transition relations*. The transition relation δ is enabled when the counter is nonzero, and the transition relation δ_0 is enabled when it is zero. For example, a transition of the form, $(u, p, -1, v) \in \delta$, says that if the counter value is positive, and we are currently in control state u, then with probability p we move in the next step to control state v and we decrement the counter by 1. A p1CA defines in the obvious way an underlying countably infinite-state (labeled) Markov chain, whose set of configurations (states) are pairs $(v,n) \in V \times \mathbb{N}$. A *run* (or *trajectory*, or *sample path*), starting at initial state (v_0, n_0) is defined in the usual way, as a sequence of configurations $(v_0,n_0),(v_1,n_1),(v_2,n_2),\ldots$ that is consistent with the transition relations of M.

As explained in [EWY10], p1CAs are in a precise sense equivalent to discrete-time *quasi-birth-death processes* (QBDs), and to *1-box recursive Markov chains*.

Quantities that play a central role for the analysis of QBDs and p1CAs (both for transient analyses and steady-state analyses, as well as for model checking) are their *termination probabilities* (also known as their *G-matrix* in the QBD literature, see, e.g., [LR99, BLM05, EWY10]). These are defined as the probabilities, $q^*_{u,v}$, of hitting counter value 0 for the first time in control state $v \in V$, when starting in configuration $(u, 1)$.

Corresponding to the termination probabilities of every QBD or p1CA is a special kind of MPS, $x = P(x)$, whose LFP solution q^* gives the termination probabilities of the p1CA. The MPSs corresponding to p1CAs have the following special structure. For each pair of control states $u, v \in V$ of the p1CA, there is a variable x_{uv}. The equation for each variable x_{uv} has the following form:

$$x_{uv} = p_{uv}^{(-1)} + \left(\sum_{w \in V} p_{uw}^{(0)} x_{wv} \right) + \sum_{y \in V} p_{uy}^{(1)} \sum_{z \in V} x_{yz} x_{zv} \tag{5.16}$$

where for all states $u, v \in V$, and $j \in \{-1, 0, 1\}$, the coefficients $p_{uv}^{(j)}$ are non-negative transition probabilities of the p1CA, and such that for all states $u \in V$, we have $\sum_{j \in \{-1,0,1\}} \sum_{v \in V} p_{uv}^{(j)} \leq 1$. We can of course clean up this MPS in P-time (by Proposition 2.12), to remove all variables x_{uv} for which $q^*_{u,v} = 0$. In what follows, we assume this has been done, and thus that for the remaining variables $\mathbf{0} < q^* \leq 1$.

In [EWY10], the decomposed Newton's method (DNM) is used *with exact arithmetic* in order to approximate the LFP for p1CAs using polynomially many arithmetic operations, i.e., in polynomial time in the *unit-cost arithmetic model of computation*. However [EWY10] did not establish any result about the rounded down version of DNM, and thus no results on the time required in the standard Turing model of computation. We establish instead results about R-DNM applied to the MPSs arising from p1CAs, in order to turn this method into a P-time algorithm in the standard model of computation.

It was shown in [EWY10] that in any path through the DAG of SCCs of the dependency graph for the MPS associated with a p1CA, M, there is at most one nonlinear SCC, i.e. the nonlinear depth is ≤ 1. Also, [EWY10] obtained a lower bound on $q^*_{\min}$, the smallest positive termination probability. Namely, if $c_{\min}$ denotes the smallest positive transition probability of a p1CA, M, and thus also the smallest positive constant or coefficient of any monomial in the corresponding MPS, $x = P(x)$, they showed:

Lemma 5.21. *(Corollary 6 from [EWY10])* $q^*_{\min} \geq c_{\min}^{r^3}$*, where* r *is the number of control states of the p1CA.*

They used these results to bound the *condition number* of the Jacobian matrix for each of the linear SCCs, and to thereby show that one can approximate q^* in polynomially many arithmetic operations using decomposed Newton's method. Here, we get a stronger result, placing the problem of computing termination probabilities for p1CA in P-time in the standard Turing model, using the results from this Chapter:

Theorem 5.22. *Let* $x = P(x)$ *be the MPS associated with p1CA,* M*, let* r *denote the number of control states of* M*, and let* m *denote the maximum number of bits required to represent the numerator and denominator of any positive rational transition probability in* M*.*

Apply R-DNM, including rounding down linear SCCs, to the MPS $x = P(x)$, using rounding parameter $h := 8mr^7 + 2mr^5 + 9r^2 + 3 + \lceil 2\log\frac{1}{\varepsilon}\rceil$ and such that for each nonlinear SCC we perform $g = h-1$ iterations, whereas for each linear SCC we only perform 1 R-NM iteration.

This algorithm computes an approximation $\tilde{q}$ to q^*, such that $\|q^* - \tilde{q}\|_\infty < \varepsilon$. The algorithm runs in time polynomial in $|M|$ and $\log\frac{1}{\varepsilon}$, in the standard Turing model of computation.

This follows from Theorem 5.17, using the fact that $\log(1/q^*_{\min})$ is polynomially bounded by Lemma 5.21, and the fact that the nonlinear depth of the MPS $x = P(x)$ for any p1CA is $f \leq 1$ ([EWY10]).

Proof. We apply Theorem 5.17, which tells us that R-DNM with parameter

$$h \geq \left\lceil 3 + 2^f \cdot \left(\log(\frac{1}{\varepsilon}) + d \cdot (\log(\alpha^{-(4n+1)}) + \log(16n) + \log(\|P(\mathbf{1})\|_\infty)) \right) \right\rceil \tag{5.17}$$

using $g = h-1$ iterations for every SCC, gives an approximation $\tilde{q}$ to q^* with $\tilde{q} \leq q^*$ and such that $\|q^* - \tilde{q}\|_\infty \leq \varepsilon$. Here $f \leq 1$ since there is at most 1 nonlinear SCC in any path through the dependency graph. Furthermore, $n = r^2$ since the variables in x are indexed by two states x_{uv}. Also, $d \leq n$, and so $d \leq r^2$. Also, $c_{\min} \geq 2^{-m}$ and so by Lemma 5.21, $q^*_{\min} \geq 2^{-mr^3}$. So $\alpha \geq 2^{-(mr^3+1)}$. To show that $\|P(\mathbf{1})\|_\infty \leq r$, by equation (5.16), $P(\mathbf{1})_{uv} = p^{(-1)}_{uv} + (\sum_{w \in V} p^{(0)}_{uw}) + \sum_{y \in V} p^{(1)}_{uy} r \leq r$. Plugging all this into equation (5.17), we get: $h \geq \lceil 3 + 2 \cdot (\log(\frac{1}{\varepsilon}) + r^2 \cdot ((4r^2+1)(mr^3+1) + \log(16r^2) + \log r\rceil$. Noting that $\log(16r^2) + \log r = \log(16r^3)$, and noting that when $r \geq 1$, $\log(16r^3) \leq 4r$, we have:

$$h \geq 3 + 8mr^7 + 2mr^5 + 9r^4 + \lceil 2 \cdot \log(\frac{1}{\varepsilon})\rceil$$

Note that the rounding parameter h and the number of iterations $g = h-1$ are both polynomials in the encoding size of the p1CA, and in $\log\frac{1}{\varepsilon}$. Thus each iteration of R-DNM can be computed in polynomial time, and we only do polynomially many iterations. Thus the entire computation of $\tilde{q}$ can be carried out in P-time in the Turing model of computation. □

5.4.1 Application to ω-regular model checking for p1CAs

Since computing termination probabilities of p1CAs (equivalently, the G-matrix of QBDs) plays such a central role in other analyses (see, e.g., [LR99, BLM05, EWY10, BKK11]), the P-time algorithm given in the previous section for computing termination probabilities of a p1CA (within arbitrary desired precision) directly facilitates P-time algorithms for various other important problems.

Here we highlight just one of these applications: a P-time algorithm *in the Turing model of computation* for model checking a p1CA with respect to any ω-regular property. An analogous result was established by Brazdil, Kiefer, and Kucera [BKK11] in the unit-cost RAM model of computation.

Theorem 5.23. *Given a p1CA, M, with states labeled from an alphabet Σ, and with a specified initial control state v, and given an ω-regular property $L(B) \subseteq \Sigma^{\omega}$, which is specified by a non-deterministic Büchi automaton, B, let $Pr_M(L(B))$ denote the probability that a run of M starting at configuration $(v,0)$ generates an ω-word in $L(B)$. There is an algorithm that, for any $\varepsilon > 0$, computes an additive ε-approximation, $\tilde{p} \geq 0$, of $Pr_M(L(B))$, i.e., with $|Pr_M(L(B)) - \tilde{p}| \leq \varepsilon$. The algorithm runs in time polynomial in $|M|$, $\log \frac{1}{\varepsilon}$, and $2^{|B|}$, in the standard Turing model of computation.*

Proof sketch. By Theorem 5.22, we know we can compute termination probabilities q^* for a p1CA, M, with additive error $\varepsilon > 0$ in time polynomial in $|M|$ and $\log \frac{1}{\varepsilon}$.

Let us first observe that if we do not insist on having the ω-regular property specified by a non-deterministic Büchi automaton $\mathcal{B}$, and instead assume it is specified by a deterministic Rabin automaton R, then the analogous theorem follows immediately as a corollary of Theorem 5.22 and results established by Brazdil, Kiefer, and Kucera in [BKK11]. Specifically, in [BKK11] it was shown that, given a p1CA, M, and a deterministic Rabin automaton, R, and given $\varepsilon > 0$, there is an algorithm that, firstly, decides in P-time whether $P_M(L(R)) > 0$, and if so computes a value $\tilde{p}$ which approximates $P_M(L(R))$ with *relative error* $\varepsilon > 0$, i.e., such that $|P_M(L(R)) - \tilde{p}|/P_M(L(R)) < \varepsilon$, and the algorithm runs in time polynomial in $|M|$, $|R|$, and $\log \frac{1}{\varepsilon}$, *in the unit-cost RAM model of computation.*

The first observation we make is that, the results in [EWY10] and [BKK11] together imply that for p1CAs there is no substantial difference in complexity between relative and absolute approximation, because the probabilities $P_M(L(R))$ can be bounded away from zero by $1/2^{poly(|M|,|R|)}$ if it is not equal to zero (which can be detected in P-time). Thus, computing $P_M(L(R))$ with given relative error $\varepsilon > 0$ is P-time equivalent to computing $P_M(L(R))$ with ε absolute error.

Secondly, a close inspection of [BKK11] shows that the *only* use made in their entire paper of the unit-cost RAM model of computation is for the purpose of computing termination probabilities for p1CAs, and specifically because they directly invoke the earlier result from [EWY10] which showed that termination probabilities q^* for a p1CA can be ε-approximated in polynomial time in the unit-cost RAM model. Thus, the only thing needed in order to obtain an absolute error ε-approximation of $P_M(L(R))$ in P-time in the standard Turing model of computation is to appeal instead to Theorem 5.22 of this chapter for computation of termination probabilities in P-time in the standard Turing model, and apply the rest of the construction in [BKK11].

Next, let's first note that we can of course use Safra's construction to convert any non-deterministic Büchi automaton $\mathcal{B}$ to a deterministic Rabin automaton of size $2^{O(|B|\log|B|)}$. So, obtaining a complexity bound that is polynomial in $2^{|B|\log|B|}$ is no more difficult.

Let us now very briefly sketch why one can in fact obtain the (slightly) better complexity bound, polynomial in $2^{|B|}$, by combining prior results regarding model checking of RMCs [EY12] with Theorem 5.22 and Lemma 5.21, and with the key result by Brazdil, et. al. in [BKK11], which establishes that non-zero *non-termination* probabilities for a p1CA are also bounded away from zero by $1/2^{poly(|M|)}$.

As shown in [CY95, EY12], for probabilistic model checking a *naive* subset construction can be used (instead of Safra's construction) to obtain from a BA, $\mathcal{B}$, a deterministic Büchi automaton, D, such that $|D| = 2^{|B|}$. (It need not be the case that $L(D) = L(B)$.) One then constructs the "product" $M \otimes D$, of the p1CA, M, with the deterministic Büchi automaton D. A key observation is that this "product" remains a p1CA. In terms of Recurisive Markov Chains(RMCs), p1CAs correspond to 1-box RMCs, and the "product" of a 1-box RMC with a deterministic BA, D, remains a 1-box RMC.

It was shown in [EY12] that given a "product" (1-box) RMC $M \otimes D$, it is possible to construct a finite-state *conditioned summary chain*, $\mathcal{M}'$, which is a finite state Markov chain and whose transition probabilities are *rational expressions in positive termination and non-termination probabilities* of the (1-box) RMC. It is then possible to identify in P-time certain bottom strongly connected components $\mathcal{T}$ of $\mathcal{M}'$, such that the probability $P_M(L(B))$ is equal to the probability that starting from a specific initial state of $\mathcal{M}'$, a run eventually hits a state in $\mathcal{T}$.

In this way, the model checking problem is boiled down to the problem of computing hitting probabilities in a *finite-state* Markov chain whose transition probabilities are simple rational expressions with numerators and denominators that are products of coefficients in a p1CA together with positive termination and non-termination probabilities of a p1CA.

It is well known that non-zero hitting probabilities for a finite-state Markov chain are the unique solution $(I-A)^{-1}b$, to a linear system of equations $x = Ax+b$, where the coefficients in A and b come from the transition probabilities of the Markov chain. The key remaining question is, *how well-conditioned is this linear system of equations?*. In other words, what happens to its unique solution if we only approximate the coefficients in A and b to within a small error? Now, the key is that applying Lemma 5.21 (which is from [EWY10]), and applying the key result in [BKK11], together shows that both positive termination and positive non-termination probabilities of the product p1CA are bounded away from 0 by $1/2^{poly(|M|,|D|)}$.

Under these conditions, *exactly the same* known condition number bounds from numerical analysis that were used in [EWY10] namely Theorem 17 of [EWY10], which is a version of Theorem 2.1.2.3 of [IK66], also establish that the linear system of equations that one has to solve for hitting probabilities in the conditioned summary chain $\mathcal{M}'$ derived from a p1CA are “polynomially well-conditioned”, meaning that approximating their non-zero coefficients within suitable $1/2^{poly}$ additive error yields a linear system of equations whose unique solution is ε-close to the unique solution of the original system, for the chosen $\varepsilon > 0$. We omit a detailed elaboration here. □

5.5 Bounds on the size of LFPs for an MPS

Theorem 5.1. *If $x = P(x)$ is a quadratic MPS in n variables, with LFP $q^* > \mathbf{0}$, and where $P(x)$ has rational coefficients and total encoding size $|P|$ bits, then*

1. $q^*_{\min} \geq 2^{-|P|(2^n-1)}$, and

2. $q^*_{\max} \leq 2^{2(n+1)(|P|+2(n+1)\log(2n+2))\cdot 5^n}$.

Proof. We first prove (1.), by lower bounding $q^*_{\min}$ in terms of the smallest constant $c_{\min}$ in $P(x)$.

Lemma 5.24. *If $x = P(x)$ has LFP $q^* > \mathbf{0}$, and least term $c_{\min}$, then $q^*_{\min} \geq \min\{1, c_{\min}\}^{2^n-1}$.*

Proof. We first observe that, since $q^* > \mathbf{0}$, and there are n variables, it must be the case that $P^n(0) > 0$. To see this, for any $y \geq 0$, let us use $Z(y)$ to denote the set of zero coordinates of y. For any $k \geq 0$, $P^{k+l}(0) \geq P^k(0)$, for all $l \geq 0$, so $Z(P^{k+l}(0)) \subseteq Z(P^k(0))$. Thus either $|Z(P^{k+1}(0))| = |Z(P^k(0))|$ or $|Z(P^{k+1}(0))| \leq |Z(P^k(0))| - 1$.

Now $|Z(0)| = n$ and $|Z(P^k(0))| \geq 0$ for all k, so there must be some least $0 \leq k \leq n$ such that $|Z(P^k(0))| = |Z(P^{k+1}(0))|$ and such that $Z(P^k(0)) = Z(P^{k+1}(0))$.

Note that, for any $y \geq 0$, $Z(P(y))$ depends only on $Z(y)$ and on $P(x)$, but not on the specific values of non-zero coordinates of y.

So if for some $n \geq k \geq 0$, $Z(P^{k+1}(0)) = Z(P^k(0))$ then, by a simple induction $Z(P^{k+l}(0)) = Z(P^k(0))$ for all $l \geq 0$. So we must have $Z(P^k(0)) = Z(P^n(0)) = Z(P^{n+l}(0))$, for all $l \geq 0$. Now $\lim_{m\to\infty} P^m(0) = q^*$. Now if $P^n(0)_i = 0$, then $P^{n+l}(0)_i = 0$ for all $l \geq 0$, and so $q^*_i = 0$. This contradicts our assumption that $q^* > 0$. So $P^n(0) > 0$.

Let us use $P^k(0)_@$ to denote the minimum value of any non-zero coordinate of $P^k(0)$. Firstly, $P(0) \neq 0$, i.e., there is some non-zero constant in some polynomial $P_i(x)$. Thus $P(0)_@ \geq c_{\min}$. We show by induction that for $k > 0$, $P^k(0)_@ \geq \min\{1, c_{\min}\}^{2^k-1}$. This is true for $k = 0$. We assume that $P^k(0)_@ \geq \min\{1, c_{\min}\}^{2^k-1}$. If for some coordinate i, $P^{k+1}(0)_i = P(P^k(0))_i > 0$, there must be a term in $P_i(x)$ which is not zero in $P(P^k(0))_i$, this is either a constant c, or a linear term cx_j with $P^k(0)_j > 0$, or a quadratic term cx_jx_l with $P^k(0)_j > 0$ and $P^k(0)_l > 0$. In any of these 3 cases, this term is $\geq c_{\min} \min\{1, P^k(0)_@\}^2$. Since $P^k(0)_@ \geq \min\{1, c_{\min}\}^{2^k-1}$, we now have that $P^{k+1}(0)_@ \geq c_{\min}(\min\{1, c_{\min}\}^{2^k-1})^2 \geq \min\{1, c_{\min}\}^{2^{k+1}-1}$. So for all k, $P^k(0)_@ \geq \min\{1, c_{\min}\}^{2^k-1}$. In particular $P^n(0)_@ \geq \min\{1, c_{\min}\}^{2^n-1}$. But $P^n(0) > 0$ so $P^n(0)_{\min} \geq \min\{1, c_{\min}\}^{2^n-1}$. We know $q^* \geq P^n(0)$, so $q^*_{\min} \geq \min\{1, c_{\min}\}^{2^n-1}$. □

To get our lower bound on $q^*_{\min}$ in terms of $|P|$ and n, we just note that clearly $c_{\min} \geq 2^{-|P|}$. This and Lemma 5.24 give the bound $q^*_{\min} \geq 2^{-|P|(2^n-1)}$ in part (1.) of the Theorem.

We now prove part (2.). The proof is similar to that of Lemma 3.37. To prove the upper bound on $q^*_{\max}$, we will need Theorem 3.38.

To apply Theorem 3.38, we now establish that q^* is an isolated solution of an MPS with LFP $q^* > \mathbf{0}$.

Lemma 5.25. *If $x = P(x)$ is a quadratic MPS with LFP $q^* > 0$, then q^* is an isolated solution of the system of equations $x = P(x)$.*

Proof. Firstly, we consider strongly connected MPSs. These can be divided into two cases, linear strongly-connected MPSs, where $B(x) = B$ is a constant matrix and $P(x)$ is affine, and nonlinear strongly-connected MPSs, where $B(x)$ is not a constant matrix and $P(x)$ is nonlinear.

For the linear case, the Jacobian is a constant $B(x) = B$, and $x = P(x) = Bx + P(0)$. We know that $\rho(B(q^*)) \leq 1$ from Corollary 5.8, and thus since $B = B(0) = B(q^*)$, from Lemma 5.7, we know that $\rho(B) < 1$, and thus $(I - B)$ is non-singular, and there is a unique solution to $x = P(x) = Bx + P(0)$, namely $q^* = (I - B)^{-1}P(0)$. Being unique, this solution is isolated.

Now suppose, for contradiction, that $x = P(x)$ is a nonlinear strongly-connected quadratic MPS but that $q^* > \mathbf{0}$ is not an isolated solution to $x = P(x)$. Because q^* is not isolated, there is another fixed-point q with $\|q^* - q\|_\infty \leq q^*_{\min}$ and $q \neq q^*$. Then $q \geq 0$ and, since q^* is the least non-negative fixed-point, $q \geq q^*$. From Lemma 3.3 we have:

$$P(q) - P(q^*) = B(\frac{1}{2}(q^* + q))(q - q^*)$$

Because q^* and q are fixed points

$$q - q^* = B(\frac{1}{2}(q^* + q))(q - q^*)$$

Lemma 5.9 now yields that since $q - q^* \geq 0$ but $q - q^* \neq 0$ and $B(\frac{1}{2}(q^* + q))$ is irreducible, $q > q^*$. Thus $q - q^* > \mathbf{0}$ is a positive eigenvector of the irreducible matrix $B(\frac{1}{2}(q^* + q))$ associated with eigenvalue 1, thus $\rho(B(\frac{1}{2}(q^* + q))) = 1$ by Lemma 5.6.

We now again invoke the assumption of non-isolation of q^*, which implies there is a vector $q' \neq q^*$ such that $q' = P(q')$ and $\|q^* - q'\|_\infty \leq \min\{q^*_{\min}, \frac{1}{2}(q - q^*)_{\min}\}$. By the same reasoning as above, we have that $q' > q^*$ and $\rho(B(\frac{1}{2}(q^* + q'))) = 1$. But now the condition $\|q^* - q'\|_\infty \leq \frac{1}{2}(q - q^*)_{\min}$ yields that $q' \leq q^* + \frac{1}{2}(q - q^*) < q$. We thus also have that $\frac{1}{2}(q^* + q) > \frac{1}{2}(q^* + q')$, and because $B(x)$ is non-constant and monotone in x, we have $B(\frac{1}{2}(q^* + q)) \geq B(\frac{1}{2}(q^* + q'))$ and $B(\frac{1}{2}(q^* + q)) \neq B(\frac{1}{2}(q^* + q'))$. However, $\rho(B(\frac{1}{2}(q^* + q))) = 1 = \rho(B(\frac{1}{2}(q^* + q')))$. This contradicts Lemma 5.4. So q^* is also isolated in this case.

This establishes that for all strongly-connected MPSs, with LFP $q^* > \mathbf{0}$, q^* is isolated.

Now suppose that $x = P(x)$ is not strongly-connected. For each SCC S of $x = P(x)$, the MPS $x_S = P_S(x_S, q^*_{D(S)})$ is strongly connected, so its LFP q^*_S is an isolated solution of $x_S = P_S(x_S, q^*_{D(S)})$. That is, there is an $\varepsilon_S > 0$ such that if q_S has $\|q_S - q^*_S\| \leq \varepsilon_S$ and $q_S = P_S(q_S, q^*_{D(S)})$, then $q_S = q^*_S$. Now take $\varepsilon = \min_S\{\varepsilon_S\}$. We claim that if $\|q - q^*\|_\infty \leq \varepsilon$ and $P(q) = q$, then $q = q^*$. We can show this by induction on the depth of strongly-connected components. If S is a bottom strongly-connected component, then q_S has $\|q_S - q^*_S\|_\infty \leq \varepsilon \leq \varepsilon_S$ and $q_S = P_S(q_S)$. So $q_S = q^*_S$. If S is a SCC and for all variables $D(S)$ that variables in S depend on, directly or indirectly, $q_{D(S)} = q^*_{D(S)}$, then q_S has

$q_S = P_S(q_S, q_{D(S)}) = P_S(q_S, q^*_{D(S)})$. But this and $\|q_S - q^*_S\|_\infty \leq \varepsilon \leq \varepsilon_S$ are enough to establish $q_S = q^*_S$. This completes the induction showing that $q = q^*$. So q^* is isolated solution for any MPS with LFP $q^* > \mathbf{0}$. □

For each x_i, let d_i be the product of the denominators of all coefficients of $P_i(x)$. Then $d_i x = d_i P_i(x)$ clearly has integer coefficients which are no larger than $2^{|P|}$. Suppose $x = P(x)$ has LFP $q^* > \mathbf{0}$, and suppose that coordinate k is the maximum coordinate of q^*, i.e., that $q^*_k = q^*_{\max}$. Now consider the system of $n+1$ polynomial equations, in $n+1$ variables (with an additional variable y), given by:

$$d_i x_i = d_i P_i(x)\ ,\ \text{ and }\ \text{for all } i \in \{1,\ldots,n\};\ \text{ and }\ x_k y = 1\ . \tag{5.18}$$

Lemma 5.25 tells us that $q^* > \mathbf{0}$ is an isolated solution of $x = P(x)$. If $z \in \mathbb{R}^n$ is any solution vector for $x = P(x)$, there is a unique $w \in \mathbb{R}$ such that $x := z$ and $y := w$ forms a solution to the equations (5.18); namely let $w = \frac{1}{z_k}$. So, letting $x := q^*$, and letting $y := \frac{1}{q^*_k}$ for all i, gives us an isolated solution of the equations (5.18). We can now apply Theorem 3.38 to the system (5.18). For $y = \frac{1}{q^*_k}$, equation (3.19) in Theorem 3.38 says that

$$2^{-2(n+1)(|P|+2(n+1)\log(2n+2))5^n} < \frac{1}{q^*_k} \quad \text{or} \quad \frac{1}{q^*_k} = 0\ .$$

Since $q^* > \mathbf{0}$, clearly $\frac{1}{q^*_k} \neq 0$, so $\frac{1}{q^*_{\max}} = \frac{1}{q^*_k} > 2^{-2(n+1)(|P|+2(n+1)\log(2n+2))5^n}$. So

$$q^*_{\max} < 2^{2(n+1)(|P|+2(n+1)\log(2n+2))5^n}. \tag{5.19}$$

□

5.5.1 How good are our upper bounds for R-DNM on MPSs?

In this chapter we have proved upper bounds on the number of iterations required by R-DNM to converge to within additive error $\varepsilon > 0$ of the LFP q^* for an arbitrary MPS $x = P(x)$.

We now discuss some important parameters of the problem in which our upper bounds can not be improved substantially.

To begin with, our upper bounds for the number of iterations required contain a term of the form $2^d \log \frac{1}{\varepsilon}$. Here d denotes the nesting depth of SCCs in the dependency graph G_P of the input MPS, $x = P(x)$.

It was already pointed out in [EKL10] (Section 7) that such a term is a lower bound using Newton's method on MPSs, even for exact Newton's method (whether decomposed or not), even for rather simple MPSs. [EKL10] provided a family of simple examples entailing the lower bound. Indeed, consider the following MPS, $x = P(x)$, which is a simpler variant of bad MPSs noted in [EKL10]. The MPS has $n+1$ variables, $x_0, \ldots, x_n$. The equations are:

$$\begin{aligned} x_i &= \frac{1}{2}x_i^2 + \frac{1}{2}x_{i-1} \ , \quad \text{for all } i \in \{1,\ldots,n\} \\ x_0 &= \frac{1}{2}x_0^2 + \frac{1}{2} \end{aligned}$$

The LFP of this MPS is $q^* = \mathbf{1}$, and it captures the termination probabilities of a (rather simple) stochastic context-free grammar, pBPA, or 1-exit Recursive Markov chain. Note that the encoding size of this MPS is $|P| = O(n)$.

As observed in [EY09], *exact* Newton's method, starting from $x^{(0)} := 0$, on the univariate equation $x_0 = \frac{1}{2}x_0^2 + \frac{1}{2}$ gains exactly one bit of precision per iteration. In other words, if $x^{(k)}$ denotes the k'th iterate, then $1 - x^{(k)} = 2^{-k}$.

Suppose we perform m iterations of exact Newton's method on the bottom SCC, $x_0 = \frac{1}{2}x_0^2 + \frac{1}{2}$, and suppose that by doing so we obtain an approximation $q'_0 = 1 - a_0$, where $a_0 = 2^{-m}$. Plugging the approximation q'_0 into the next higher SCC, the equation for x_1 becomes $x_1 = \frac{1}{2}x_1^2 + \frac{1}{2}q'_0$. For the rest of the argument we do not need to appeal to Newton iterations: even *exact* computation of the LFPs for the remaining SCCs will yield bad approximations overall unless $1 - q'_0 \leq \frac{1}{2^{2^n}}$ (showing that the system of equations is terribly *ill-conditioned*).

Indeed, by induction on $i \geq 0$, suppose that the value obtained for LFP of x_i is $q'_i = (1 - a_i)$. Then after plugging in q'_i in place of x_i in the SCC for x_{i+1}, the adjusted LFP, q'_{i+1}, of the next higher SCC: $x_{i+1} = (1/2)(x_1)^2 + (1/2)(1 - a_i)$, becomes $q'_{i+1} = 1 - \sqrt{a_i}$. Thus, by induction on depth, the adjusted LFP of x_n becomes $q'_n = 1 - a_0^{2^{-n}}$. But $a_0 = 2^{-m}$. Thus $q'_n = 1 - 2^{-m2^{-n}}$.

We would like to have error $1 - q'_n = 2^{-m2^{-n}} \leq \varepsilon$. Taking logs, we get that we must perform at least $m \geq 2^n \log \frac{1}{\varepsilon}$ Newton iterations on the bottom SCC alone.

Note that n here is also the depth d of SCCs in this example.

Other terms in our upper bounds on the number of iterations required to compute the LFP of a general MPS are $\log \frac{1}{q^*_{\min}}$, and $\log q^*_{\max}$. Simple "repeated squaring" MPSs, with $x_i = x_{i-1}^2$, $x_0 = \{\frac{1}{2} \text{ or } 2\}$, show that we can have $q^*_{\min} \leq \frac{1}{2^{2^n}}$, and $q^*_{\max} \geq 2^{2^n}$, where n is the number of variables. In Theorem 5.1 we give explicit lower bounds on $q^*_{\min}$

and explicit upper bounds on $q^*_{\max}$, in terms of $|P|$ and n, showing that linear-double-exponential dependence on n is indeed the worst case possible.

However, it should be noted that the worst-case bounds on $q^*_{\min}$ and $q^*_{\max}$ are not representative of many important families of MPSs. In particular, note that MPSs corresponding to termination probabilities must have $q^*_{\max} \leq 1$. Furthermore, for a number of classes of probabilistic systems we can prove bounds of the form $\log \frac{1}{q^*_{\min}} \leq poly(|P|)$. Indeed, for MPSs corresponding to quasi-birth-death processes and probabilistic 1-counter automata, such a bound was proved in [EWY10].

If the family of MPSs happens to have $\log \frac{1}{q^*_{\min}}, \log q^*_{\max} \leq poly(|P|)$, then our upper bounds show that the total number of iterations of R-DNM needed is only exponential in d, the depth of SCCs, and thus if $d \leq \log |P|$, then for such MPSs R-DNM runs *in P-time in the encoding size of the input,* $|P|$ *and* $\log \frac{1}{\varepsilon}$*, in the standard Turing model of computation*, to compute an approximation to the LFP q^*, within additive error $\varepsilon > 0$.

It should be noted that for the case of *strongly connected MPSs* only, and only for *Exact Newton's Method*, without rounding, [EKL10] obtained comparable result to ours in terms of worst-case dependence on $\log \frac{1}{q^*_{\min}}$ and $\log q^*_{\max}$. Their bounds are with respect to *relative error*, and their bounds for strongly connected MPSs do not depend at all on $q^*_{\max}$, but of course if $q^*_{\max}$ is large, then in order to obtain absolute (additive) error $\varepsilon > 0$, the relative error required is $\varepsilon' = \frac{\varepsilon}{q^*_{\max}}$, and since their bounds depend on $\log \frac{1}{\varepsilon'}$ they depend (indirectly) on $\log q^*_{\max}$, with the same magnitude as ours. However, in [EKL10] they did not obtain any constructive bounds in terms of $|P|$, $q^*_{\min}$ or $q^*_{\max}$ for MPSs that are not strongly connected, nor did they obtain any results for rounded versions of Newton's method. Using exact Newton's method of course entails the assumption of a unit-cost arithmetic model of computation, rather than the Turing model.

Chapter 6

Stochastic Context-Free Grammars and Regular Languages

A number of important problems on SCFGs can be viewed as instances of the following *regular pattern matching problem* for different regular languages:

Given an SCFG G and a regular language L, given e.g., by a deterministic finite automaton (DFA) D, compute the probability $\mathbb{P}_G(L)$ that G generates a string in L, i.e. compute the sum for all the strings in L of the probability that G generates that string.

A simple example is when $L = \Sigma^*$, the set of all strings over the terminal alphabet Σ of the SCFG G. Then this problem simply asks to compute the probability $\mathbb{P}_G(L(G))$ of the language $L(G)$ generated by the grammar G. Assuming that the start nonterminal of the grammar is S, this is precisely the probability q_S^G defined in Chapter 2, which amounts to the probability of termination starting at nonterminal S in the SCFG G. We gave a P-time algorithm for approximating q_S^G in Chapter 3.

Another simple example is when L is a singleton, $L = \{w\}$, for some string w; in this case the problem corresponds to the basic parsing question of computing the probability that a given string w is generated by the SCFG G. In the paper [ESY12b], we showed that the probability $\mathbb{P}_G(\{w\})$ of string w under SCFG G can also be computed to any precision in P-time in the size of G, w and the number of bits of precision. We will not explain that algorithm in this thesis, since we instead discuss the more general problem of computing the probability that the SCFG generates a string in a given regular language.

Another basic well-studied problem is the computation of *prefix probabilities*: given an SCFG G and a string w, compute the probability that G generates a string with prefix w [JL91, Sto95]. This is useful in online processing in speech recognition [JL91] and corresponds to the case $L = w\Sigma^*$. A more complex problem is the computation of *infix probabilities* [CMGS91, NS11], where we wish to compute the probability that G generates a string that contains a given string w as a substring, which corresponds to the language $L = \Sigma^* w \Sigma^*$.

As usual, even when rule probabilities of the SCFG G are rational, the probabilities we wish to compute can be irrational and we aim to approximate them to a desired precision.

Stochastic context-free grammars are closely related to *1-exit recursive Markov chains* (1-RMC) [EY09], and to *stateless probabilistic pushdown automata* (also called pBPA) [EKM06]; these are two equivalent models for a subclass of probabilistic programs with recursive procedures. The above regular pattern matching problem for SCFGs is equivalent to the problem of computing the probability that a computation of a given 1-RMC (or pBPA) terminates and satisfies a given regular property. In other words, it corresponds to the quantitative model checking problem for 1-RMCs with respect to regular *finite string* properties.

Previous Work. As mentioned above, there has been, on the one hand, substantial work in the NLP literature on different cases of the problem for various regular languages L, and on the other hand, there has been work in the verification and algorithms literature on the analysis and model checking of recursive Markov chains and probabilistic pushdown automata.

The model checking problem for RMCs (equivalently pPDAs) and ω-regular properties was studied in [EKM06, EY12]. This is of course a more general problem than the problem for SCFGs (which correspond to 1-RMCs) and regular languages (the finite string case of ω-regular languages). It was shown in [EY12] that in the case of 1-RMCs, the qualitative problem of determining whether the probability that a run satisfies the property is 0 or 1 can be solved in P-time in the size of the 1-RMC, and that for the quantitative problem of approximating the probability, there is an algorithm that runs in PSPACE, and no better complexity bound was known.

The particular cases of computing prefix and infix probabilities for an SCFG have been studied in the NLP literature, but no polynomial time algorithm for general SCFGs is known. Jelinek and Lafferty gave an algorithm for grammars in Chomsky Normal Form (CNF) [JL91]. Another algorithm for prefix probabilities by Stolcke [Sto95] applies to general SCFGs, but in the presence of unary and ε-rules, the algorithm does not run in polynomial time. The problem of computing infix probabilities was studied in [CMGS91, NS09, NS11], and in particular [NS09, NS11] cast it in the general regular language framework, and studied the general problem of computing the probability $\mathbb{P}_G(L(D))$ of the language $L(D)$ of a DFA D under an SCFG G. From G and D they construct a product *weighted context-free grammar* (WCFG) G': a CFG with (positive) weights on the rules, which may not be probabilities, in particular the weights on the rules of a nonterminal may sum to more than 1. The desired probability $\mathbb{P}_G(L(D))$ is the weight of $L(G')$. As in the case of SCFGs, this weight is given by the LFP of an MPS $y = P_{G'}(y)$. Nederhof and Satta then solve the system using the decomposed Newton method from [EY09] and Broyden's (quasi-Newton) method, and present experimental results for infix probability computations.

However, note that unlike in the case of termination probabilities for SCFGs, the system now is not a PPS (thus our result of Chapter 3 does not apply). Also our bounds on general MPSs from Chapter 5 are exponential in the depth of (not necessarily critical) strongly connected components of $x = P(x)$, and furthermore they also depend linearly on $\log(\frac{1}{q^*_{\min}})$, where $q^*_{\min} = \min_i q^*_i$, which can be $\approx \frac{1}{2^{2^{|P|}}}$. As we describe next, we do far better in this chapter for the MPSs that arise from the "product" of an SCFG and a DFA.

Our Results. We study the general problem of computing the probability $\mathbb{P}_G(L(D))$ that a given SCFG G generates a string in the language $L(D)$ of a given DFA D. We show that, under a certain mild assumption on G, this probability can be computed to any desired precision in time polynomial in the encoding sizes of G & D and the number of bits of precision.

We now sketch briefly the approach and state the assumption on G. First we construct from G and D the product weighted CFG $G' = G \otimes D$ as in [NS09] and construct the corresponding MPS $y = P_{G'}(y)$, whose LFP contains the desired probability $\mathbb{P}_G(L(D))$ as one of its components. The system is monotone but not probabilistic. We eliminate (in P-time) those variables that have value 0 in the LFP, and apply Newton, with suitable rounding in every step. The heart of the analysis shows there is a tight algebraic correspondence between the behaviour of Newton's method on this MPS and its behaviour on the probabilistic polynomial system (PPS) $x = P_G(x)$ of G. In particular, this correspondence shows that, with exact arithmetic, the two computations converge at the same rate. By exploiting this, and by extending recent results we established for PPSs, we obtain the conditional polynomial upper bound. Specifically, call a PPS $x = P(x)$ *critical* if the spectral radius of the Jacobian of $P(x)$, evaluated at the LFP q^* is equal to 1 (it is always ≤ 1). We can form a dependency graph between the variables of a PPS, and decompose the variables and the system into strongly connected components (SCCs); an SCC is called critical if the induced subsystem on that SCC is critical. The *critical depth* of a PPS is the maximum number of critical SCCs on any path of the DAG of SCCs (i.e. the max nesting depth of critical SCCs). We show that if the PPS of the given SCFG G has bounded (or even logarithmic) critical depth, then we can compute $\mathbb{P}_G(L(D))$ (for any DFA D) in polynomial time in the size of G, D and the number of bits of precision.

Furthermore, we show this condition is satisfied by a broad class of SCFGs used in applications. Specifically, a standard way the probabilities of rules of an SCFG are set is by using the EM (inside-outside) algorithm. We show that the SCFGs constructed in this way are guaranteed to be non-critical (i.e., have critical depth 0). So for these SCFGs, and any DFA, the algorithm runs in P-time.

6.1 gives definitions and background. 6.2 establishes tight algebraic connections between the behaviour of Newton on the PPS of the SCFG, and on the MPS of the product WCFG. 6.3 proves the claimed bounds on rounded Newton's method. 6.4 shows the non-criticality of SCFGs obtained by the EM method.

6.1 Definitions

We will say that a WCFG, $G = (V, \Sigma, R, p)$ is in *Simple Normal Form* (SNF) if every nonterminal $A \in V$ belongs to one of the following three types:

1. type L: every rule $r \in R_A$, has the form $A \xrightarrow{p(r)} B$.
2. type Q: there is a single rule in R_A: $A \xrightarrow{1} BC$, for some $B, C \in V$.
3. type T: there is a single rule in R_A: either $A \xrightarrow{1} \varepsilon$, or $A \xrightarrow{1} a$ for some $a \in \Sigma$.

Note that if a WCFG is in SNF, then the associated MPS is in SNF. This is why we have used the same terminology of "simple normal form" in both cases.

For convenience, let us now recall from Chapter 2 some of the basic definitions and notations associated with WCFGs and SCFGs.

For a WCFG, G, strings $\alpha, \beta \in (V \cup \Sigma)^*$, and $\pi = r_1 \ldots r_k \in R^*$, we write $\alpha \overset{\pi}{\Rightarrow} \beta$ if the leftmost derivation starting from α, and applying the sequence π of rules, derives β. We let $p(\alpha \overset{\pi}{\Rightarrow} \beta) = \prod_{i=1}^{k} p(r_k)$ if $\alpha \overset{\pi}{\Rightarrow} \beta$, and $p(\alpha \overset{\pi}{\Rightarrow} \beta) = 0$ otherwise. If $A \overset{\pi}{\Rightarrow} w$ for $A \in V$ and $w \in \Sigma^*$, we say that π is a *complete* derivation from A and its *yield* is $y(\pi) = w$. There is a natural one-to-one correspondence between the complete derivations of w starting at A and the *parse trees* of w rooted at A, and this correspondence preserves weights.

For a WCFG, $G = (V, \Sigma, R, p)$, nonterminal $A \in V$, and terminal string $w \in \Sigma^*$, we let $p_A^{G,w} = \sum_{\{\pi | y(\pi) = w\}} p(A \overset{\pi}{\Rightarrow} w)$. For a general WCFG, $p_A^{G,w}$ need not be a finite value (it may be $+\infty$, since the sum may not converge). Note however that if G is an SCFG, then $p_A^{G,w}$ defines the probability that, starting at nonterminal A, G generates w, and thus it is clearly finite.

For any WCFG, $G = (V, \Sigma, R, p)$, with $n = |V|$, assume the nonterminals in V are indexed as $A_1, \ldots, A_n$. Recall that there is an MPS associated with G, denoted $x = P_G(x)$. Here $x = (x_1, \ldots, x_n)$ denotes an n-vector of variables. Likewise $P_G(x) = (P_G(x)_1, \ldots, P_G(x)_n)$ denotes an n-vector of multivariate polynomials over the variables $x = (x_1, \ldots, x_n)$. For a vector $\kappa = (\kappa_1, \kappa_2, \ldots, \kappa_n) \in \mathbb{N}^n$, we use the notation x^κ to denote the monomial $x_1^{\kappa_1} x_2^{\kappa_2} \ldots x_n^{\kappa_n}$. For a non-terminal $A_i \in V$, and a string $\alpha \in (V \cup \Sigma)^*$, let $\kappa_i(\alpha) \in \mathbb{N}$ denote the number of occurrences of A_i in the string α. We define $\kappa(\alpha) \in \mathbb{N}^n$ to be $\kappa(\alpha) = (\kappa_1(\alpha), \kappa_2(\alpha), \ldots, \kappa_n(\alpha))$.

In the MPS $x = P_G(x)$, corresponding to each nonterminal $A_i \in V$, there will be one variable x_i and one equation, namely $x_i = P_G(x)_i$, where: $P_G(x)_i \equiv \sum_{r=(A \to \alpha) \in R_{A_i}} p(r) x^{\kappa(\alpha)}$. If there are no rules associated with A_i, i.e., if $R_{A_i} = \emptyset$, then by default we define $P_G(x)_i \equiv 0$. Note that if $r \in R_{A_i}$ is a terminal rule, i.e., $\kappa(r) = (0, \ldots, 0)$, then $p(r)$ is one of the constant terms of $P_G(x)_i$.

Note: *Throughout this chapter, for any n-vector z, whose i'th coordinate z_i "corresponds" to nonterminal A_i, we often find it convenient to use z_{A_i} to refer to z_i.* So, e.g., we alternatively use x_{A_i} and $P_G(x)_{A_i}$, instead of x_i and $P_G(x)_i$.

We use $|G|$ to denote the encoding size (i.e., number of bits) of a input WCFG G.

Given any WCFG (SCFG) $G = (V, \Sigma, R, p)$ we can compute in linear time an SNF form WCFG (resp. SCFG) $G' = (V' \Sigma, R', p')$ of size $|G'| = O(|G|)$ with $V' \supseteq V$ such that $p_A^{G,w} = p_A^{G',w}$ for all $A \in V$, $w \in \Sigma^*$ (see Proposition 2.9). Thus, for the problems studied in this chapter, we may assume wlog that a given input WCFG or SCFG is in SNF form.

A DFA, $D = (Q, \Sigma, \Delta, s_0, F)$, has states Q, alphabet Σ, transition function $\Delta : Q \times \Sigma \to Q$, start state $s_0 \in Q$ and final states $F \subseteq Q$. We extend Δ to strings: $\Delta^* : Q \times \Sigma^* \to Q$ is defined by induction on the length $|w| \geq 0$ of $w \in \Sigma^*$: for $s \in Q$, $\Delta^*(s, \varepsilon) := s$. Inductively, if $w = aw'$, with $a \in \Sigma$, then $\Delta^*(s, w) := \Delta^*(\Delta(s, a), w')$. We define $L(D) = \{w \in \Sigma^* \mid \Delta^*(s_0, w) \in F\}$.

Given a WCFG G and a DFA D over the same terminal alphabet, for any nonterminal A of G, we define $p_A^{G,D} = \sum_{w \in L(D)} p_A^{G,w}$. If G is an SCFG, $p_A^{G,D}$ simply denotes the probability that G, starting at A, generates a string in $L(D)$. Our goal is to compute

$p_A^{G,D}$, given SCFG G and DFA D. In general, $p_A^{G,D}$ may be an irrational probability, even when all of the rule probabilities of G are rational values. So one natural goal is to approximate $p_A^{G,D}$ to within desired precision. More precisely, the approximation problem is this: given as input an SCFG, G, with a specified nonterminal A, a DFA, D, over the same terminal alphabet Σ, and a rational error threshold $\delta > 0$, output a rational value $v \in [0,1]$ such that $|v - p_A^{G,D}| < \delta$. We would like to do this as efficiently as possible as a function of the input size: $|G|$, $|D|$, and $\log(1/\delta)$.

To compute $p_A^{G,D}$, it will be useful to define a WCFG obtained as the *product* of an SCFG and a DFA. We assume, wlog, that the input SCFG is in SNF form. The ***product*** (or ***intersection***) of an SCFG $G = (V, \Sigma, R, p)$ in SNF form, and DFA, $D = (Q, \Sigma, \Delta, s_0, F)$, is defined to be a new WCFG, $G \otimes D = (V', \Sigma, R', p')$, where the set of nonterminals is $V' = Q \times V \times Q$. Assuming $n = |V|$ and $d = |Q|$, then $|V'| = d^2 n$. The rules R' and rule probabilities p' of the product $G \otimes D$ are defined as follows (recall G is assumed to be in SNF):

- Rules of form L: For every rule of the form $(A \xrightarrow{p} B) \in R$, and every pair of states $s, t \in Q$, there is a rule $(sAt) \xrightarrow{p} (sBt)$ in R'.

- Rules of form Q: for every rule $(A \xrightarrow{1} BC) \in R$, and for all states $s, t, u \in Q$, there is a rule $(sAu) \xrightarrow{1} (sBt)(tCu)$ in R'.

- Rules of form T: for every rule $(A \xrightarrow{1} a) \in R$, where $a \in \Sigma$, and for every state $s \in Q$, if $\Delta(s, a) = t$, then there is a rule $(sAt) \xrightarrow{1} a$ in R'.

  For every rule $(A \xrightarrow{1} \varepsilon) \in R$, and every $s \in Q$, there is a rule $(sAs) \xrightarrow{1} \varepsilon$

Associated with the WCFG, $G \otimes D$, is the MPS $y = P_{G\otimes D}(y)$, where y is now a $d^2 n$-vector of variables, where $n = |V|$ and $d = |Q|$. The LFP solution of this MPS captures the probabilities $p_A^{G,D}$ in the following sense:

Proposition 6.1. *(cf. [NS11], or [EY12] for a variant of this) For any SCFG, $G = (V, \Sigma, R, p)$, and DFA, $D = (Q, \Sigma, \Delta, s_0, F)$, the LFP solution $q^{G\otimes D}$ of the MPS $x = P_{G\otimes D}(x)$, satisfies $\mathbf{0} \le q^{G\otimes D} \le \mathbf{1}$. Furthermore, for any $A \in V$ and $s, t \in Q$, $q^{G\otimes D}_{(sAt)} = \sum_{\{w \mid \Delta^*(s,w) = t\}} p_A^{G,w}$. Thus, for every $A \in V$, $p_A^{G,D} = \sum_{t \in F} q^{G\otimes D}_{(s_0At)}$.*

6.2 Balance, Collapse, and Newton's method

For an SCFG, $G = (V, \Sigma, R, p)$, and a DFA, $D = (Q, \Sigma, \Delta, s_0, F)$, we want to relate the behaviour of Newton's method on the MPS associated with the WCFG, $G \otimes D$, to that of the PPS associated with the SCFG G. We shall show that there is indeed a tight correspondence, regardless of what the DFA D is. This holds even when G itself is a convergent WCFG, and thus $x = P_G(x)$ is an MPS. We need an abstract algebraic way to express this correspondence. A key notion will be *balance*, and the *collapse* operator defined on balanced vectors and matrices.

Consider the LFP q^G of $x = P_G(x)$, and LFP $q^{G\otimes D}$ of $y = P_{G\otimes D}(y)$. By Propos. 2.10 and 6.1, for any $A \in V$, $q^G_A = \sum_{w\in\Sigma^*} p_A^{G,w}$ is the probability (weight) that G, starting at A, generates any finite string. Likewise $q^{G\otimes D}_{(sAt)} = \sum_{\{w|\Delta^*(s,w)=t\}} p_A^{G,w}$ is the probability (weight) that, starting at A, G generates a finite string w such that $\Delta^*(s,w) = t$. Thus, for any $A \in V$ and $s \in Q$, $q^G_A = \sum_{t\in Q} q^{G\otimes D}_{(sAt)}$.

It turns out that analogous relationships hold between many other vectors associated with G and $G\otimes D$, including between the Newton iterates obtained by applying Newton's method to their respective PPS (or MPS) and the product MPS. Furthermore, associated relationships also hold between the Jacobian matrices $B_G(x)$ and $B_{G\otimes D}(y)$ of $P_G(x)$ and $P_{G\otimes D}(y)$, respectively.

Let $n = |V|$ and let $d = |Q|$. A vector $y \in \mathbb{R}^{d^2n}$, whose coordinates are indexed by triples $(sAt) \in Q \times V \times Q$, is called ***balanced*** if for any non-terminal A, and any pair of states $s, s' \in Q$, $\sum_{t\in Q} y_{(sAt)} = \sum_{t\in Q} y_{(s'At)}$. In other words, y is balanced if the value of the sum $\sum_{t\in Q} y_{(sAt)}$ is independent of the state s. As already observed, $q^{G\otimes D} \in \mathbb{R}^{d^2n}_{\geq 0}$ is balanced. Let $\mathfrak{B} \subseteq \mathbb{R}^{d^2n}$ denote the set of balanced vectors. Let us define the ***collapse*** mapping $\mathfrak{C} : \mathfrak{B} \to \mathbb{R}^n$. For any $A \in V$, $\mathfrak{C}(y)_A := \sum_t y_{(sAt)}$. Note: $\mathfrak{C}(y)$ is well-defined, because for $y \in \mathfrak{B}$, and any $A \in V$, the sum $\sum_t y_{(sAt)}$ is by definition independent of the state s.

We next extend the definition of balance to matrices. A matrix $M \in \mathbb{R}^{d^2n\times d^2n}$ is called ***balanced*** if, for any non-terminals $B, C \in V$ and states $s, u \in Q$, and for any pair of states $v, v' \in Q$, $\sum_t M_{(sBt),(uCv)} = \sum_t M_{(sBt),(uCv')}$, and for any $s, v \in Q$ and $s', v' \in Q$, $\sum_{t,u} M_{(sBt),(uCv)} = \sum_{t,u} M_{(s'Bt),(uCv')}$. Let $\mathfrak{B}^\times \subseteq \mathbb{R}^{d^2n\times d^2n}$ denote the set of balanced matrices. We extend the ***collapse*** map $\mathfrak{C}$ to matrices. $\mathfrak{C} : \mathfrak{B}^\times \to \mathbb{R}^{n\times n}$ is defined as follows. For any $M \in \mathfrak{B}^\times$, and any $B, C \in V$, $\mathfrak{C}(M)_{BC} := \sum_{t,u} M_{(sBt),(uCv)}$. Note, again, $\mathfrak{C}(M)$ is well-defined.

We denote the Newton operator, $\mathcal{N}$, applied to a vector $x' \in \mathbb{R}^n$ for the PPS $x = P_G(x)$ associated with G by $\mathcal{N}_G(x')$. Likewise, we denote the Newton operator applied to a vector $y' \in \mathbb{R}^{d^2n}$ for the MPS $y = P_{G\otimes D}(y)$ associated with $G \otimes D$ by $\mathcal{N}_{G\otimes D}(y')$. For a real square matrix M, let $\rho(M)$ denote the spectral radius of M. The main result of this section is the following:

Theorem 6.2. *Let $x = P_G(x)$ be any PPS (or MPS), with n variables, associated with an SCFG (or WCFG) G, and let $y = P_{G\otimes D}(y)$ be the corresponding product MPS, for any DFA D, with d states. For any balanced vector $y \in \mathfrak{B} \subseteq \mathbb{R}^{d^2n}$, with $y \geq 0$, $\rho(B_{G\otimes D}(y)) = \rho(B_G(\mathfrak{C}(y)))$. Furthermore, if $\rho(B_{G\otimes D}(y)) < 1$, then $\mathcal{N}_{G\otimes D}(y)$ is defined and balanced, $\mathcal{N}_G(\mathfrak{C}(y))$ is defined, and $\mathfrak{C}(\mathcal{N}_{G\otimes D}(y)) = \mathcal{N}_G(\mathfrak{C}(y))$. Thus, $\mathcal{N}_{G\otimes D}$ preserves balance, and the collapse map $\mathfrak{C}$ "commutes" with $\mathcal{N}$ over non-negative balanced vectors, irrespective of what the DFA D is.*

We establish this via a series of lemmas that reveal many algebraic and analytic properties of balance, collapse, and their interplay with Newton's method. Lemma 6.3 first establishes a series of algebraic and analytic properties of arbitrary balanced vectors and matrices. Lemma 6.4 then uses these to establish properties of the specific balanced matrices and vectors arising during iterations of Newton's method on PPSs (and MPSs), and on corresponding product MPSs. Theorem 6.2 is an immediate consequence of Lemma 6.4, parts $(i)\&(iv)$, below.

Lemma 6.3. *Consider the set $\mathfrak{B} \subseteq \mathbb{R}^{d^2n}$ of balanced vectors, and the set $\mathfrak{B}^\times \subseteq \mathbb{R}^{d^2n\times d^2n}$ of balanced matrices. Let $\mathfrak{B}_{\geq 0} = \mathfrak{B} \cap \mathbb{R}^{d^2n}_{\geq 0}$ and $\mathfrak{B}^\times_{\geq 0} = \mathfrak{B} \cap \mathbb{R}^{d^2n\times d^2n}_{\geq 0}$.*

(i) $\mathfrak{B}$ and $\mathfrak{B}^\times$ are both closed under linear combinations. In other words: $\sum_i \alpha_i v^{\langle i\rangle} \in \mathfrak{B}$ and $\sum_i \alpha_i M^{\langle i\rangle} \in \mathfrak{B}^\times$, if, $\forall\, i$, $v^{\langle i\rangle} \in \mathfrak{B}$ and $M^{\langle i\rangle} \in \mathfrak{B}^\times$.

Furthermore, $\mathfrak{C}$ is a linear map on both $\mathfrak{B}$ and $\mathfrak{B}^\times$. In other words: $\mathfrak{C}(\sum_i \alpha_i v^{\langle i\rangle}) = \sum_i \alpha_i \mathfrak{C}(v^{\langle i\rangle})$ and $\mathfrak{C}(\sum_i \alpha_i M^{\langle i\rangle}) = \sum_i \alpha_i \mathfrak{C}(M^{\langle i\rangle})$, whenever, $\forall i$, $\alpha_i \in \mathbb{R}$, $v^{\langle i\rangle} \in \mathfrak{B}$, and $M^{\langle i\rangle} \in \mathfrak{B}^\times$.

(ii) If $M \in \mathfrak{B}^\times$ and $v \in \mathfrak{B}$, then $Mv \in \mathfrak{B}$ and $\mathfrak{C}(Mv) = \mathfrak{C}(M)\mathfrak{C}(v)$.

(iii) If $M, M' \in \mathfrak{B}^\times$, then $MM' \in \mathfrak{B}^\times$ and $\mathfrak{C}(MM') = \mathfrak{C}(M)\mathfrak{C}(M')$.

(iv) If $M \in \mathfrak{B}^\times_{\geq 0}$, and $v \in \mathbb{R}^{d^2n}$ is any vector, then $\mathfrak{C}(Mv) \geq \mathfrak{C}(M)\mathfrak{C}(v)$, where we extend the map $\mathfrak{C}$ to arbitrary $v' \in \mathbb{R}^{d^2n}$ by letting $\mathfrak{C}(v')_A := \min_s \sum_t v'_{(sAt)}$.

(v) If $M \in \mathfrak{B}^\times_{\geq 0}$, then $\rho(M) = \rho(\mathfrak{C}(M))$. In other words, the collapse operator $\mathfrak{C}$ preserves the spectral radius of balanced non-negative matrices.

(vi) If $v \in \mathfrak{B}_{\geq 0}$, then $\|v\|_\infty \leq \|\mathfrak{C}(v)\|_\infty$. If $M \in \mathfrak{B}^\times_{\geq 0}$ then $\|M\|_\infty \leq d\|\mathfrak{C}(M)\|_\infty$.

Proof.

(i): This can be verified directly from the definitions of balance and collapse. In particular, for any nonterminal $A \in V$, and any states $s, s' \in Q$:

$$\begin{aligned}
\sum_t (\sum_i \alpha_i v^{\langle i\rangle})_{(sAt)} &= \sum_i \alpha_i \sum_t v^{\langle i\rangle}_{(sAt)} \\
&= \sum_i \alpha_i \mathfrak{C}(v^{\langle i\rangle})_A \quad \text{(because every } v^{\langle i\rangle} \text{ is balanced)} \\
&= \sum_i \alpha_i \sum_t v^{\langle i\rangle}_{(s'At)} \\
&= \sum_t (\sum_i \alpha_i v^{\langle i\rangle})_{(s'At)}
\end{aligned}$$

Also, we have $\mathfrak{C}(\sum_i \alpha_i v^{\langle i\rangle})_A := \sum_t (\sum_i \alpha_i v^{\langle i\rangle})_{(sAt)} = \sum_i \alpha_i \mathfrak{C}(v^{\langle i\rangle})_A$.

Likewise, for any nonterminals $B, C \in V$, and any states $s, u \in Q$ and $v, v' \in Q$:

$$\begin{aligned}
\sum_t (\sum_i \alpha_i M^{\langle i\rangle})_{(sBt),(uCv)} &= \sum_i \alpha_i \sum_t M^{\langle i\rangle}_{(sBt),(uCv)} \\
&= \sum_i \alpha_i \sum_t M^{\langle i\rangle}_{(sBt),(uCv')} \quad \text{(because every } M^{\langle i\rangle} \text{ is balanced)} \\
&= \sum_t (\sum_i \alpha_i M^{\langle i\rangle})_{(sBt),(uCv')}
\end{aligned}$$

Similarly, for any nonterminals B, C, and any states $s, v, s', v' \in Q$:

$$\begin{aligned}
\sum_{t,u} (\sum_i \alpha_i M^{\langle i\rangle})_{(sBt),(uCv)} &= \sum_i \alpha_i \sum_{t,u} M^{\langle i\rangle}_{(sBt),(uCv)} \\
&= \sum_i \alpha_i \sum_{t,u} M^{\langle i\rangle}_{(s'Bt),(uCv')} \quad \text{(because every } M^{\langle i\rangle} \text{ is balanced)} \\
&= \sum_{t,u} (\sum_i \alpha_i M^{\langle i\rangle})_{(s'Bt),(uCv')}
\end{aligned}$$

Now, $\mathfrak{C}(\sum_i \alpha_i M^{\langle i\rangle})_{B,C} := \sum_{t,u} (\sum_i \alpha_i M^{\langle i\rangle})_{(sBt),(uCv)} = \sum_i \alpha_i \sum_{t,u} M^{\langle i\rangle}_{(sBt),(uCv)} = \sum_i \alpha_i \mathfrak{C}(M^{\langle i\rangle})_{B,C}$.

(ii): For any non-terminal B and state s:

$$\begin{aligned}
\sum_t (Mv)_{(sBt)} &= \sum_{t,u,C,z} M_{(sBt),(uCz)} v_{uCz} \\
&= \sum_{u,C,z} (\sum_t M_{(sBt),(uCz)}) v_{uCz} \\
&= \sum_{C,u} (\sum_t M_{(sBt),(uCz)}) \sum_z v_{uCz} \quad \text{(since M is balanced)} \\
&= \sum_{C,u} (\sum_t M_{(sBt),(uCz)}) \mathfrak{C}(v)_C \quad \text{(since v is balanced)} \\
&= \sum_{C} (\sum_{t,u} M_{(sBt),(uCz)}) \mathfrak{C}(v)_C \\
&= \sum_C \mathfrak{C}(M)_{B,C} \mathfrak{C}(v)_C \, \text{(since M is balanced)} \\
\\
&= (\mathfrak{C}(M)\mathfrak{C}(v))_B
\end{aligned}$$

which is independent of s. So $\mathfrak{C}(Mv)_B = \sum_t (Mv)_{(sBt)} = (\mathfrak{C}(M)\mathfrak{C}(v))_B$.

(iii): For any non-terminal D, E, and states $s, w, x \in Q$:

$$\begin{aligned}
\sum_t (MM')_{(sDt),(wEx)} &= \sum_{t,u,C,v} M_{(sDt),(uCv)} M'_{(uCv),(wEx)} \\
&= \sum_{u,C,v} (\sum_t M_{(sDt),(uCv)}) M'_{(uCv),(wEx)} \\
&= \sum_{C,u} (\sum_t M_{(sDt),(uCv)}) \sum_v M'_{(uCv),(wEx)} \quad \text{(since M is balanced)}
\end{aligned}$$

Since $M' \in \mathfrak{B}^{\times}$, the last sum is independent of x, which is what we aimed to show. Next consider:

$$\begin{aligned}
\sum_{t,w} (MM')_{(sDt),(wEx)} &= \sum_{t,w,u,C,v} M_{(sDt),(uCv)} M'_{(uCv),(wEx)} \\
&= \sum_{u,w,C,v} (\sum_t M_{(sDt),(uCv)}) M'_{(uCv),(wEx)} \\
&= \sum_{C,u,w} (\sum_t M_{(sDt),(uCv)}) \sum_v M'_{(uCv),(wEx)} \quad \text{(since M is balanced)} \\
&= \sum_{C,u} (\sum_t M_{(sDt),(uCv)}) \sum_{v,w} M'_{(uCv),(wEx)} \\
&= \sum_{C,u} (\sum_t M_{(sDt),(uCv)}) \mathfrak{C}(M')_{C,E} \quad \text{(since M' is balanced)} \\
&= \sum_C \mathfrak{C}(M)_{D,C} \mathfrak{C}(M')_{C,E} \quad \text{(since B is balanced)} \\
&= (\mathfrak{C}(M)\mathfrak{C}(M'))_{D,E}
\end{aligned}$$

So, $\sum_{t,w} (MM')_{(sDt),(wEx)}$ is independent of s, x and $\mathfrak{C}(MM')_{D,E} = \sum_{t,w} (MM')_{(sDt),(wEx)} = (\mathfrak{C}(M)\mathfrak{C}(M'))_{D,E}$, for any $D, E \in V$.

(iv): For any non-terminal B and state s:

$$\begin{aligned}
\sum_t (Mv)_{(sBt)} &= \sum_{t,u,C,z} M_{(sBt),(uCz)} v_{uCz} \\
&= \sum_{u,C,z} (\sum_t M_{(sBt),(uCz)}) v_{uCz} \\
&= \sum_{C,u} (\sum_t M_{(sBt),(uCz)}) \sum_z v_{uCz} \quad \text{(since } M \text{ is balanced)} \\
&\geq \sum_{C,u} (\sum_t M_{(sBt),(uCz)}) \min_u \sum_z v_{uCz} \quad \text{(since } (\textstyle\sum_t M_{(sBt),(uCz)}) \geq 0) \text{ for any } C, u \\
&= \sum_C \mathfrak{C}(M)_{B,C} \mathfrak{C}(v)_C = (\mathfrak{C}(M)\mathfrak{C}(v))_B
\end{aligned}$$

Since this holds for any B and any s, $\mathfrak{C}(Mv)_B = \min_s \sum_t (Mv)_{(sBt)} \geq (\mathfrak{C}(M)\mathfrak{C}(v))_B$.

(vi): (we will prove part (v) below) Since $v \in \mathfrak{B}_{\geq 0}$, $v_{(sAt)} \leq \sum_{t'} v_{(sAt')} = \mathfrak{C}(v)_A$ so $\|v\|_\infty \leq \|\mathfrak{C}(v)\|_\infty$. For $M \in \mathfrak{B}^\times_{\geq 0}$:

$$\begin{aligned}
\|M\|_\infty &= \max_{s,B,t} \sum_{u,C,v} M_{(sBt),(uCv)} \\
&\leq \max_{s,B} \sum_{u,C,v,t} M_{(sBt),(uCv)} \\
&= \max_{s,B} \sum_{C,v} \mathfrak{C}(M)_{B,C} \\
&= \max_B d \sum_C \mathfrak{C}(M)_{B,C} \\
&= d\|\mathfrak{C}(M)\|_\infty
\end{aligned}$$

(v): By standard facts from Perron-Frobenius theory (see, e.g., Theorem 8.3.1 of [HJ85]), the non-negative matrix $\mathfrak{C}(M)$, has as an eigenvalue $\rho(\mathfrak{C}(M))$ associated with which is a non-negative eigenvector $v_G \neq 0$. That is $\mathfrak{C}(M)v_G = \rho(\mathfrak{C}(M))v_G$ for some non-zero $v_G \geq 0$. Now consider any non-negative balanced vector u with $\mathfrak{C}(u) = v_G$. (Such a u obviously exists.) Let $f(u) = \frac{1}{\rho(\mathfrak{C}(M))}Mu$. By part (ii), Mu is balanced and $\mathfrak{C}(Mu) = \mathfrak{C}(M)v_G = \rho(\mathfrak{C}(M))v_G$. So, $f(u)$ is non-negative and balanced and has $\mathfrak{C}(f(u)) = v_G$. The set of non-negative balanced vector u with $\mathfrak{C}(u) = v_G$ is compact (it is a product of simplices) and the continuous function f maps this set into itself. So by *Brouwer's fixed point theorem*, f has a fixed point, that is a u^* with $u^* = \frac{1}{\rho(\mathfrak{C}(M))}Mu^*$. That is, u^* is an eigenvector of M with eigenvalue $\rho(\mathfrak{C}(M))$. So $\rho(M) \geq \rho(\mathfrak{C}(M))$.

In the other direction, we use the fact (see, e.g., Theorem 5.6.12 of [HJ85]) that for any square matrix N, $\lim_{k\to\infty} \|N^k\|_\infty = 0$ if and only if $\rho(N) < 1$.

Now for $M \in \mathfrak{B}^\times_{\geq 0}$ assume, for contradiction, that $\rho(M) > \rho(\mathfrak{C}(M))$. Then $\rho(\frac{1}{\rho(M)}M) =$

$\frac{1}{\rho(M)}\rho(M) = 1 > \frac{1}{\rho(M)}\rho(\mathfrak{C}(M)) = \rho(\frac{1}{\rho(M)}\mathfrak{C}(M))$. Thus, by the above fact from matrix theory, we have that $\lim_{k\to\infty} \|(\frac{1}{\rho(M)}\mathfrak{C}(M))^k\|_\infty = 0$.

But for any $k \geq 1$,

$$\begin{aligned} 0 \leq \|(\frac{1}{\rho(M)}M)^k\|_\infty &\leq d\|\mathfrak{C}((\frac{1}{\rho(M)}M)^k)\|_\infty \quad \text{(by part } (vi)) \\ &= d\|\mathfrak{C}(\frac{1}{\rho(M)}M)^k\| \quad \text{(by part } (iii)) \\ &= d\|(\frac{1}{\rho(M)}\mathfrak{C}(M))^k\|_\infty \quad \text{(by part } (i)) \end{aligned}$$

And thus, since the right hand side goes to 0 as $k \to \infty$, we must also have $\lim_{k\to\infty} \|(\frac{1}{\rho(M)}M)^k\|_\infty = 0$, but this is a contradiction, because $\rho(\frac{1}{\rho(M)}M) = 1$. So, our assumption $\rho(M) > \rho(\mathfrak{C}(M))$ must be false.

Having established both directions, we conclude that $\rho(M) = \rho(\mathfrak{C}(M))$. □

Lemma 6.4. *Let $\mathfrak{B}_{\geq 0} = \mathfrak{B} \cap \mathbb{R}^{d^2 n}_{\geq 0}$ and $\mathfrak{B}^{\times}_{\geq 0} = \mathfrak{B} \cap \mathbb{R}^{d^2 n \times d^2 n}_{\geq 0}$.*
We have $q^{G\otimes D} \in \mathfrak{B}_{\geq 0}$ and $\mathfrak{C}(q^{G\otimes D}) = q^G$, and:

(i) If $y \in \mathfrak{B}_{\geq 0} \subseteq \mathbb{R}^{d^2 n}_{\geq 0}$ then $B_{G\otimes D}(y) \in \mathfrak{B}^{\times}_{\geq 0}$, and $\mathfrak{C}(B_{G\otimes D}(y)) = B_G(\mathfrak{C}(y))$.

(ii) If $y \in \mathfrak{B}_{\geq 0}$, then $P_{G\otimes D}(y) \in \mathfrak{B}_{\geq 0}$, and $\mathfrak{C}(P_{G\otimes D}(y)) = P_G(\mathfrak{C}(y))$.

(iii) If $y \in \mathfrak{B}_{\geq 0}$ and $\rho(B_G(\mathfrak{C}(y))) < 1$, then $I - B_{G\otimes D}(y)$ is non-singular, $(I - B_{G\otimes D}(y))^{-1} \in \mathfrak{B}^{\times}_{\geq 0}$, and $\mathfrak{C}((I - B_{G\otimes D}(y))^{-1}) = (I - B_G(\mathfrak{C}(y)))^{-1}$.

(iv) If $y \in \mathfrak{B}_{\geq 0}$ and $\rho(B_G(\mathfrak{C}(y))) < 1$, then $\mathcal{N}_{G\otimes D}(y) \in \mathfrak{B}^{\times}$ and $\mathfrak{C}(\mathcal{N}_{G\otimes D}(y)) = \mathcal{N}_G(\mathfrak{C}(y))$.

Proof.

Firstly, let us recall why $q^{G\otimes D} \in \mathfrak{B}_{\geq 0}$ and $\mathfrak{C}(q^{G\otimes D}) = q^G$. Recall these are the LFP q^G, of $x = P_G(x)$, and the LFP $q^{G\otimes D}$ of $y = P_{G\otimes D}(y)$. By Propositions 2.10 and 6.1, for any nonterminal $A \in V$, $q^G_A = \sum_{w\in\Sigma^*} p^{G,w}_A$ is the probability (weight) that G generates any finite string w. Likewise $q^{G\otimes D}_{(sAt)} = \sum_{\{w|\Delta^*(s,w)=t\}} p^{G,w}_A$ is the probability (weight) that, starting at A, G generates a finite string w such that $\Delta^*(s,w) = t$. Thus, clearly, for any $A \in V$, and any $s \in Q$, $q^G_A = \sum_{t\in Q} q^{G\otimes D}_{(sAt)} = \mathfrak{C}(q^{G\otimes D})_A$. Now we prove the enumerated assertions one by one:

(i): We need to argue both that $B_{G\otimes D}(y) \in \mathfrak{B}^{\times}_{\geq 0}$, and that $\mathfrak{C}(B_{G\otimes D}(y)) = B_G(\mathfrak{C}(y))$, for $y \in \mathfrak{B}_{\geq 0}$. Again, recall that we are assuming wlog that G is in SNF form. We split the proof into cases depending on the type of non-terminal A in $B_{G\otimes D}(y)_{(sAt),(uEv)}$. Let $\delta_{\alpha,\beta}$ denote the Dirac function: $\delta_{\alpha,\beta} := 1$ if $\alpha = \beta$, and $\delta_{\alpha,\beta} := 0$ if $\alpha \neq \beta$.

Type `Q`: For any non-terminal A of type `Q`, the only rule in R_A has the form $A \xrightarrow{1} BC$, and $P_G(x)_A \equiv x_B x_C$. And, for any states $s,t \in Q$, $P_{G\otimes D}(y)_{(sAt)} \equiv \sum_{w\in Q} y_{(sBw)} y_{(wCt)}$. Thus

$$B_{G\otimes D}(y)_{(sAt),(uEv)} \doteq \frac{\partial P_{G\otimes D}(y)_{(sAt)}}{\partial y_{(uEv)}} = \delta_{t,v} \cdot \delta_{E,C} \cdot y_{(sBu)} + \delta_{s,u} \cdot \delta_{E,B} \cdot y_{(vCt)}$$

Thus

$$\sum_t B_{G\otimes D}(y)_{(sAt),(uEv)} = \delta_{E,C} \cdot y_{(sBu)} + \delta_{s,u} \cdot \delta_{E,B} \cdot \sum_t y_{(vCt)}$$

Since y is balanced, $\sum_t y_{(vCt)}$ is independent of v, so $\sum_t B_{(sAt),(uEv)}$ is independent of v. Next we note that:

$$\sum_{t,u} B_{G\otimes D}(y)_{(sAt),(uEv)} = \delta_{E,C} \sum_u y_{(sBu)} + \delta_{E,B} \sum_t y_{(vCt)}$$

Thus

$$\sum_{t,u} B_{G\otimes D}(y)_{(sAt),(uEv)} = \delta_{E,C} \mathfrak{C}(y)_B + \delta_{E,B} \mathfrak{C}(y)_C = B_G(\mathfrak{C}(y))$$

Type `T`: For any non-terminal A of type `T`, $P_G(x)_A$ does not depend on x, and $P_{G\otimes D}(y)_{sAt}$ does not depend on y, for any $s,t \in Q$. Thus $\sum_t B_{G\otimes D}(y)_{(sAt),(uCv)} = 0$, and $\sum_{t,u} B_{G\otimes D}(y)_{(sAt),(uCv)} = 0 = B_G(\mathfrak{C}(y))_{A,C}$.

Type `L`: For any non-terminal A of type `L`, recall that $P_G(x)_A = \sum_{r\in R_A} p_r x_{B_r}$. And for any states s,t, $P_{G\otimes D}(y)_{(sAt)} = \sum_{r\in R_A} p_r y_{(sB_r t)}$.

Thus, all the entries of $B_G(x))_{A,C}$ and $B_{G\otimes D}(y)_{(sAt),(uCv)}$ are independent of x and y, respectively. And

$$B_{G\otimes D}(y)_{(sAt),(uCv)} = \frac{\partial P_{G\otimes D}(y)_{(sAt)}}{\partial y_{(uCv)}} = \delta_{s,u} \cdot \delta_{t,v} \cdot B_G(x)_{A,C}$$

Consequently $\sum_t B_{G\otimes D}(y)_{(sAt),(uCv)} = \delta_{s,u} B_G(x)_{A,C} =$, which is independent of v. And, $\sum_{t,u} B_{G\otimes D}(y)_{(sAt),(uCv)} = B_G(x)_{A,C}$, which is independent of s and v, and $B_G(x)_{A,C} = B_G(\mathfrak{C}(y))_{A,C}$, because $B_G(x)_{A,C}$ is independent of x.

Having shown that for all nonterminals A and C, and all nonterminals $s,u \in Q$, the sum $\sum_t B_{G\otimes D}(y)_{(sAt),(uCv)}$ is independent of v. And we have also shown that for all nonterminals A and C, the sum $\sum_{t,u} B_{G\otimes D}(y)_{(sAt),(uCv)}$ is independent of s and v, and furthermore, that the latter sum (which is by definition $\mathfrak{C}(B_{G\otimes D}(y))_{A,C}$), is equal to

$B_G(\mathfrak{C}(y))$. Thus our proof for part (i) is complete.

(ii): Part (ii) could be proved using a case-by-case analysis similar to part (i). Instead, we shall use part (i). Recall that $P_G(x)$ and $P_{D\otimes G}(y)$ have no polynomials of degree more than 2. Furthermore:

$$P_G(x) = P_G(0) + B_G(\frac{1}{2}x)x$$

And

$$P_{G\otimes D}(y) = P_{G\otimes D}(0) + B_{G\otimes D}(\frac{1}{2}y)y$$

By the previous parts of this Lemma, and by Lemma 6.3, we know that $B_{G\otimes D}(\frac{1}{2}y)y$ is balanced, and $\mathfrak{C}(B_{G\otimes D}(\frac{1}{2}y)y) = B_G(\frac{1}{2}\mathfrak{C}(y))\mathfrak{C}(y)$. All that remains is to show that $P_{G\otimes D}(0)$ is balanced and that $\mathfrak{C}(P_{G\otimes D}(0)) = P_G(0)$, and again use the properties established in Lemma 6.3.

Now, unless a non-terminal A has type $\mathtt{T}$, $P_G(0)_A = 0$, and for any states $s,t \in Q$, $P_{G\otimes D}(0)_{(sAt)} = 0$. So, in these cases, there is nothing to prove. If the nonterminal A does have type $\mathtt{T}$, then $P_G(x)_A = 1$. If there is a rule $A \xrightarrow{1} a$, for some $a \in \Sigma$, then for any state $s \in Q$, there is a unique state $t' \in Q$ with $\Delta(s,a) = t'$. If instead there is a rule $A \xrightarrow{1} \varepsilon$, then let $t' := s$. In both cases, note that $\sum_t P_{G\otimes D}(y)_{(sAt)} = 1 = P_G(\mathfrak{C}(y))_A$, since $P_{G\otimes D}(y)_{(sAt)} = 1$ when $t = t'$ and $P_{G\otimes D}(y)_{(sAt)} = 0$ otherwise. Thus also $\mathfrak{C}(P_{G\otimes D}(y)) = P_G(\mathfrak{C}(y))$ in all cases.

(iii): By assumption, $\rho(B_G(\mathfrak{C}(y))) < 1$, so by Lemma 6.3 (v), $\rho(B_{G\otimes D}(y)) < 1$. It is a basic fact that for any square $M \geq 0$ if $\rho(M) < 1$ then $(I - M)$ is non-singular and $(I-M)^{-1} = \sum_{i=0}^{\infty} M^i$. (See, e.g., [LT85], Theorem 15.2.2, page 531). Thus $I - B_{G\otimes D}(y)$ is non-singular, and $(I - B_{G\otimes D}(y))^{-1} = \sum_{i=0}^{\infty}(B_{G\otimes D}(y))^i$. Note that each $(B_{G\otimes D}(y))^i$, for $i \geq 0$, is balanced, by using the previous parts of this Lemma and Lemma 6.3 (iii), and thus so are the partial sums $\sum_{i=0}^{k}(B_{G\otimes D}(y))^i$, for any $k \geq 0$. Therefore $(I - B_{G\otimes D}(y))^{-1} = \lim_{k\to\infty}\sum_{i=1}^{k}(B_{G\otimes D}(y_{G\otimes D}))^i$ is a limit of balanced non-negative matrices. But then $(I - B_{G\otimes D}(y))^{-1}$ must be balanced, because the definition of balance for a matrix M requires equalities between continuous (in fact, linear) functions of the entries, and thus if all the matrices $\sum_{i=1}^{k}(B_{G\otimes D}(y_{G\otimes D}))^i$ satisfy these conditions, then so does their limit.

Furthermore $\mathfrak{C}$ is a linear and continuous function on matrices, so $\mathfrak{C}((I - B_{G\otimes D}(y))^{-1}) = \sum_{i=1}^{\infty}\mathfrak{C}(B_{G\otimes D}(y)^i) = \sum_{i=1}^{\infty}\mathfrak{C}(B_{G\otimes D}(y))^i = (I - \mathfrak{C}(B_{G\otimes D}(y)))^{-1}$. By part (i) of this

Lemma, this is equal to $(I - B_G(\mathfrak{C}(y)))^{-1}$. Done.

(iv): By part (ii) of this Lemma, $P_{G\otimes D}(y)$ is balanced and $\mathfrak{C}(P_{G\otimes D}(y)) = P_G(\mathfrak{C}(y))$. Part (iii) of this lemma says that $(I - B_{G\otimes D}(y))^{-1}$ is balanced and $\mathfrak{C}((I - B_{G\otimes D}(y))^{-1}) = (I - \mathfrak{C}(B_{G\otimes D}(y)))^{-1}$. Now we can apply the various algebraic properties of balanced vectors and matrices from Lemma 6.3 to conclude that

$$\mathcal{N}_{G\otimes D}(y) := y + (I - B_{G\otimes D}(y)^{-1}(P_{G\otimes D}(y) - y)$$

is balanced and that $\mathfrak{C}(\mathcal{N}_{G\otimes D}(y)) = \mathfrak{C}(y) + (I - B_G(\mathfrak{C}(y)))^{-1}(P_G(\mathfrak{C}(y)) - \mathfrak{C}(y)) = \mathcal{N}_G(\mathfrak{C}(y))$.

□

As mentioned already, Theorem 6.2 follows immediately from Lemma 6.4, parts $(i)\&(iv)$.

An easy consequence of Thm. 6.2 (and Prop. 2.11) is that if we use Newton's method with exact arithmetic on the PPS or MPS, $x = P_G(x)$, and on the product MPS, $y = P_{G\otimes D}(y)$, they converge at the same rate:

Corollary 6.5. *For any PPS or MPS, $x = P_G(x)$, with LFP $q^G > 0$, and corresponding product MPS, $y = P_{G\otimes D}(y)$, if we use Newton's method with* exact arithmetic*, starting at $x^{(0)} := 0$, and $y^{(0)} := 0$, then all the Newton iterates $x^{(k)}$ and $y^{(k)}$ are well-defined, and for all k:* $\quad x^{(k)} = \mathfrak{C}(y^{(k)})$.

6.3 Rounded Newton on PPSs and product MPSs

To work in the Turing model of computation (as opposed to the unit-cost RAM model) we have to consider *rounding* between iterations of Newton's method, as in Chapter 3.

Definition 6.6. *(***Rounded-down Newton's method** *(***R-NM***), with parameter h.) Given an MPS, $x = P(x)$, with LFP q^*, where $q^* > \mathbf{0}$, in R-NM with integer rounding parameter $h > 0$, we compute a sequence of iteration vectors $x^{[k]}$. Starting with $x^{[0]} := \mathbf{0}$, $\forall k \geq 0$ we compute $x^{[k+1]}$ as follows:*

1. *Compute $x^{\{k+1\}} := \mathcal{N}_P(x^{[k]})$, where $\mathcal{N}_P(x)$ is the Newton op. defined in (2.2).*

2. *For each coordinate $i = 1,\dots,n$, set $x_i^{[k+1]}$ to be equal to the maximum multiple of 2^{-h} which is $\leq \max(x_i^{\{k+1\}}, 0)$. (In other words, round down $x^{\{k+1\}}$ to the nearest multiple of 2^{-h}, while ensuring the result is non-negative.)*

Unfortunately, rounding can cause iterates $x^{[k]}$ to become unbalanced. Nevertheless, we can handle this. For any PPS, $x = P(x)$, with Jacobian matrix $B(x)$, and LFP q^*, $\rho(B(q^*)) \leq 1$ ([EY09, ESY12b]). If $\rho(B(q^*)) < 1$, we call the PPS **non-critical**. Otherwise, if $\rho(B(q^*)) = 1$, we call the PPS **critical**. For SCFGs whose PPS $x = P_G(x)$ is non-critical, we get good bounds, even though R-NM iterates can become unbalanced.

6.3.1 Non-critical SCFGs

Theorem 6.7. *For any $\varepsilon > 0$, and for an SCFG, G, if the PPS $x = P_G(x)$ has LFP $0 < q^G \leq 1$ and $\rho(B_G(q^G)) < 1$, then if we use R-NM with parameter $h+2$ to approximate the LFP solution of the MPS $y = P_{G\otimes D}(y)$, then $\|q^{G\otimes D} - y^{[h+1]}\|_\infty \leq \varepsilon$ where $h := 14|G| + 3 + \lceil \log(1/\varepsilon) + \log d \rceil$.*

Thus we can compute the probability $q_A^{G,D} = \sum_{t\in F} q_{s_0At}^{G\otimes D}$ within additive error $\delta > 0$ in time polynomial in the input size: $|G|$, $|D|$ and $\log(1/\delta)$, in the standard Turing model of computation.

We first need to recall, and establish, a series of Lemmas and Theorems.

Lemma 6.8. *If $x = P(x)$ is a strongly connected PPS (in SNF form), with Jacobian $B(x)$, and if $B(\frac{1}{2}\mathbf{1})v \leq v$ for some vector $v > 0$, then $\frac{\|v\|_\infty}{v_{\min}} \leq 2^{|P|}$*

Proof. (This proof is a variant of that of Lemmas 3.12 and 5.9) Let $l = \arg\max_i v_i$, and let $k = \arg\min_j v_j$. Since $x = P(x)$ is in SNF form, every non-zero entry of the matrix $B(\frac{1}{2}\mathbf{1})$ is either $1/2$ or is a coefficient of some monomial in some polynomial $P_i(x)$ of $P(x)$. Moreover, $B(\frac{1}{2}\mathbf{1})$ is irreducible. Calling the entries of $B(\frac{1}{2}\mathbf{1})$, $b_{i,j}$, we have a sequence of *distinct* indices, $i_1, i_2, \ldots, i_m$, with $l = i_1$, $k = i_m$, $m \leq n$, where each $b_{i_j i_{j+1}} > 0$. (Just take the "shortest positive path" from l to k.) For any j:

$$(B(\frac{1}{2}\mathbf{1})v)_{i_{j+1}} \geq b_{i_j i_{j+1}} v_{i_j}$$

By simple induction: $v_k \geq (\prod_{j=1}^{m-1} b_{i_j i_{j+1}}) v_l$. Note that $|P|$ includes the encoding size of each positive coefficient of every polynomial $P_i(x)$. We argued before that each $b_{i_j i_{j+1}}$ is either a coefficient of $x = P(x)$, or is equal to $1/2$. Furthermore, if we consider the equation $x_{i_j} = P(x)_{i_j}$, and denote its encoding size as $|P_{i_j}|$, then it is easy to see $b_{i_j i_{j+1}} \geq 2^{-|P_{i_j}|}$, because either $b_{i_j i_{j+1}}$ appears in $P(x)_{i_j}$, or else $b_{i_j i_{j+1}} = 1/2$, but it is always the case that $|P_{i_j}| \geq 1$. Now, the i_j's are distinct (because we are using a shortest path). Therefore, since $|P| = \sum_{i=1}^{n} |P_i|$, we must have $\prod_{j=1}^{m-1} b_{i_j i_{j+1}} \geq 2^{-|P|}$, and thus we have: $v_k \geq 2^{-|P|} v_l$. □

Theorem 6.9. *If $x = P(x)$ is an MPS with n variables, with LFP $q^* \leq 1$, and $\rho(B(q^*)) < 1$, and if we use* any *rounded-down Newton iteration method defined by $x^{[0]} := 0$, and for all $k \geq 0$, and $x^{[k+1]} := \max(0, \mathcal{N}(x^{(k)}) - e_k)$, where e_k is* some error vector *such that $0 \leq (e_k)_i \leq 2^{-(h+2)}$ for all $i \in \{1,\ldots,n\}$, then for any $0 < \varepsilon \leq 1$, $\|q^* - x^{[h+1]}\|_\infty \leq \varepsilon$, whenever the chosen parameter h satisfies $h \geq \lceil \log \|(I - B(q^*))^{-1}\|_\infty + \log \frac{1}{\varepsilon} \rceil$.*

Proof. We shall use Lemma 3.9 to prove this. We need to find a vector v, with $B(q^*)v \leq v$ and $v > 0$, called a *cone vector*, such that we can bound the ratio $\frac{v_{\max}}{v_{\min}}$. Here $v_{\max} = \max_i v_i$, and $v_{\min} = \min_i v_i$.

Since we know that $\rho(B(q^*)) < 1$, we have that $(I - B(q^*))$ is nonsingular, and $(I - B(q^*))^{-1} = \sum_{i=0}^{\infty} B(q^*)^i$. We simply take $v := \frac{1}{\|(I - B(q^*))^{-1}\|_\infty}(I - B(q^*))^{-1}\mathbf{1}$ as our cone vector.

Then $B(q^*)v = v - \frac{1}{\|(I - B(q^*))^{-1}\|_\infty}\mathbf{1} \leq v$ and $v = \frac{1}{\|(I - B(q^*))^{-1}\|_\infty}(\mathbf{1} + B(q^*)\mathbf{1} + B(q^*)^2\mathbf{1}\ldots) \geq \frac{1}{\|(I - B(q^*))^{-1}\|_\infty}\mathbf{1}$. The latter not only shows that $v > 0$, but also that $v_{\min} \geq \frac{1}{\|(I - B(q^*))^{-1}\|_\infty}$. Recall that by definition, since $(I - B(q^*))^{-1}$ is non-negative, $\|(I - B(q^*))^{-1}\|_\infty$ is the maximum row sum of any row of $(I - B(q^*))^{-1} = \sum_{i=0}^{\infty} B(q^*)^i$. It follows that $v_{\max} \leq 1$, since $B(q^*)^0 = I$.

Now, $x^{[0]} := 0$, and $q^* \leq 1$, so we know that $q^* - x^{[0]} \leq 1 \leq \|(I - B(q^*))^{-1}\|_\infty v \leq 2^h \varepsilon v$ (by definition of h). Now, for all $k > 0$, $e_k \leq 2^{-(h+2)}\mathbf{1} \leq \frac{1}{4}\varepsilon \frac{1}{\|(I - B(q^*))^{-1}\|_\infty}\mathbf{1} \leq \frac{1}{4}\varepsilon v$.

Applying Lemma 3.9, if $q^* - x^{[k]} \leq \lambda v$, then $q^* - x^{[k+1]} \leq q^* - \mathcal{N}(x^{[k]}) + e_k \leq (\frac{\lambda}{2} + \frac{1}{4})\varepsilon v$. It follows by induction that, for all $k \geq 1$, $q^* - x^{[k]} \leq (2^{h-k} + \frac{1}{2})\varepsilon v$ When $k = h+1$, this gives $q^* - x^{[h+1]} \leq \varepsilon v$. Since $v_{\max} = \|v\|_\infty \leq 1$, this means that $\|q^* - x^{[h+1]}\|_\infty \leq \varepsilon$ as required. □

Theorem 6.10. *If the PPS $x = P(x)$ with LFP solution q^* has $\rho(B(q^*)) < 1$ and we use any rounded-down Newton iteration, starting at $x^{[0]} = 0$, defined by $x^{[k+1]} = \max(0, x^{[k]} + (I - B(x^{[k]}))^{-1}(P(x^{[k]}) - x^{[k]}) - e_k)$, for any error vectors e_k where $0 \leq (e_k)_i \leq 2^{-(h+2)}$ for all $i \in \{1,\ldots,n\}$, then for any given $0 < \varepsilon \leq 1$, $\|q^* - x^{[h+1]}\|_\infty \leq \varepsilon$, where $h = 14|P| + 3 + \lceil \log(1/\varepsilon) \rceil$.*

Theorem 6.10 follows from Theorem 6.9 and an upper bound on $\|(I - B(q^*))^{-1}\|_\infty$. The following Lemma gives us this, from which Theorem 6.10 follows immediately:

Lemma 6.11. *If the PPS $x = P(x)$ with LFP solution q^* has $\rho(B(q^*)) < 1$ then*

$$\|(I - B(q^*))^{-1}\|_\infty \leq 2^{14|P|+3}$$

Proof. We split into several cases, based on q^*.

Case 1: $q^* < \mathbf{1}$. In this case we just need to use Theorem 3.24 (i), in which we set $y := q^*$, combined with Theorem 3.14, to conclude that:

$$\|(I - B(q^*))^{-1}\|_\infty \leq 2^{14|P|+1}$$

Case 2: $q^* = \mathbf{1}$. In this case we can instead use the following result from Chapter 3:

Lemma 6.12. *For a PPS $x = P(x)$, if $(I - B(\mathbf{1}))$ is non-singular then*

$$\|(I - B(\mathbf{1}))^{-1}\|_\infty \leq 3^n n 2^{|P|} \leq 2^{3|P|}$$

Proof. Note that we obtained this bound in the proof of Theorem 3.15 but under the assumption that $q^* < 1$. This assumption was not used in obtaining the bound so we can use the same proof here.

If we take $(I - B(\mathbf{1}))$ to be the matrix A of Lemma 3.16, then noting that the product of all the denominators in $(I - B(\mathbf{1}))$ is at most $2^{|P|}$, this yields:

$$\|(I - B(\mathbf{1}))^{-1}\|_\infty \leq n 2^{|P|} \|(I - B(\mathbf{1}))\|_\infty^n$$

Of course $\|(I - B(\mathbf{1}))\|_\infty \leq 1 + \|B(\mathbf{1})\|_\infty \leq 3$ (note that here we are using the fact that the system is in SNF). Thus

$$\|(I - B(\mathbf{1}))^{-1}\|_\infty \leq 3^n n 2^{|P|}$$

Furthermore, as discussed before Theorem 3.15 for any PPS $x = P(x)$ we can assume wlog that the equation for every variable requires at least 3 bits, and thus that $|P| \geq 3n \geq n \log 3 + \log n$. Therefore $3^n n 2^{|P|} \leq 2^{3|P|}$. □

Case 3: Neither $q^* < \mathbf{1}$ nor $q^* = \mathbf{1}$. To finish the proof of Lemma 6.11, we will combine the above two results for the first two cases to deal with the case when neither $q^* < \mathbf{1}$ nor $q^* = \mathbf{1}$, but that nevertheless $\rho(B(q^*)) < 1$. (It is indeed possible for all three of these conditions to hold, when some coordinates of q^* are 1, and others less than 1.)

Let A (for "always") denote the set of variables x_i for which $q_i^* = 1$, and let M (for "maybe") denote the set of variables x_i for which $0 < q_i^* < 1$. We can obviously assume that both A and M are non-empty; otherwise one of the two above theorems gives the result. Furthermore, variables in A obviously cannot depend on those in M (neither directly nor indirectly). Thus we can describe $B(q^*)$ by the following block decomposition

$$B(q^*) = \begin{pmatrix} B(q^*)_M & B(q^*)_{M,A} \\ 0 & B(q^*)_A \end{pmatrix}$$

We need a lemma:

Lemma 6.13. *For any matrix M satisfying the block decomposition given by $M = \begin{pmatrix} A & B \\ 0 & D \end{pmatrix}$, if both A and D are square and non-singular matrices, then M is also non-singular, and:*

$$\|M^{-1}\|_\infty \leq \max\{\|A^{-1}\|_\infty + \|A^{-1}\|_\infty \|B\|_\infty \|D^{-1}\|_\infty, \|D^{-1}\|_\infty\}$$

Proof. The standard formula for the blockwise inverse of a matrix gives $\begin{pmatrix} A & B \\ 0 & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & -A^{-1}BD^{-1} \\ 0 & D^{-1} \end{pmatrix}$, provided that A and D are non-singular. (The formula can easily be verified directly by multiplying by $\begin{pmatrix} A & B \\ 0 & D \end{pmatrix}$.)

Now recall that the l_∞ norm for a matrix C is $\|C\|_\infty = \max_i \sum_j |C_{ij}|$, i.e., it is the maximum sum across any row of the absolute value of the entries of the row. So

$$\|M^{-1}\|_\infty \quad \leq \quad \max\{\|A^{-1}\|_\infty + \|A^{-1}\|_\infty \|B\|_\infty \|D^{-1}\|_\infty, \|D^{-1}\|_\infty\}$$

□

Now, $(I - B(q^*)) = \begin{pmatrix} I - B(q^*)_M & -B(q^*)_{M,A} \\ 0 & I - B(q^*)_A \end{pmatrix}$, so:

$$\|(I - B(q^*))^{-1}\|_\infty \leq$$

$$\max\{ \; \|(I - B(q^*)_M)^{-1}\|_\infty + \|(I - B(q^*)_M)^{-1}\|_\infty \|B(q^*)_{M,A}\|_\infty \|(I - B(q^*)_A)^{-1}\|_\infty \; , \; \|(I - B(q^*)_A)^{-1}\|_\infty \; \}$$

Since we always, wlog, assume that $x = P(x)$ is a PPS is SNF form, $\|B(q^*)\|_\infty \leq 2$. More specifically, $\|B(q^*)_{M,A}\|_\infty \leq 2$. By Case 1, since $\mathbf{0} < q^*_M < \mathbf{1}$, $\|(I - B(q^*)_M)^{-1}\|_\infty \leq 2^{14|P_M|+1}$, where $|P_M|$ denotes the encoding size of the system of equations $x_M = P(x_M, 1_A)_M$, restricted to the variables in M, and with 1 plugged in for all variables in A. Also, by Lemma 6.12, since $q^*_A = \mathbf{1}$, $\|(I - B(q^*)_A)^{-1}\|_\infty \leq 2^{3|P_A|}$, where $x_A = P(x)_A$ denotes the system of equations restricted to variables in A (note that these do not depend on variables in M). Thus,

$$\|(I - B(q^*))^{-1}\|_\infty \leq \max\{2^{14|P_M|+1} + 2^{14|P_M|+2+3|P_A|}, 2^{3|P_A|}\}$$

This can be simplified to $\|(I - B(q^*))^{-1}\|_\infty \leq 2^{14|P|+3}$. This completes the proof of Lemma 6.11. □

We now have enough to deal with Theorem 6.7, the non-critical case.

Proof of Theorem 6.7. Lemma 6.4 yields that $(I - B_{G\otimes D}(q^{G\otimes D}))^{-1} \in \mathfrak{B}^{\times}_{\geq 0}$, and that $\mathfrak{C}((I - B_{G\otimes D}(q^{G\otimes D}))^{-1}) = (I - (B_G(q^G))^{-1}$. Lemma 6.3$(vi)$ relates the norms: $\|(I - B_{G\otimes D}(q^{G\otimes D}))^{-1}\|_\infty \leq d\|(I - (B_G(q^G))^{-1}\|_\infty$. We need a bound on the latter norm. Lemma 6.11 shows $\|(I - B_G(q^G))^{-1}\|_\infty \leq 2^{14|G|+3}$. So $\|(I - B_{G\otimes D}(q^{G\otimes D}))^{-1}\|_\infty \leq d2^{14|G|+3}$. Plugging this bound into Theorem 6.9 yields the result. □

6.3.2 General SCFGs

For any SCFG, G, and corresponding PPS, $x = P_G(x)$, with LFP $q^* > \mathbf{0}$, the *dependency graph*, $H_G = (V, E)$, has the variables (or the nonterminals of G) as nodes and has the following edges: $(x_i, x_j) \in E$ if and only if x_j appears in some monomial in $P_G(x)_i$ with a positive coefficient. We can decompose the dependency graph H_G into its SCCs, and form the DAG of SCCs, H'_G. For each SCC, $\mathcal{S}$, suppose its corresponding equations are $x_{\mathcal{S}} = P_G(x_{\mathcal{S}}, x_{D(\mathcal{S})})_{\mathcal{S}}$, where $D(\mathcal{S})$ is the set of variables $x_j \notin \mathcal{S}$ such that there is a path in H_G from some variable $x_i \in \mathcal{S}$ to x_j. We call a SCC, $\mathcal{S}$, of H_G, a ***critical SCC*** if the PPS $x_{\mathcal{S}} = P_G(x_{\mathcal{S}}, q^G_{D(\mathcal{S})})_{\mathcal{S}}$ is critical. In other words, the SCC $\mathcal{S}$ is critical if we plug in the LFP values q^G into variables that are in lower SCCs, $D(\mathcal{S})$, then the resulting PPS is critical. We note that an arbitrary PPS, $x = P_G(x)$ is non-critical if and only if it has no critical SCC. We define the ***critical depth***, $\mathfrak{c}(G)$, of $x = P_G(x)$ as follows: it is the maximum length, k, of any sequence $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$, of SCCs of H_G, such that for all $i \in \{1, \ldots, k-1\}$, $\mathcal{S}_{i+1} \subseteq D(\mathcal{S}_i)$, and furthermore, such that for all $j \in \{1, \ldots, k\}$, $\mathcal{S}_j$ is critical. Let us call a critical SCC, $\mathcal{S}$, of H_G a ***bottom-critical SCC***, if $D(\mathcal{S})$ does not contain any critical SCCs. As we now show, by using earlier results ([EY09, EGK13]) we can compute in P-time the critical SCCs of a given PPS, and its critical depth.

Let $x = P(x)$ be a PPS (wlog in SNF), with LFP $q^* > 0$, let $B(x)$ be its Jacobian matrix, and let $H = (V, E)$ be its dependency graph. If B is a square matrix and I, J are subsets of indices, we will use $B_{I,J}$ to denote the submatrix with rows in I and columns in J, and we use B_I to denote the square submatrix $B_{I,I}$.

Proposition 6.14. *Given a PPS $x = P(x)$ with LFP $q^* > 0$, we can compute in polynomial time its critical SCCs and its critical depth.*

Proof. We know that for each SCC $\mathcal{S}$ of H, either all the variables (nodes) of the SCC have value 1 in the LFP q^*, or they all have value < 1; moreover, if they have value 1, then so do all the variables that they can reach in H, i.e., $q^*_{\mathcal{S}} = \mathbf{1}$ implies $q^*_{D(\mathcal{S})} = \mathbf{1}$ [EY09]. Furthermore, we can determine which variables and SCCs have value 1, and which value < 1, in polynomial time [EY09] (this was improved to strongly polynomial time in [EGK13]). We also know that $\rho(B(q^*)) \leq 1$, thus a PPS is critical if and only if $\rho(B(q^*)) = 1$. Furthermore, by Theorem 3.6, if $q^* < \mathbf{1}$, then $\rho(B(q^*)) < 1$.

Therefore, for each SCC $\mathcal{S}$, we can determine whether it is critical as follows. If $q^*_{\mathcal{S}} < \mathbf{1}$ then $\mathcal{S}$ is not critical. If $q^*_{\mathcal{S}} = \mathbf{1}$, then $\mathcal{S}$ is critical if and only if $\rho(B(\mathbf{1})_{\mathcal{S}}) = 1$, and it is not critical if and only if $\rho(B(\mathbf{1})_{\mathcal{S}}) < 1$; we can determine which of the two is the case as follows. Since the spectral radius of $B(\mathbf{1})_{\mathcal{S}}$ is at most 1, $\rho(B(\mathbf{1})_{\mathcal{S}}) = 1$ if and only if there is a vector $u \neq 0$ such that $(B(\mathbf{1})_{\mathcal{S}}) \cdot u = u$ (and we can take $u \geq 0$ to be an eigenvector for the eigenvalue 1 in this case since the matrix is nonnegative), or equivalently since the constraints are homogeneous in u, this is the case if and only if the set of linear equations $\{(B(\mathbf{1})_{\mathcal{S}}) \cdot u = u; \sum_i u_i = 1\}$ has a solution. This can be checked in (strongly) polynomial time by standard methods.

Once we have identified the critical SCCs, it is straightforward to compute the critical depth in linear time in the size of the DAG of SCCs by a traversal of the DAG in topological order. □

Proposition 6.15. *A PPS $x = P(x)$ is critical if and only if at least one of its SCCs is critical.*

Proof. (Only if): Suppose first that the PPS is critical, i.e., that $\rho(B(q^*)) = 1$. Let $v \geq 0$, $v \neq 0$, be an eigenvector of $B(q^*)$ for the eigenvalue 1, i.e., $B(q^*)v = v$. Let $\mathcal{S}$ be a lowest SCC that contains a variable with nonzero value in v, i.e. $v_{\mathcal{S}} \neq 0$ and $v_{D(\mathcal{S})} = 0$. Then $v_{\mathcal{S}} = B(q^*)_{\mathcal{S},\mathcal{S}\cup D(\mathcal{S})} \cdot v_{\mathcal{S}\cup D(\mathcal{S})} = B(q^*)_{\mathcal{S}} \cdot v_{\mathcal{S}}$. Thus, $v_{\mathcal{S}}$ is an eigenvector of $B(q^*)_{\mathcal{S}}$ with eigenvalue 1, hence $\rho(B(q^*)_{\mathcal{S}}) \geq 1$, and since we always have $\rho(B(q^*)_{\mathcal{S}}) \leq 1$, if follows that $\mathcal{S}$ is a critical SCC.

(If): Conversely, suppose that there is a critical SCC, and let $\mathcal{S}$ be a highest critical SCC in the DAG of SCC's. Then $\rho(B(q^*)_{\mathcal{S}}) = 1$. Let $u \geq 0$ be an eigenvector of $B(q^*)_{\mathcal{S}}$ with eigenvalue 1. Let $E(\mathcal{S})$ be the (possibly empty) set of variables which depend on variables in $\mathcal{S}$ but are not themselves in $\mathcal{S}$. If $E(\mathcal{S}) = \emptyset$ then let v be a vector with $v_{\mathcal{S}} = u$ and $v_i = 0$ for all variables $x_i \notin \mathcal{S}$. Then $B(q^*)v = v$, i.e., v is an eigenvector of $B(q^*)$ with eigenvalue 1, hence $\rho(B(q^*)) \geq 1$ and the PPS is critical.

Suppose that $E(\mathcal{S})$ is nonempty. Then $E(\mathcal{S})$ contains no critical SCCs by our choice of $\mathcal{S}$. This implies by our proof above for the (only if) direction that the PPS $x_{E(\mathcal{S})} = P(x_{E(\mathcal{S})}, x_{D(E(\mathcal{S}))})$ is not critical, i.e., $\rho(B(q^*)_{E(\mathcal{S})}) < 1$. Thus, $(I - B(q^*)_{E(\mathcal{S})})^{-1}$ exists. Let v be the vector with $v_{\mathcal{S}} = u$, $v_{E(\mathcal{S})} = (I - B(q^*)_{E(\mathcal{S})})^{-1} B(q^*)_{E(\mathcal{S}),\mathcal{S}} \cdot v_{\mathcal{S}}$ and $v_i = 0$ for all x_i not in either $\mathcal{S}$ or $E(\mathcal{S})$.

We claim that $B(q^*)v = v$. If x_i does not depend on a variable in $\mathcal{S}$, then any x_j which x_i depends on also does not depend on $\mathcal{S}$ and so has $v_j = 0$. So $(B(q^*)v)_i = 0 = v_i$. Next we consider $(B(q^*)v)_{\mathcal{S}}$. Since $D(\mathcal{S})$ is disjoint from $\mathcal{S}$ and $E(\mathcal{S})$, $v_{D(\mathcal{S})} = 0$. So $(B(q^*)v)_{\mathcal{S}} = (B(q^*))_{\mathcal{S}} \cdot v_{\mathcal{S}} = v_{\mathcal{S}}$. Lastly consider $(B(q^*)v)_{E(\mathcal{S})}$.

$$\begin{aligned}
(B(q^*)v)_{E(\mathcal{S})} &= B(q^*)_{E(\mathcal{S})} \cdot v_{E(\mathcal{S})} + B(q^*)_{E(\mathcal{S}),\mathcal{S}} \cdot v_{\mathcal{S}} \\
&= v_{E(\mathcal{S})} - (I - B(q^*)_{E(\mathcal{S})}) \cdot v_{E(\mathcal{S})} + B(q^*)_{E(\mathcal{S}),\mathcal{S}} \cdot v_{\mathcal{S}} \\
&= v_{E(\mathcal{S})} - B(q^*)_{E(\mathcal{S}),\mathcal{S}} \cdot v_{\mathcal{S}} + B(q^*)_{E(\mathcal{S}),\mathcal{S}} \cdot v_{\mathcal{S}} \\
&= v_{E(\mathcal{S})}
\end{aligned}$$

So $B(q^*)v = v$. Therefore, $\rho(B(q^*) \geq 1$ and hence the PPS is critical. □

PPSs with nested critical SCCs are hard to analyse directly. It turns out we can circumvent this by perturbng the probabilities in the SCFG G to obtain an SCFG G' with no critical SCCs, and showing that the perturbations are small enough so that they do not change the probabilities of interest by much. Concretely:

Theorem 6.16. *For any $\varepsilon > 0$, and for any SCFG, G, in SNF form, with $q^G > 0$, with critical depth $\mathfrak{c}(G)$, consider the new SCFG, G', obtained from G by the following process: for each bottom-critical SCC, $\mathcal{S}$, of $x = P_G(x)$, find any rule $r = A \xrightarrow{p} B$ of G, such that A and B are both in $\mathcal{S}$ (since G is in SNF, such a rule must exist in every critical SCC). Reduce the probability p, by setting it to $p' = p(1 - 2^{-(14|G|+3)2^{\mathfrak{c}(G)}} \varepsilon^{2^{\mathfrak{c}(G)}})$. Do this for all bottom-critical SCCs. This defines G', which is non-critical. Using G' instead of G, if we apply R-NM, with parameter $h+2$ to approximate the LFP $q^{G'\otimes D}$ of MPS $y = P_{G'\otimes D}(y)$, then $\|q^{G\otimes D} - x^{[h+1]}\|_\infty \leq \varepsilon$ where $h := \lceil \log d + (3 \cdot 2^{\mathfrak{c}(G)} + 1)(\log(1/\varepsilon) + 14|G| + 3) \rceil$.*
Thus we can compute $q_A^{G,D} = \sum_{t \in F} q_{s_0At}^{G\otimes D}$ within additive error $\delta > 0$ in time polynomial in: $|G|$, $|D|$, $\log(1/\delta)$, and $2^{\mathfrak{c}(G)}$, in the Turing model of computation.

The proof is very involved, and we will have to develop it in several steps, using some preliminary Lemmas and Theorems.

Let us first mention that, in Section 6.5, we shall give a family of SCFGs, and a 3-state DFA that checks the infix probability of the string *aa*, and we shall explain why those examples indicate it will likely be difficult to overcome the exponential dependence on the critical-depth $\mathfrak{c}(G)$ in the above bounds.

We now start the developments needed towards a proof of Theorem 6.16. To deal with critical SCCs, we need a way to analyse how an error in the LFP q^* inside one

SCC, $\mathcal{S}$, where $q^*_{\mathcal{S}} = 1$, affects those SCCs that depend on it. Later we will use this to show how a perturbation of the bottom-critical SCCs affects all SCCs (Theorem 6.20).

Theorem 6.17. *Given a PPS, $y = P(y)$ in SNF form, such that for a subvector x of y, whose equations are $x = P(x, y_{D(x)})$, when restricting $y = P(y)$ to the variables in x, and if we let $y_{D(x)} := z$, for a real-valued vector $0 \leq z < \mathbf{1}$, and if the resulting PPS, $x = P(x,z)$ has LFP $q^*_z > 0$, and if $q^*_{\mathbf{1}}$ is the LFP solution of $x = P(x,\mathbf{1})$ (note that $q^*_{\mathbf{1}} \geq q^*_z$), then:*

(i) If $q^*_{\mathbf{1}} < \mathbf{1}$ then, $\|q^*_{\mathbf{1}} - q^*_z\|_\infty \leq 2^{14|P|+2}\|\mathbf{1} - z\|_\infty$

(ii) If the PPS $x = P(x,\mathbf{1})$ is strongly connected and $q^*_{\mathbf{1}} = \mathbf{1}$ then

$$\|\mathbf{1} - q^*_z\|_\infty \leq 2^{3|P|}\sqrt{\|\mathbf{1} - z\|_\infty}.$$

(iii) If the PPS, $x = P(x,1)$, is strongly connected and $q^*_{\mathbf{1}} = \mathbf{1}$, and $\rho(B(\mathbf{1},\mathbf{1})) < 1$ then $\|\mathbf{1} - q^*_z\|_\infty \leq 2^{3|P|}\|\mathbf{1} - z\|_\infty$.

Bad examples of PPSs, given in [EKL10], and in section 5.5.1, show that there are critical PPSs with $q^*_1 = 1$, and with $\|\mathbf{1} - q^*_z\|_\infty \geq \sqrt{\|\mathbf{1} - z\|_\infty}$. Thus we cannot hope to get a bound linear in $\|\mathbf{1} - z\|_\infty$ in all cases. Cases (i) and (iii) of Theorem 6.17 say that we can get a linear bound *except for* critical PPSs, where we indeed need a square root in the strongly connected case (case (ii)).

Proof of Theorem 6.17. We first prove the following:

Lemma 6.18. *For $0 \leq z \leq z' \leq \mathbf{1}$, and for all $0 \leq x \leq 1$, $\|P(x,z') - P(x,z)\|_\infty \leq 2\|z - z'\|_\infty$*

Proof. Consider the k'th coordinate, $P(x,y)_k$, of the PPS polynomials $P(x,y)$, in SNF form. We distinguish cases based on the type of x_k. If x_k has type $\mathtt{Q}$: then $P(x,z)_k$ and $P(x,z')_k$ both have the form $x_i x_j$, or both have form $z^{(\prime)}_i x_j$, or both the form $x_i z^{(\prime)}_j$, or both the form $z^{(\prime)}_i z^{(\prime)}_j$. Thus, since $0 \leq z \leq z' \leq 1$, and $0 \leq x \leq 1$, we have $0 \leq P(x,z')_k - P(x,z)_k \leq z'_i z'_j - z_i z_j \leq 2\|z - z'\|_\infty$.

In the case where x_k has type $\mathtt{L}$, we have $0 \leq P(x,z')_k - P(x,z)_k \leq \sum_j p_{k,j}(z'_j - z_j) \leq \|z - z'\|_\infty$, because the coefficients $p_{k,j}$ of the type $\mathtt{L}$ equation must sum to ≤ 1.

Finally, if x_k has type $\mathtt{T}$, $P(x,z)_k$ and $P(x,z')_k$ are equal constants, so their difference is 0. □

Lemma 6.19. *If $x = P(x,z)$ is a PPS with LFP $q^*_z > 0$ and $x = P(x,z')$ has LFP $q^*_{z'} > 0$ for some $0 \leq z \leq z' \leq \mathbf{1}$, and $(I - B(\frac{1}{2}(q^*_{z'} + q^*_z), z'))$ is non-singular then*

$$\|q^*_{z'} - q^*_z\|_\infty \leq 2\|(I - B(\tfrac{1}{2}(q^*_{z'} + q^*_z), z'))^{-1}\|_\infty \|z' - z\|_\infty$$

Proof. From Lemma 4.30 , applied to the PPS $x = P(x,z')$, (where we let $y := q^*_z$), we have:

$$(q^*_{z'} - q^*_z) = (I - B(\frac{1}{2}(q^*_{z'} + q^*_z), z'))^{-1}(P(q^*_z, z') - q^*_z)$$

We can take norms:

$$\|q^*_{z'} - q^*_z\|_\infty = \|(I - B(\tfrac{1}{2}(q^*_{z'} + q^*_z), z'))^{-1}\|_\infty \|(P(q^*_z, z') - q^*_z)\|_\infty$$

Now we just apply Lemma 6.18, to obtain that $\|(P(q^*_z, z') - q^*_z)\|_\infty \leq 2\|z' - z\|_\infty$. □

To get parts (i) and (ii) of Theorem 6.17, we apply Theorem 3.24. For establishing (i) of Theorem 6.17, we need to apply (i) of Theorem 3.24 to the PPS, $x = P(x, \mathbf{1})$, with $y := q^*_z$. This gives

$$\|(I - B(\tfrac{1}{2}(q^*_z + q^*_{\mathbf{1}}), \mathbf{1}))^{-1}\|_\infty \leq 2^{10|P|} \max\{2(\mathbf{1} - q^*_z)^{-1}_{\min}, 2^{|P|}\}$$

Now, since in part (i) of Theorem 6.17, we are given that $q^*_{\mathbf{1}} < 1$, we know that $q^*_z \leq q^*_{\mathbf{1}} \leq \mathbf{1} - 2^{-4|P|}\mathbf{1}$, by Theorem 3.14 . So we have

$$\|(I - B(\tfrac{1}{2}(q^*_z + q^*_{\mathbf{1}}), \mathbf{1}))^{-1}\|_\infty \leq 2^{14|P|+1}$$

Lemma 6.19 now tells us that:

$$\|q^*_{\mathbf{1}} - q^*_z\|_\infty \leq 2^{14|P|+2}\|\mathbf{1} - z\|_\infty$$

This finishes the proof of part (i) of Theorem 6.17.

To prove part (ii) of Theorem 6.17, first remember that we assume $x = P(x, \mathbf{1})$ is strongly connected. We use part (ii) of Theorem 3.24.

By assumption, $q^*_{\mathbf{1}} = \mathbf{1}$. We take $z = \frac{1}{2}(\mathbf{1} + q^*_y)$, giving:

$$\|(I - B(\tfrac{1}{2}(\mathbf{1} + q^*_z), \mathbf{1}))^{-1}\|_\infty \leq 2^{4|P|} \frac{2}{(\mathbf{1} - q^*_z)_{\min}} \tag{6.1}$$

Now

$$\begin{aligned} B(\frac{1}{2}\mathbf{1}, \mathbf{1})(\mathbf{1} - q^*_z) &\leq B(\frac{1}{2}(\mathbf{1} + q^*_z), \mathbf{1})(\mathbf{1} - q^*_z) \\ &= P(\mathbf{1}, \mathbf{1}) - P(q^*_z, \mathbf{1}) \quad \text{(by Lemma 3.3)} \\ &\leq P(\mathbf{1}, \mathbf{1}) - P(q^*_z, z) = \mathbf{1} - q^*_z \end{aligned}$$

Now we apply Lemma 6.8, letting v be $\mathbf{1} - q^*_z$ in the statement of that Lemma, and considering $B(\frac{1}{2}\mathbf{1}, \mathbf{1})$ in place of the $B(\frac{1}{2}\mathbf{1})$ in the statement of the Lemma. This tells us that $\frac{\|\mathbf{1} - q^*_z\|_\infty}{(\mathbf{1} - q^*_z)_{\min}} \leq 2^{|P|}$.

Now, if we substitute this into the equation (6.1), we get

$$\|(I - B(\tfrac{1}{2}(1 + q^*_z), \mathbf{1}))^{-1}\|_\infty \leq 2^{5|P|+1} \frac{1}{\|\mathbf{1} - q^*_z\|_\infty}$$

Lemma 6.19 now gives:

$$\|\mathbf{1} - q^*_z\|_\infty \leq 2\|(I - B(\tfrac{1}{2}(\mathbf{1} + q^*_z), \mathbf{1}))^{-1}\|_\infty \|1 - z\|_\infty$$

Inserting our bound for the norm of $(I - B(\frac{1}{2}(\mathbf{1} + q^*_z), \mathbf{1}))^{-1}$ gives:

$$\|\mathbf{1} - q^*_z\|_\infty \leq 2^{5|P|+2} \frac{1}{\|\mathbf{1} - q^*_z\|_\infty} \|1 - z\|_\infty$$

re-arranging and taking the square root gives:

$$\|\mathbf{1} - q^*_z\|_\infty \leq \sqrt{2^{5|P|+2} \|\mathbf{1} - z\|_\infty}$$

As long as the encoding size is $|P| \geq 2$, which we can clearly assume, we have:

$$\|\mathbf{1} - q^*_z\|_\infty \leq 2^{3|P|} \sqrt{\|\mathbf{1} - z\|_\infty}$$

For part (iii), the significance of the condition that $\rho(B(\mathbf{1}, \mathbf{1})) < 1$ is that it implies $(I - B(\mathbf{1}, \mathbf{1}))^{-1}$ exists, and $(I - B(\mathbf{1}, \mathbf{1}))^{-1} \geq (I - B(\frac{1}{2}(\mathbf{1} + q^*_z), 1)$. So, we use a bound on $\|(I - B(\mathbf{1}, \mathbf{1}))^{-1}\|_\infty$:

Lemma 6.19 gives:

$$\|\mathbf{1} - q^*_z\|_\infty \leq 2\|(I - B(\tfrac{1}{2}(\mathbf{1} + q^*_z), \mathbf{1}))^{-1}\|_\infty \|\mathbf{1} - z\|_\infty$$

Now $\|(I - B(\frac{1}{2}(\mathbf{1} + q^*_z), \mathbf{1}))^{-1}\|_\infty \leq \|(I - B(\mathbf{1}, \mathbf{1}))^{-1}\|_\infty$. We can apply Lemma 6.12 on the PPS $x = P(x, \mathbf{1})$, which yields $\|(I - B(\mathbf{1}, \mathbf{1}))^{-1}\|_\infty \leq 2^{3|P|}$. Now we have

$$\|\mathbf{1} - q^*_z\|_\infty \leq 2^{3|P|} \|\mathbf{1} - z\|_\infty$$

as required. □

The following Theorem shows the effect of perturbing a PPS with positive critical depth. We need to get a bound on how much the LFP changes and a bound on $\|(I - B_\delta(q^*))^{-1}\|_\infty$ for the perturbed PPS which quantifies how far from critical it is.

Theorem 6.20. *Suppose $x = P(x)$ is a PPS in SNF form that has critical depth at most $\mathfrak{c}$. Let $\delta \in \mathbb{R}$, such that $0 \leq \delta \leq 2^{-3|P|-1}$. Suppose that in every bottom-critical SCC of $x = P(x)$ we reduce a single positive coefficient, p, by setting it to $p' = p(1-\delta)$, resulting in the PPS $x = P_\delta(x)$. Then $\|q^* - q^*_\delta\|_\infty \leq 2^{14|P|+2}\delta^{(1/2^{\mathfrak{c}})}$ where q^* and q^*_δ are the LFP solutions of $x = P(x)$ and $x = P_\delta(x)$, respectively. Furthermore, $\|(I - B_\delta(q^*_\delta))^{-1}\|_\infty \leq 2^{8|P|+2}\delta^{-3}$.*

Proof. If $\mathfrak{c} = 0$, we have no critical SCCs, so we don't change any coefficients, and $q^* = q^*_\delta$, and the remaining claim about $\|(I - B_\delta(q^*_\delta))^{-1}\|_\infty$ follows directly from Lemma 6.11.

So, we can assume $\mathfrak{c} > 0$ in the rest of the proof. To establish that q^* and q^*_δ are close, we will use Theorem 6.17. For any SCC, S, of a PPS $x = P(x)$, either $q^*_S = \mathbf{1}$ or $q^*_S < \mathbf{1}$, because every variable in S depends (directly or indirectly) on every other, so if any of them are < 1, then so are all the others.

Let S be an SCC with $q^*_S = \mathbf{1}$ and with $(q^*_\delta)_S < \mathbf{1}$. The SCC S necessarily only depends on SCCs, T, with $q^*_T = 1$, because otherwise we wouldn't have $q^*_S = \mathbf{1}$. We want to show that

$$\|\mathbf{1} - (q^*_\delta)_S\|_\infty \leq \delta^{(1/2^{\mathfrak{c}_{S\cup D(S)}})} \cdot 2^{6|P_{S\cup D(S)}|}$$

where $\mathfrak{c}_{S\cup D(S)}$ is the critical depth in $x_{S\cup D(S)} = P_{S\cup D(S)}(x_{S\cup D(S)})$, and $|P_{S\cup D(S)}|$ denotes the encoding size of the latter PPS. To prove this by induction, we can assume

$$\|\mathbf{1} - (q^*_\delta)_{D(S)}\|_\infty \leq \delta^{(1/2^{\mathfrak{c}_{D(S)}})} \cdot 2^{6|P_{D(S)}|} \tag{6.2}$$

The base case is when S is a bottom-critical SCC, that does not depend on any other critical SCCs. Then even if $D(S)$ is non-empty, $q^*_{D(S)} = (q^*_\delta)_{D(S)}$. However, we do change a single coefficient p in S, by setting it to $p' = p(1-\delta)$. Note that because the PPS is in SNF form, p must appear in a equation $x_i = P(x_S, \mathbf{1})_i$ where x_i is of type $\mathtt{L}$, and thus the coefficient p appears in a single term px_j. We wish to consider a new PPS in SNF form, parametrized by the possible values $z \in \{(1-\delta), 1\}$ that we multiply p by. To do this, we can simply add a new variable x_{n+1} (for this particular SCC, S), and we then replace the term px_j by px_{n+1}, and we add a new equation $x_{n+1} = zx_j$ to our system of equations. We denote this new PPS by $(x_S, x_{n+1}) = Q_S((x_S, x_{n+1}), z)$. Note that this is indeed a SNF form PPS for either $z \in \{(1-\delta), 1\}$. Note also that in terms of encoding size, we have $|Q_S| \leq 2|P_S|$.

The LFP solution of $(x_S, x_{n+1}) = Q_S((x_S, x_{n+1}), 1)$, in the S coordinates has $q^*_S = \mathbf{1}$, and the LFP solution of $(x_S, x_{n+1}) = Q_S((x_S, x_{n+1}), (1-\delta))$ in the S coordinates is

$(q^*_\delta)_S$. Thus, by Theorem 6.17 (ii), we get $\|\mathbf{1}-(q^*_\delta)_S\|_\infty \leq 2^{3|Q_S|}\sqrt{\delta} \leq 2^{6|P_S|}\sqrt{\delta}$. In this case $\mathfrak{c}_{S\cup D(S)} = 1$ so this is enough to establish the inductive claim in inequality (6.2).

Next, suppose that S is a critical SCC that depends on a different critical SCC. q^*_S is the LFP solution of $x_S = P_S(x_S, q^*_{D(S)})$ and $(q^*_\delta)_S$ is the LFP solution of $x_S = P_S(x_S,(q^*_\delta)_{D(S)})$. By Theorem 6.17 (ii), $\|\mathbf{1}-(q^*_\delta)_S\|_\infty \leq 2^{3|P_S|}\sqrt{\|\mathbf{1}-(q^*_\delta)_{D(S)})\|_\infty}$. Substituting using the inductive assumption in inequality (6.2) gives:

$$\begin{aligned}
\|\mathbf{1}-(q^*_\delta)_S\|_\infty &\leq 2^{3|P_S|}\sqrt{\|\mathbf{1}-(q^*_\delta)_{D(S)}\|_\infty} \\
&\leq 2^{3|P_S|}\sqrt{\delta^{(1/2^{\mathfrak{c}_{D(S)}})}2^{6|P_{D(S)}|}} \\
&= 2^{3|P_S|+\frac{6}{2}|P_{D(S)}|}\delta^{(1/2^{\mathfrak{c}_{D(S)}+1})} \\
&\leq \delta^{(1/2^{\mathfrak{c}_{S\cup D(S)}})}2^{3|P_{S\cup D(S)}|}
\end{aligned}$$

The last inequality holds because $\mathfrak{c}_{S\cup D(S)} = \mathfrak{c}_{D(S)}+1$. This is because S is itself a critical SCC. Note also that $|P_{S\cup D(S)}| = |P_S| + |P_{D(S)}|$ since $x_S = P(x_S, x_{D(S)})_S$ and $x_{D(S)} = P(x_{D(S)})_{D(S)}$ are disjoint subsets of the equations in $x = P(x)$.

Finally suppose that S is not a critical SCC but does have $q^*_S = 1$ and depends on some critical SCC. Again q^*_S is the LFP solution of $x_S = P_S(x_S, q^*_{D(S)})$ and $(q^*_\delta)_S$ is the LFP solution of $x_S = P_S(x_S,(q^*_\delta)_{D(S)})$. By Theorem 6.17 (iii): $\|\mathbf{1}-(q^*_\delta)_S\|_\infty \leq 2^{3|P_S|}\|\mathbf{1}-(q^*_\delta)_{D(S)})\|_\infty$. Substituting the inductive assumption (6.2) gives $\|\mathbf{1}-(q^*_\delta)_S\|_\infty \leq 2^{3|P_S|+6|P_{D(S)}|}\delta^{(1/2^{\mathfrak{c}_{D(S)}})}$ which simplifies to $\|\mathbf{1}-(q^*_\delta)_S\|_\infty \leq \delta^{(1/2^{\mathfrak{c}_{S\cup D(S)}})}2^{6|P_{S\cup D(s)}|}$. This is because S itself is non-critical, so $\mathfrak{c}_{D(S)} = \mathfrak{c}_{S\cup D(S)}$.

Let A (for "always") denote the set of variables x_i for which $q^*_i = 1$, and let M (for "maybe") denote the set of variables x_i for which $0 < q^*_i < 1$. A is non-empty as otherwise we would have no critical SCCs. Every variable x_i in A is part of some SCC S with $q^*_S = 1$. So our induction has already given that

$$\|\mathbf{1}-(q^*_\delta)_A\|_\infty \leq \delta^{1/2^{\mathfrak{c}}}2^{6|P_A|}$$

If M is empty, this bound on $\|q^* - q^*_\delta\|_\infty$ is enough. Otherwise we have to use Theorem 6.17 (i). This gives that $\|q^*_M - (q^*_\delta)_M\|_\infty \leq 2^{14|P_M|+2}\|\mathbf{1}-(q^*_\delta)_A\|_\infty$. Substituting gives $\|q^*_M - (q^*_\delta)_M\|_\infty \leq 2^{14|P|+2}\delta^{1/2^{\mathfrak{c}}}$. We have now shown that

$$\|q^* - q^*_\delta\|_\infty \leq 2^{14|P|+2}\delta^{1/2^{\mathfrak{c}}}$$

The only thing left to complete the proof of Theorem 6.20 is to get a bound on $\|(I-B_\delta(q^*_\delta))^{-1}\|_\infty$. For this we will use the techniques of the proof of Theorem

6.10. Call the set of variables for which $(q^*_\delta)_i = 1$, A_δ and the set of variables x_i for which $0 < (q^*_\delta)_i < 1$, M_δ. Since $q^*_\delta \leq q^*$, $M \subseteq M_\delta$ and $A_\delta \subseteq A$. It is worth noting that variables belonging to critical SCCs are in $A \cap M_\delta$. We will first show that if a variable x_i depends (directly or indirectly) on some variable x_j for which we have reduced a coefficient in $P_\delta(x)_j$, then $(q^*_\delta)_i \leq 1 - 2^{-|P|}\delta$. For any such x_i, consider a *shortest* sequence $x_{l_1}, x_{l_2}, \dots, x_{l_m}$, such that (1): $l_1 = j$ and $P_\delta(x)_j$ has a reduced coefficient in it, (2): $l_m = i$, and (3): for every $0 \leq k < m$, $P_\delta(x)_{l_{k+1}}$ contains a term with x_{l_k}. There is some term $p_{j,h}x_h$ in $P_j(x)$ which has been changed to $p_{j,h}(1-\delta)x_h$ in $P_\delta(x)_j$. Since $x = P(x)$ is a PPS, $P(\mathbf{1})_j \leq 1$, but note that $P_\delta(x)_j$ is not proper, as indeed we must have that $P_\delta(\mathbf{1})_j \leq P(\mathbf{1})_j - p_{j,h}\delta \leq 1 - p_{j,h}\delta$. Also note that $(q^*_\delta)_j = P_\delta(q^*_\delta)_j \leq P_\delta(\mathbf{1})_j \leq 1 - p_{j,h}\delta$. For any $0 \leq k < m$, if $x_{l_{k+1}}$ has type $\mathtt{Q}$, then $(q^*_\delta)_{l_{k+1}} \leq (q^*_\delta)_{l_k}$. If $x_{l_{k+1}}$ has type $\mathtt{L}$, then $1 - (q^*_\delta)_{l_{k+1}} \geq p_{l_{k+1},l_k}(1 - (q^*_\delta)_{l_k})$. By an easy induction $1 - (q^*_\delta)_i \geq (\prod_{\{k | x_{l_k} \text{ has Type L}\}} p_{l_{k+1},l_k})(1 - (q^*_\delta)_j)$. Thus:

$$1 - (q^*_\delta) \geq (\prod_{\{k | x_{l_k} \text{ has Type L}\}} p_{l_{k+1},l_k}) p_{j,h}\delta$$

Since this is the shortest sequence satisfying the stated conditions, for any $0 \leq k < m$, $P_\delta(x)_{l_k}$ has not had any coefficients reduced, and furthermore the x_{l_k}'s are all distinct variables. So all these coefficients p_{l_{k+1},l_k} and $p_{j,h}$ are distinct coefficients in $x = P(x)$. The encoding size $|P|$ is at least the number of bits describing these rationals p_{l_{k+1},l_k} and $p_{j,h}$ and thus

$$(q^*_\delta)_i \leq 1 - 2^{-|P|}\delta$$

Next we show that the PPS $x = P_\delta(x)$ is non-critical. Suppose, for a contradiction that $x = P_\delta(x)$ is critical. Then it has some critical SCC S. But then S must have also been an SCC in the PPS $x = P(x)$, because the dependency graphs of these PPSs are the same (we never reduce a positive probability to 0). For S to be a critical SCC in $x = P_\delta(x)$, we must have that $(q^*_\delta)_S = \mathbf{1}$ and $\rho(B_\delta(\mathbf{1})_S) = 1$. However, $q^* \geq q^*_\delta$ and $\rho(B(\mathbf{1})_S) \geq \rho(B_\delta(\mathbf{1})_S) = 1$. So $q^*_S = \mathbf{1}$. Lemma 6.5 of [EY09] shows that for any strongly connected PPS, $x = P(x)$, with Jacobian $B(x)$, and with LFP, q^*, if $x < q^*$, then $\rho(B(x)) < 1$. Thus, by continuity of eigenvalues, $\rho(B(q^*)) \leq 1$. Applying this to the strongly connected PPS $x_S = P(x_S, \mathbf{1})_S$, since $q^*_S = \mathbf{1}$, we get $\rho(B(\mathbf{1})_S) \leq 1$. Thus $\rho(B(\mathbf{1})_S) = 1$ i.e. S is a critical SCC of $x = P(x)$. Either S is a bottom-critical-SCC or it depends on some bottom-critical-SCC. So every variable x_i in S depends on some variable x_j for which we have reduced a coefficient in $P_\delta(x)_j$. So for every x_i in S, $q^*_i \leq 1 - 2^{-|P|}\delta$. But this contradicts our earlier assertion that $q^*_S = \mathbf{1}$.

$B_\delta(q^*_\delta)$ has the block decomposition $B_\delta(q^*_\delta) = \begin{pmatrix} B_\delta(q^*_\delta)_{M_\delta} & B_\delta(q^*_\delta)_{M_\delta,A_\delta} \\ 0 & B_\delta(q^*_\delta)_{A_\delta} \end{pmatrix}$.

It is possible that A_δ is empty, in which case the bound we will obtain on $\|(I - B_\delta(q^*_\delta)_{M_\delta})^{-1}\|_\infty$ will be enough to show the theorem. So we suppose here that A_δ is non-empty. M_δ is non-empty since we assumed that we have at least one critical SCC.

We need to show that both $I - B_\delta(q^*_\delta)_{M_\delta}$ and $I - B_\delta(q^*_\delta)_{A_\delta}$ are nonsingular, and we need to get upper bounds on $\|(I - B_\delta(q^*_\delta)_{M_\delta})^{-1}\|_\infty$ and $\|(I - B_\delta(q^*_\delta)_{A_\delta})^{-1}\|_\infty$. Once we do so, we can then apply Lemma 6.13 to get a bound on $\|(I - B(q^*_\delta))^{-1}\|_\infty$.

First, let us show that $I - B_\delta(q^*_\delta)_{A_\delta}$ is non-singular, and also bound $\|(I - B_\delta(q^*_\delta)_{A_\delta})^{-1}\|_\infty$.

We note that $P(x)_{A_\delta} = P_\delta(x)_{A_\delta}$. We have shown that any variable x_i for which we have reduced a coefficient in $P_\delta(x)_i$ has $q^*_i \leq 1 - 2^{-|P|}\delta$ and so x_i is not in A_δ. Thus the equations in $x_{A_\delta} = P_\delta(x_{A_\delta})_{A_\delta})$ are a subset of the equations $x = P(x)$ and so the encoding size of this PPS is at most $|P|$. We have also shown that the PPS $x = P_\delta(x)$ is non-critical. So we can apply Lemma 6.12 to the PPS $x_{A_\delta} = P_\delta(x_{A_\delta})_{A_\delta})$, which gives $\|(I - B_\delta(q^*_\delta)_{A_\delta})^{-1}\|_\infty \leq 2^{3|P|}$.

Now, let us show that $I - B_\delta(q^*_\delta)_{M_\delta}$ is non-singular, and also bound $\|(I - B_\delta(q^*_\delta)_{M_\delta})^{-1}\|_\infty$.

Consider the PPS, restricted to the variables in M_δ. Note that no variable in A_δ can depend on these. Thus, restricting the PPS $x = P_\delta(x)$ to the variables in M_δ defines a PPS $x_{M_\delta} = P_\delta(x_{M_\delta}, \mathbf{1})_{M_\delta}$. Note that the LFP of this is $(q^*_\delta)_{M_\delta} < 1$, by definition of M_δ. To simplify notation in the current argument, we shall denote this PPS by $y = R(y)$, and we shall use $r^* := (q^*_\delta)_{M_\delta}$ to denote its LFP. Furthermore, let us use $B_R(y)$ to denote its Jacobian. We note, firstly, that $B_R(r^*) = B_\delta(q^*_\delta)_{M_\delta}$. The way to see this is to note that $q^*_\delta = (r^*, \mathbf{1})$ and so the entries of both matrices are $\frac{\partial (P_\delta)_i}{\partial x_j}(q^*_\delta)$ for $x_i, x_j \in M_\delta$.

So, rephrased, we want to show $\rho(B_R(r^*)) < 1$, and we want to find a bound on $(I - B_R(r^*))^{-1}$. To do this, we need to follow the proof of Theorem 3.24 (i) in the case $y = r^*$.

We need to use Lemma 3.27, with $A = B_R(r^*)$ and $u = \mathbf{1} - r^*$. By Lemma 3.5 , $B_R(r^*)(\mathbf{1} - r^*) \leq \mathbf{1} - r^*$. We want to find any β so that condition (I) of Lemma 3.27 applies to variables y_i such that either y_i has type Q or else $R(1)_i < 1$. Namely for such variables y_i, it should be the case that $(B_R(r^*)(\mathbf{1} - r^*))_i \leq (1 - \beta)(\mathbf{1} - r^*)_i$.

Let us first note that, for any y_i, $r^*_i \leq 1 - 2^{|P|}\delta$. We have shown that if a variable x_i depends on some variable x_j for which we have reduced a coefficient in $P_\delta(x)_j$, then $(q^*_\delta)_i \leq 1 - 2^{-|P|}\delta$. If $x_i \in M_\delta$ depends on no such variables, then $x_i \in M$. But then we have $q^*_i \leq 1 - 2^{-4|P|} \leq 1 - 2^{-|P|}\delta$ because we assumed that $\delta \leq 2^{-3|P|}$. So for any $x_i \in M_\delta$, $(q^*_\delta)_i \leq 1 - 2^{-|P|}\delta$.

In the case where $y_i = R(y)_i$ has form $\mathtt{Q}$, for some y_j, y_k, $R(y)_i = y_j y_k$ and so

$$\begin{aligned}
B_r(r^*)(\mathbf{1}-r^*))_i &= r_j^*(1-r_k^*)+r_k^*(1-r_j^*) \\
&= r_j^*+r_k^*-2r_j^*r_k^* \\
&= (1-r_j^*r_k^*)-(1+r_j^*r_k^*-r_j^*-r_k^*) \\
&= (1-r_i^*)-(1-r_k^*)(1-r_j^*) \\
&= (1-r_i^*)-\frac{1}{2}((1-r_k^*)(1-r_j^*)+(1-r_j^*)(1-r_k^*)) \\
&\leq (1-r_i^*)-\frac{1}{2}2^{-|P|}\delta((1-r_j^*)+(1-r_k^*)) \\
&\leq (1-r_i^*)-\frac{1}{2}2^{-|P|}\delta((1-r_j^*)+(1-r_k^*)-(1-r_j^*)(1-r_k^*)) \\
&= (1-r_i^*)-\frac{1}{2}2^{-|P|}\delta(1-r_i^*) \\
&= (1-\frac{1}{2}2^{-|P|}\delta)(1-r_i^*)
\end{aligned}$$

Some variables x_i with $P_\delta(\mathbf{1})_i < 1$ have $P(\mathbf{1})_i < 1$, in which case $P(\mathbf{1})_i \leq 1-2^{|P|}$. If a variable x_i has $P_\delta(\mathbf{1})_i < 1$ but $P(\mathbf{1})_i = 1$ then we have reduced some coefficient in $P_\delta(x)_i$ by multiplying it by $1-\delta$ so we have $P_\delta(\mathbf{1})_i \leq 2^{-|P|}\delta$. So for any y_i with $R(\mathbf{1})_i < 1$, $R(\mathbf{1})_i \leq 2^{-|P|}\delta$. So if $R(\mathbf{1})_i < 1$,

$$\begin{aligned}
(B_R(r^*)(\mathbf{1}-r^*))_i &\leq (B_R(\frac{1}{2}(\mathbf{1}+r^*))(\mathbf{1}-r^*))_i \\
&\leq (R(\mathbf{1}))_i-(R(r^*))_i \\
&\leq (1-2^{-|P|}\delta)-(r^*)_i \\
&\leq (1-2^{-|P}\delta)(1-q_\delta^*)_i
\end{aligned}$$

So condition (I) of Lemma 3.27, with $\beta = 2^{-(|P|+1)}\delta$, applies to variables y_i which either have type $\mathtt{Q}$ or have $R_i(\mathbf{1}) < 1$.

It remains to find an α such that condition (II) of Lemma 3.27 that applies to y_i which either has type $\mathtt{L}$ and satisfies $R(\mathbf{1})_i = 1$. (Note that there aren't any variables of type $\mathtt{T}$ in M_δ, and thus none in y.) We need Lemma 3.31 from chapter 3.

So given y_i of type $\mathtt{L}$ and with $R_i(\mathbf{1}) = 1$, there is a sequence $y_{l_l}, y_{l_2}, \ldots, y_{l_m}$ with $l_m = i$, with y_{l_1} of type $\mathtt{Q}$ or $R(\mathbf{1})_{l_m} < 1$ and for every $0 \leq k < m$, $R(y)_{l_{k+1}}$ contains a term with y_{l_k}. Without loss of generality, we consider the shortest such sequence. Then for $0 < k \leq m$, y_{l_k} does not have type $\mathtt{Q}$ so it must have type $\mathtt{L}$. Also $R(\mathbf{1})_{l_k} = 1$. So $R(y)_{l_k}$ contains a term $p_{l_k,l_{k-1}} y_{k-1}$. We have that, $B_R(r^*)_{l_k,l_{k-1}} = p_{l_k,l_{k-1}}$. Because $R(\mathbf{1})_{l_k} = 1$, this term has not been reduced in P_δ, so $p_{l_k,l_{k-1}}$ is a coefficient in $x = P(x)$. That this is the shortest sequence implies that each of these is a distinct coefficient in $x = P(x)$.

So $\prod_{k=1}^{m-1} p_{l_{k+1},l_k} \geq 2^{-|P|}$. Now $(B_R(r^*)^{m-1})_{i,l_m} \geq \prod_{k=1}^{m-1} B_R(r^*)_{l_{k+1},l_k} = \prod_{k=1}^{m-1} p_{l_{k+1},l_k} \geq 2^{-|P|}$.

So condition (II) of Lemma 3.27 applies to y_i of type L with $R_i(\mathbf{1}) = 1$ when $\alpha = 2^{-|P|}$.

We can now use Lemma 3.27 with $A = B_R(r^*)$, $u = \mathbf{1} - r^*$, $\alpha = 2^{-|P|}$ and $\beta = 2^{-|P|}\delta$, giving

$$\|(I - B_R(r^*))^{-1}\|_\infty \leq \frac{n}{(\mathbf{1}-r^*)^2_{\min} 2^{-|P|} 2^{-|P|}\delta}$$

We have argued that $(\mathbf{1} - r^*)_{\min} \geq 2^{-|P|}\delta$. Using $n \leq 2^{|P|}$ as a (very) conservative bound on n, we have:

$$\|(I - B_\delta(q^*_\delta)_{M_\delta})^{-1}\|_\infty \leq 2^{5|P|}\delta^{-3} \tag{6.3}$$

If A_δ is empty, then $B_\delta(q^*_\delta) = B_\delta(q^*_\delta)_{M_\delta}$ and so we are done.

Otherwise we appeal to Lemma 6.13 with the block decomposition $I - B_\delta(q^*_\delta) = \begin{pmatrix} I - B_\delta(q^*_\delta)_{M_\delta} & -B_\delta(q^*_\delta)_{M_\delta,A_\delta} \\ 0 & I - B_\delta(q^*_\delta)_{A_\delta} \end{pmatrix}$. Letting $\mathcal{Z} = (I - B_\delta(q^*_\delta)_{M_\delta})$, applying Lemma 6.13, we get:

$$\begin{aligned} \|(I - B_\delta(q^*_\delta))^{-1}\|_\infty \quad \leq \quad & \max\{\|\mathcal{Z}^{-1}\|_\infty + \|\mathcal{Z}^{-1}\|_\infty \|B_\delta(q^*_\delta)_{M_\delta,A_\delta}\|_\infty \|(I - B_\delta(q^*_\delta)_{A_\delta})^{-1}\|_\infty, \\ & \|(I - B_\delta(q^*_\delta)_{A_\delta})^{-1}\|_\infty\} \end{aligned}$$

and $\|(I - B_\delta(q^*_\delta)_{A_\delta})^{-1}\|_\infty \leq 2^{3|P|}$ and $\|B_\delta(q^*_\delta)_{M_\delta,A_\delta}\|_\infty \leq 2$. Combining with the bound above in (6.3), we get:

$$\|(I - B_\delta(q^*_\delta))^{-1}\|_\infty \leq \max\{2^{5|P|}\delta^{-3} + 2^{5|P|}\delta^{-3} 2^{3|P|} 2, 2^{3|P|}\}$$

Or, more simply, $\|(I - B_\delta(q^*_\delta))^{-1}\|_\infty \leq 2^{8|P|+2}\delta^{-3}$. □

We are finally ready to prove Theorem 6.16, to which this entire section was dedicated.

Theorem 6.16. *For any $\varepsilon > 0$, and for any SCFG, G, in SNF form, with $q^G > 0$, with critical depth $\mathfrak{c}(G)$, consider the new SCFG, G', obtained from G by the following process: for each bottom-critical SCC, $\mathcal{S}$, of $x = P_G(x)$, find any rule $r = A \xrightarrow{p} B$ of G, such that A and B are both in $\mathcal{S}$ (since G is in SNF, such a rule must exist in every critical SCC). Reduce the probability p, by setting it to $p' = p(1 - 2^{-(14|G|+3)2^{\mathfrak{c}(G)}}\varepsilon^{2^{\mathfrak{c}(G)}})$. Do this for all bottom-critical SCCs. This defines G', which is non-critical.*

Using G' instead of G, if we apply R-NM, with parameter $h+2$ to approximate the LFP solution $q^{G'\otimes D}$ of the MPS $y = P_{G'\otimes D}(y)$, then $\|q^{G\otimes D} - x^{[h+1]}\|_\infty \leq \varepsilon$ where $h := \lceil \log d + (3\cdot 2^{\mathfrak{c}(G)}+1)(\log(1/\varepsilon)+14|G|+3)\rceil$.

Thus we can compute the probability $q_A^{G,D} = \sum_{t\in F} q_{s_0At}^{G\otimes D}$ within additive error $\delta > 0$ in time polynomial in: $|G|$, $|D|$, $\log(1/\delta)$, and $2^{\mathfrak{c}(G)}$, in the standard Turing model of computation.

Proof of Theorem 6.16. Note that for an SCFG, G, and its corresponding PPS, $x = P_G(x)$, the bit encoding size of G is at least as big as that of the PPS. In other words, we have $|G| \geq |P_G|$. So, we can apply Theorem 6.20 to the PPS $x = P_G(x)$ with $\delta := 2^{-(14|G|+3)2^{\mathfrak{c}(G)}}\varepsilon^{2^{\mathfrak{c}(G)}}$, yielding that $\|q^G - q^{G'}\|_\infty \leq \frac{\varepsilon}{2}$ and $\|(I - B_{G'}(q^{G'}))^{-1}\|_\infty \leq 2^{8|G|+2+3(14|G|+3)2^{\mathfrak{c}G}}\varepsilon^{-3\cdot 2^{\mathfrak{c}(G)}}$. Now Lemma 6.4 and Lemma 6.3 (vi) allow us to convert this bound on $\|(I - B_{G'}(q^{G'}))^{-1}\|_\infty$ to a bound on $\|(I - B_{G'\otimes D}(q^{G'\otimes D}))^{-1}\|_\infty$. Namely:

$$\|(I - B_{G'\otimes D}(q^{G'\otimes D}))^{-1}\|_\infty \leq d2^{8|G|+2+3(14|G|+3)2^{\mathfrak{c}(G)}}\varepsilon^{-3\cdot 2^{\mathfrak{c}(G)}}$$

Now Theorem 6.9 gives that $\|q^*_{G'\otimes D} - x^{[h+1]}\|_\infty \leq \frac{\varepsilon}{2}$ since
$h \geq \log\|(I - B_{G'\otimes D}(q^*_{G'\otimes D}))^{-1}\|_\infty + \log(1/\frac{\varepsilon}{2})$. Thus

$$\begin{aligned}
\|q^{G\otimes D} - x^{[h+1]}\|_\infty &\leq \|q^{G\otimes D} - q^{G'\otimes D}\|_\infty + \|q^{G'\otimes D} - x^{[h+1]}\|_\infty \\
&\leq \|q^G - q^{G'}\|_\infty + \|q^{G'\otimes D} - x^{[h+1]}\|_\infty \quad \text{(by Lemma 6.4 \& Lemma 6.3}(vi)) \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\
&= \varepsilon\,.
\end{aligned}$$

□

6.4 Non-criticality of SCFGs obtained by EM

In doing parameter estimation for SCFGs, in either the supervised or unsupervised (EM) settings (see, e.g., [NS08]), we are given a CFG, $\mathcal{H}$, with start nonterminal S, and we wish to extend it to an SCFG, G, by giving probabilities to the rules of $\mathcal{H}$. We also have some probability distribution, $\mathcal{P}(\pi)$, over the complete derivations, π, of $\mathcal{H}$ that start at start non-terminal S. (In the unsupervised case, we begin with an SCFG, and the distribution $\mathcal{P}$ arises from the prior rule probabilities, and from the training corpus of strings.) We then assign each rule of $\mathcal{H}$ a (new) probability as follows to obtain (or update) G:

$$p(A \to \gamma) := \frac{\sum_\pi \mathcal{P}(\pi) C(A \to \gamma, \pi)}{\sum_\pi \mathcal{P}(\pi) C(A, \pi)} \tag{6.4}$$

where $C(r, \pi)$ is the number of times the rule r is used in the complete derivation π, and $C(A, \pi) = \sum_{r \in R_A} C(r, \pi)$. Equation (6.4) only makes sense when the sums $\sum_\pi \mathcal{P}(\pi) C(A, \pi)$ are finite and nonzero, which we assume; we also assume every nonterminal and rule of $\mathcal{H}$ appears in some complete derivation π with $\mathcal{P}(\pi) > 0$.

Proposition 6.21. *If we use parameter estimation to obtain SCFG G using equation (6.4), under the stated assumptions, then G is consistent*[1]*, i.e. $q^G = \mathbf{1}$, and* furthermore *the PPS $x = P_G(x)$ is non-critical, i.e., $\rho(B_G(\mathbf{1})) < 1$.*

It follows from Prop. 6.21 and Thm. 6.7, that for SCFGs obtained by parameter estimation and EM, we can compute the probability $q_A^{G,D}$ of generating a string in $L(D)$ to within any desired precision in P-time, for any DFA D.

A first step toward establishing Proposition 6.21 is the following Lemma, from which we derive a (left) *cone vector* for $B_G(\mathbf{1})$, which ultimately allows us to show $\rho(B_G(\mathbf{1})) < 1$.

Recall that, for a string $\alpha \in (V \cup \Sigma)^*$, with $n = |V|$, $\kappa(\alpha)$ is the n-vector where, for $A \in V$, $\kappa_A(\alpha)$ is the number of times A appears in α. For $A \in V$, define $\mathbf{e}^A$ to be the unit n-vector with $(\mathbf{e}^A)_A = 1$ and $(\mathbf{e}^A)_B =$ for $B \neq A$. Define $K(\pi) = \sum_A C(A, \pi) \mathbf{e}^A$.

Lemma 6.22. *Let S denote the designated start nonterminal. Then*

$$\mathbf{e}^S = (I - B_G(\mathbf{1})^T)(\sum_\pi \mathcal{P}(\pi) K(\pi))$$

Proof. Firstly, we need to relate $B_G(\mathbf{1})$ to the probabilities of the rules. Given a rule $A \to \gamma$ we define $B_{A \to \gamma}(x) := B_{G_{A \to \gamma}}(x)$ where $G_{A \to \gamma}$ is an SCFG with the same nonterminals and terminals as G but with only one rule, $A \xrightarrow{1} \gamma$, which has probability 1. So then $B_{A \to \gamma}(\mathbf{1})$ is zero outside the A row. We allow that G may or may not be in normal form. We can say that

$$P_G(x)_A = \sum_{r = (A \to \gamma) \in R_A} p(r) \prod_{B \in V} x_B^{\kappa_B(\gamma)}$$

In terms of the "partial" SCFGs, G_r, associated with each rule $r \in R$, this says $P_G(x)_A = \sum_{r \in R_A} p(r) P_{G_r}(x)_A$. The A row of $B_G(x)$ is then $\sum_{r \in R_A} p(r) B_r(x)_A$. Since $B_{A \to \gamma}(x_G)$ is

[1] Consistency of the obtained SCFGs is well-known; see, e.g., [NS06, NS08] & references therein; also [SB97] has results related to Prop. 6.21 for restricted grammars.

zero outside of the A row, $B_G(x) = \sum_A \sum_{r\in R_A} p(r)B_r(x)$. That is:

$$B_G(x) = \sum_{r\in R} p(r)B_r(x) \tag{6.5}$$

So we can obtain $B_G(\mathbf{1})$ from each of the $B_r(\mathbf{1})$. $B_{A\to\gamma}(\mathbf{1})$ is zero except in the A row. For any non-terminal B, $B_{A\to\gamma}(x)_{A,B} = \frac{\partial}{\partial x_B}\prod_C x_C^{\kappa(\gamma)_C} = \kappa_B(\gamma)x_B^{\kappa_B(\gamma)-1}\prod_{C\neq B} x_C^{\kappa(\gamma)_C}$. Evaluated at $\mathbf{1}$, this yields:

$$(B_{A\to\gamma}(\mathbf{1}))_{A,B} = \kappa_B(\gamma) \tag{6.6}$$

Now we look at what happens to the count of non-terminals in the derivation π. We have $S \overset{\pi}{\Rightarrow} w$ for some $w \in \Sigma^*$. That is, $\pi = r_1r_2\dots r_k \in R^*$, and $\alpha_0 \overset{r_1}{\Rightarrow} \alpha_1 \overset{r_2}{\Rightarrow} \alpha_2 \overset{r_2}{\Rightarrow} \dots \overset{r_k}{\Rightarrow} \alpha_m$, for $\alpha_0 = S$, $\alpha_m = w$ and some $\alpha_1, \alpha_2, \dots, \alpha_{m-1} \in (V \cup \Sigma)^*$.

Consider $\alpha_i \overset{r_i}{\Rightarrow} \alpha_{i+1}$ for some $0 \le i \le m-1$. The rule r_i is $A_i \to \gamma_i$ for some non-terminal A_i and some string γ_i. Replacing A_i by γ_i affects the counts of the non-terminals by $\kappa(\alpha_{i+1}) - \kappa(\alpha_i) = \kappa(\gamma_i) - \mathbf{e}^{A_i}$. Note that for any nonterminal A, and rule $A \to \gamma$, we have $B_{A\to\gamma}(\mathbf{1})^T\mathbf{e}^A = \kappa(\gamma)$, by equation (6.6), so

$$(I - B_{A\to\gamma}(\mathbf{1})^T)\mathbf{e}^A = \mathbf{e}^A - \kappa(\gamma) \tag{6.7}$$

Since for any string $w \in \Sigma^*$, we have $\kappa(w) = \mathbf{0}$, we get:

$$\begin{aligned}
\mathbf{e}^S &= \mathbf{e}^S - \kappa(w) \\
&= \sum_{i=0}^{m-1} \kappa(\alpha_i) - \kappa(\alpha_{i+1}) \\
&= \sum_A \sum_{(A\to\gamma)\in R_A} (C(A\to\gamma,\pi))(\mathbf{e}^A - \kappa(\gamma)) \\
&= \sum_A \sum_{(A\to\gamma)\in R_A} (C(A\to\gamma,\pi))(I - B_{A\to\gamma}(\mathbf{1})^T)\mathbf{e}^A \quad \text{(by (6.7))}
\end{aligned}$$

This is true for any complete derivation π, so we can use the probability distribution $\mathcal{P}(\pi)$, which has $\sum_\pi \mathcal{P}(\pi) = 1$ to obtain:

$$\begin{aligned}
\mathbf{e}^S &= \sum_\pi \mathcal{P}(\pi) \sum_A \sum_{(A\to\gamma)\in R_A} (C(A\to\gamma,\pi))(I - B_{A\to\gamma}(\mathbf{1})^T)\mathbf{e}^A \\
&= \sum_{A\in V} (\sum_{(A\to\gamma)\in R_A} \sum_\pi \mathcal{P}(\pi)(C(A\to\gamma,\pi))(I - B_{A\to\gamma}(\mathbf{1})^T))\mathbf{e}^A \\
&= \sum_A (I - B_G(\mathbf{1})^T)(\sum_\pi \mathcal{P}(\pi)C(A,\pi))\mathbf{e}^A \\
&= (I - B_G(\mathbf{1})^T)(\sum_\pi \mathcal{P}(\pi)K(\pi))\ .
\end{aligned}$$

□

Proof of Theorem 6.21. Define $v = (\sum_\pi \mathcal{P}(\pi)K(\pi))$. Then we have that $v = B_G(\mathbf{1})^T v + \mathbf{e}^S$. We want to use Lemma 3.27 to show that $\rho(B_G(\mathbf{1})^T) < 1$. We can do this by applying it to the vector $u = \frac{1}{\|v\|_\infty} v$. We do not need explicit bounds on α, β and $u_{\min}$, but we need to show that the conditions hold for some positive α, β and $u_{\min}$. Firstly, we note that $v > 0$, since every non-terminal in G appears in some derivation π with $\mathcal{P}(\pi) > 0$. So $u > 0$. Since $u = \frac{1}{\|v\|_\infty} v$, $\|u\|_\infty = 1$. Note that $u = \frac{1}{\|v\|_\infty}(B_G(\mathbf{1})^T v + \mathbf{e}^S) = B_G(\mathbf{1})^T u + \frac{1}{\|v\|_\infty}\mathbf{e}^S$. Thus $B_G(\mathbf{1})^T u = u - \frac{1}{\|v\|_\infty}\mathbf{e}^S \leq u$. In the S coordinate (and only in the S coordinate), we have that $(B_G(\mathbf{1})^T u)_S = u_S - \frac{1}{\|v\|_\infty} < u_S$, so there is some $\beta > 0$ for which $(B_G(\mathbf{1})^T u)_S \leq (1-\beta)u_S$. For this β, u_S satisfies condition (I) of Lemma 3.27. We need to find an α for which all non-terminals other than S satisfy condition (II) of Lemma 3.27.

Consider a non-terminal $A \neq S$. A appears in some complete derivation π with $\mathcal{P}(\pi) > 0$. There is some sequence of (not necessarily consecutive) rules $r_i : D_i \to \gamma_i$, $i = 1, \ldots, k$, appearing in that order in π, such that $D_1 = S$, $D_i \in \gamma_{i-1}$ for all $2 \leq i \leq k$, and $A \in \gamma_k$. Without loss of generality $k \leq n$, since otherwise there must be i, j with $2 \leq i < j \leq k$ such that $D_i = D_j$ and so the shorter sequence $r_1, \ldots, r_{i-1}, r_j, \ldots, r_k$ would have satisfied the above conditions.

For any $1 \leq i \leq k-1$, $(B_{r_i}(\mathbf{1}))_{D_i, D_{i+1}} = \kappa(\gamma_i)_{D_{i+1}} \geq 1$, and similarly $(B_{r_k}(\mathbf{1}))_{D_k, A} \geq 1$. Now any r_j, with $1 \leq j \leq k$, appears in π which has $\mathcal{P}(\pi) > 0$. So $p(r_j) > 0$. But $B_G(\mathbf{1}) \geq p(r_j)B_{r_j}(\mathbf{1})$. So for any $1 \leq i \leq k-1$, $(B_G(\mathbf{1}))_{D_i, D_{i+1}} \geq p(r_i) > 0$ and similarly $B_G(\mathbf{1})_{D_k, A} > 0$. So $(B_G(\mathbf{1})^k)_{S,A} > 0$. Then $((B_G(\mathbf{1})^T)^k)_{A,S} = ((B_G(\mathbf{1})^k)^T)_{A,S} = (B_G(\mathbf{1})^k)_{S,A} > 0$. We then define $\alpha_A = ((B_G(\mathbf{1})^T)^k)_{A,S}$. If we take $\alpha = \min_{\{A \in V | A \neq S\}} \alpha_A$, then $\alpha > 0$ and all non-terminals $A \neq S$ satisfy condition (II) of Lemma 3.27: i.e., for each $A \neq S$, there is a k with $((B_G(\mathbf{1})^T)^k)_{A,S} \geq \alpha$. We can now apply Lemma 3.27 which yields that $\rho(B_G(\mathbf{1})^T) < 1$. So $\rho(B_G(\mathbf{1})) = \rho(B_G(\mathbf{1})^T) < 1$. So, G is not critical. Consistency of G, i.e., the fact that $q^G = \mathbf{1}$, also follows. This holds because, firstly, we can easily see that G is a *proper* SCFG. In other words, for any nonterminal A, the sum of the rule probabilities is 1, because $\sum_{r \in R_A} p(r) = \sum_{r \in R_A} \frac{\sum_\pi \mathcal{P}(\pi)C(r,\pi)}{\sum_\pi \mathcal{P}(\pi)C(A,\pi)} = 1$.

Thus, G has a PPS, $x = P_G(x)$, such that $P_G(\mathbf{1}) = \mathbf{1}$, and $\rho(B_G(\mathbf{1})) < 1$. Lemma 6.3 of [EY09] tells us that for any vectors $0 \leq x \leq y$, $B_G(y)(y-x) \geq P_G(y) - P_G(x)$. Let $y = \mathbf{1}$, and let $x = q^G$. Then we have $B_G(\mathbf{1})(\mathbf{1} - q^G) \geq P_G(\mathbf{1}) - P_G(q^G) = \mathbf{1} - q^G$, since we have argued both $\mathbf{1}$ and q^G are fixed points of P_G. But $B_G(\mathbf{1})$ is a non-negative square matrix, and $(\mathbf{1} - q^G) \geq 0$. Theorem 8.3.2 of [HJ85] tells us that for a square matrix $M \geq 0$, and vector $v \geq 0$, if $v \neq 0$ and $Mv \geq v$, then $\rho(M) \geq 1$. We know that

$B_G(\mathbf{1})(\mathbf{1}-q^G) \geq \mathbf{1}-q^G$, but we have already established that $\rho(B_G(\mathbf{1})) < 1$. Thus it must be the case that $(\mathbf{1}-q^G) = 0$. In other words, G is consistent. □

6.5 A bad example for infix probabilities

We now present a family of SCFGs, G_n, of size $O(n)$, and with critical-depth n, and we give a fixed 3-state DFA, D. We use these to indicate why it is likely to be difficult to overcome the exponential dependence on critical-depth of the given SCFG, G, in order to obtain a P-time algorithms for computing the probability (within desired precision) that an arbitrary G generates a string in $L(D)$.

The DFA D, is depicted in Figure 1. It has only 3 states and the property it checks is whether aa is an "infix" of the string. In other words, $L(D) = \{waaw' \mid w \in \Sigma^* \text{ and } w' \in \Sigma^*\}$. The family of SCFGs G_n is defined by the following rules:

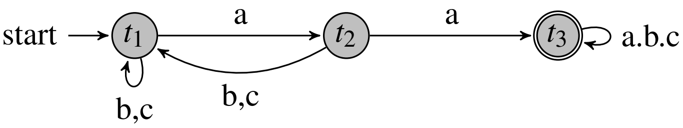

Figure 6.1: Automaton for the infix aa

$A_0 \xrightarrow{0.5} A_0A_0$
$A_0 \xrightarrow{0.5} A_1$
$A_1 \xrightarrow{0.5} A_1A_1$
$A_1 \xrightarrow{0.5} A_2$
$\dots$
$A_n \xrightarrow{1} caB_nac$

$B_n \xrightarrow{1} B_{n-1}B_{n-1}$
$B_{n-1} \xrightarrow{1} B_{n-2}B_{n-2}$
$\dots$
$B_0 \xrightarrow{0.5} \varepsilon$
$B_0 \xrightarrow{0.5} b$

Proposition 6.23. $q^{G_n} = \mathbf{1}$. *In other words, the probability of termination (generating a finite string) starting at any nonterminal in G_n is* 1.

Furthermore, $q^{G_n \otimes D}_{(t_1 A_0 t_3)} = \frac{1}{2}$ is the probability that this SCFG G_n, starting at A_0, generates a string which has infix aa. On the other hand, $q^{G_n \otimes D}_{(t_1 A_i t_3)} = 2^{-2^i}$ is the same probability, starting at A_i.

The proof of this proposition is not at all difficult (using simple induction, and the formula for solving quadratic equations).

Let us argue why this causes severe difficulties for the approximate computation of $q^{G \otimes D}$. Note that $q^{G_n \otimes D}_{(t_1 A_0 t_3)} = \frac{1}{2}$ and $q^{G_n \otimes D}_{(t_1 A_n t_3)} = 2^{-2^n}$. However, in the product MPS $y = P_{G \otimes D}(y)$ the variable $y_{(t_1 A_0 t_3)}$ depends on the variable $y_{(t_1 A_n t_3)}$, and furthermore, if we, for example, "under-approximate" $q^{G_n \otimes D}_{(t_1 A_n t_3)} = 2^{-2^n}$, and instead set $y_{(t_1 A_n t_3)} := 0$, or, what effectively achieves the same result, if we change the product MPS by setting $P_{G \otimes D}(y)_{t_1 A_n t_3} \equiv 0$, then in the resulting modified MPS, with new LFP $\tilde{q}^{G_n \otimes D}$, we would get $\tilde{q}^{G_n \otimes D}_{(t_1 A_0 t_3)} = 0$.

Likewise, one can show that if we "over-approximate" $q^{G_n \otimes D}_{(t_1 A_n t_3)}$, even very slightly, setting $P_{G \otimes D}(y)_{t_1 A_n t_3} \equiv \frac{1}{2^{poly}}$ in a consistent way, then we will end up with a new LFP $\tilde{q}^{G_n \otimes D}$, such that $\tilde{q}^{G_n \otimes D}_{(t_1 A_0 t_3)} \approx 1$ (in other words, very close to 1).

In both cases, the resulting approximate solution $\tilde{q}^{G_n \otimes D}_{(t_1 A_0 t_3)}$ is terribly far from the actual solution $\frac{1}{2}$. (Note that this is irrespective of the algorithm that is used to compute the other probabilities.)

Furthermore, we can not in any way use the fact that we can detect in P-time and remove variables x_A from the PPS $x = P_{G_n}(x)$ for which $q^{G_n}_A = 1$, because indeed $q^G = \mathbf{1}$, and yet in the product $q^{G \otimes D}$ there are coordinates with wildly different probabilities that we wish to compute.

Chapter 7

Conclusions and further work

We have given algorithms for approximating termination or extinction probabilities for SCFGs, MT-BPs, BMDPs and p1CAs to within error $\varepsilon > 0$, in time polynomial in the encoding size of the model and $\log 1/\varepsilon$. To do so, we used Newton's method on PPSs and MPSs, and in the case of BMDPs and their associated max/min-PPSs, we used a generalisation of Newton's method.

Most of the key open questions about the complexity of computing termination probabilities for these models have now been resolved. Many interesting future directions of research regarding these models relate to computing other quantities of interest. In general, our results should be extendable to give similar (hopefully, polynomial time) algorithms for other quantities (probabilities, expectations of relevant random variables, etc.) whose computation has been studied in the literature.

Our detailed analysis of Newton's method for PPSs and MPSs represents one of the rare cases that we are aware of where essentially optimal *worst case* complexity bounds have been obtained for the behaviour of Newton's method for solving a broad class of nonlinear systems of equations, as a function of both the encoding size of the system of equations, and the desired error of the approximation. Newton's method is one of the main workhorses of numerical methods for solving nonlinear systems of equations, and nonlinear optimization problems. Can we apply some of the insights we have gained in our analysis of Newton's method for MPSs, to the context of other natural classes of nonlinear systems?

In the short term, there are a few obvious open problems that suggest themselves:

1. We have shown that, given a stochastic context-free grammar and a DFA representing a regular language, we can approximate the probability that the grammar produces a word in the regular language, and we can do so in polynomial time

as long as the stochastic grammar has bounded *critical depth* (which holds for any SCFG obtained via EM iteration).

One might ask whether the exponential dependence on critical depth could be improved. We have no complexity-theoretic hardness result at the moment, showing that it is unlikely that we can obtain a polynomial time algorithm even in the cases of high critical depth. This contrasts with the case of approximating the LFP of a general MPS or the decision problem (as opposed to approximation) for the LFP of a PPS, where PosSLP-hardness results [EY09] mean that we are unlikely to get polynomial time in the general case. So, it remains open whether we can find a polynomial time algorithm, or establish a hardness result, for this regular language problem.

2. There are some open problems remaining, related to the problem of *model checking* 1-exit Recursive Markov Chains/SCFGs against ω-regular or LTL specifications, which was studied in [EKM06, EY12].

   From the results of [EY12], it follows that the relevant model checking quantities that we wish to compute constitute the solution to a linear system of equations *whose coefficients are expressions that involve quantities which amount to the probability with which a given SCFG generates a string in a given regular language*. These linear systems of equations arise as the equations for the hitting probabilities in a certain *conditioned summary chain* which is a finite state Markov chain derivable from the model.

   Thus, one may hope that using our algorithm for computing *approximately* the probability that a SCFG generates a string in a given regular language, we can calculate the coefficients of the relevant linear system of equations to sufficient accuracy, in order to be able to approximate the model checking quantities by using the solution to such approximate linear systems of equations.

   However, this raises the question of how accurately we need to approximate those coefficients (and in particular, how *well conditioned* those linear systems of equations are).

   It would be useful to establish general syntactic conditions under which suitable well-conditioning and non-criticality conditions hold, such that these model checking probabilities can be approximated in polynomial time in size of the model and $\log(1/\varepsilon)$.

3. There also remain interesting model checking questions that remain open, relating to multi-type branching processes.

   In recent work [CDK12], Chen et. al. have considered some model checking questions for labelled multi-type branching processes with respect to certain branching-time properties (specified by deterministic parity tree automata or by a certain tree extension of PCTL).

   They developed a PSPACE procedure for deciding whether the relevant model checking probabilities are at least a give value. The main ingredient of their procedure involves computation of extinction probabilities for branching processes, and the related decision problems. Since we have shown in this thesis that the decision problem for extinction probabilities is in unit-cost-P-time, it would be interesting to show that the same holds for these model checking quantities. More importantly, a key question is whether we can *approximate* key model checking quantities for MT-BPs in polynomial time (in the Turing model of computation).

   Another area where our results on efficient analysis of MT-BPs could potentially have applications is in quantitative analysis of certain cancer models based on MT-BPs that have been studied in the literature (see [RBCN13, BRA$^+$13] and the references therein).

4. We can also consider model checking Branching Markov Decision Processes (BMDPs).

   For the model checking properties we considered above about MT-BPs, we can consider the optimal probability of a BMDP satisfying such a property. However, for a number of important objectives of interest, it is easy to see that *static* optimal policies (i.e., memoryless, and context-independent optimal strategies) do not exist in general. This makes the relevant computational problems more challenging, in particular because it is not obvious how one can reduce these problems to computation of a fixed point for a system of nonlinear equations. But for some useful objectives, such as reachability, we can establish that there are static optimal policies, and we can cast the problem as a solution of a set of fixed-point equations. This is part of work in progress which we hope to complete soon.

5. Another important family of computational questions arises in the context of statistical Natural Language Processing (NLP).

In particular, one question is, can we give a complexity analysis of standard algorithms for learning SCFGs, such as the EM algorithm (aka the inside-outside algorithm)? It is possible to use our algorithms to prove a polynomial running time result for each iteration of the EM algorithm, and furthermore to show that even when iterations of EM are approximated the results will indeed monotonically improve (or at least not decrease by much) the likelihood function. This is part of work in progress. But it is not clear at all what other natural extra assumptions and results are required to establish efficient convergence of the EM algorithm globally. (There are complexity theoretic hardness results in the literature which imply that without strong assumptions it is not possible to establish efficient convergence.)

Further interesting questions arise when we consider more general models used in NLP, that go beyond SCFGs, for example language models that incorporate some context information. Can our results be applied to such extensions to the SCFG model that are used in NLP? In some cases, the answer is easily seen to be "yes", because the relevant problems constitute a straightforward extension algorithms we have developed for termination probability. But for other problems it is less clear.

These are just some of the many interesting future directions of research that could be pursued, building on the results in this thesis.

Bibliography

[ABE^{+}05] R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T. Reps, and M. Yannakakis. Analysis of recursive state machines. *ACM Trans. Program. Lang. Syst.*, 27(4), 2005.

[ABKPM09] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM J. Comput.*, 38(5):1987–2006, 2009.

[Apo74] T. Apostol. *Mathematical Analysis*. Addison-Wesley, 2nd edition, 1974.

[BBEK11] T. Brázdil, V. Brozek, K. Etessami, and A. Kucera. Approximating the termination value of one-counter mdps and stochastic games. *Proc. of 38th ICALP (2)*, pages 332–343, 2011.

[BBFK06] TomáŽ Brázdil, Václav Brođek, Vojtech Forejt, and Antonín Kucera. Reachability in recursive markov decision processes. *Proc. 17th Int. CONCUR,*, page 358Ű374, 2006.

[BKK11] T. Brázdil, S. Kiefer, and A. Kucera. Efficient analysis of probabilistic programs with an unbounded counter. *Proc. of 23rd Int. Conf. on Computer Aided Verification(CAV)*, pages 208–224, 2011.

[BLM05] D. Bini, G. Latouche, and B. Meini. *Numerical methods for Structured Markov Chains*. Oxford University Press, 2005.

[BRA^{+}13] I. Bozic, J. G. Reiter, B. Allen, T. Antal, K. Chatterjee, P. Shah, Y. S. Moon, A. Yaqubie, N. Kelly, D. T. Le, E. J. Lipson, P. B. Chapman, Jr L. A. Diaz, B. Vogelstein, and M. A Nowak. Evolutionary dynamics of cancer in response to targeted combination therapy. *eLife*, 2, 2013.

[Cam94] P. Cameron. *Combinatorics: topics, techniques, algorithms*. Cambridge U. Press, 1994.

[CDK12] T. Chen, K. Dräger, and S. Kiefer. Model checking stochastic branching processes. In *MFCS*, pages 271–282, 2012.

[CMGS91] A. Corazza, R. De Mori, D. Gretter, and G. Satta. Computation of probabilities for an island-driven parser. *IEEE Trans. PAMI*, 13(9):936–950, 1991.

[Con92] Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[CY98] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. *IEEE Trans. on Automatic Control*, 43(10):1399–1418, 1998.

[DEKM99] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of Proteins and Nucleic Acids*. Cambridge U. Press, 1999.

[DR05] E. Denardo and U. Rothblum. Totally expanding multiplicative systems. *Linear Algebra Appl.*, 406:142–158, 2005.

[EGK13] J. Esparza, A. Gaiser, and S. Kiefer. A strongly polynomial algorithm for criticality of branching processes and consistency of stochastic context-free grammars. *Inf. Process. Lett.*, 113(10-11):381–385, 2013.

[EGKS08] J. Esparza, T. Gawlitza, S. Kiefer, and H. Seidl. Approximative methods for monotone systems of min-max-polynomial equations. *Proc. of 35th ICALP (1)*, pages 698–710, 2008.

[EKL10] Javier Esparza, Stefan Kiefer, and Michael Luttenberger. Computing the least fixed point of positive polynomial systems. *Siam Journal on Computing*, 39(6):2282–2355, 2010.

[EKM06] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. *Logical Methods in Computer Science*, 2(1):1 – 31, 2006.

[ESY12a] Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. Polynomial time algorithms for branching markov decision processes and probabilistic min(max) polynomial Bellman equations. *ICALP*, 2012. see full Arxiv version, http://arxiv.org/abs/1202.4798.

[ESY12b] Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars. *STOC*, 2012. see full Arxiv version, http://arxiv.org/abs/1201.2374.

[ESY13a] K. Etessami, A. Stewart, and M. Yannakakis. Stochastic context-free grammars, regular languages, and newton's method. In *Proc. 40th Int. Coll. on Automata, Languages, and Programming (ICALP'13)*, pages 199–211, 2013. see full Arxiv version, http://arxiv.org/abs/1302.6411.

[ESY13b] Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. Upper bounds for Newton's method on monotone polynomial systems, and P-time model checking of probabilistic one-counter automata. *CAV*, 2013.

[EWY08] K Etessami, D Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. *Proc. of 35th ICALP (1)*, 2008. see full tech report at `http://homepages.inf.ed.ac.uk/kousha/bib_index.html` .

[EWY10] Kousha Etessami, Dominik Wojtczak, and Mihalis Yannakakis. Quasi-birth-death processes, tree-like qbds, probabilistic 1-counter automata, and pushdown systems. *Performance Evaluation*, 67(9), 2010.

[EY05] Kousha Etessami and Mihalis Yannakakis. Recursive markov decision processes and recursive stochastic games. *32nd Int. Coll. on Automata, Languages and Programming (ICALP'05)*, 2005.

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

[EY10] Kousha Etessami and Mihalis Yannakakis. On the complexity of nash equilibria and other fixed points. *Siam Journal on Computing*, 2010.

[EY12] Kousha Etessami and Mihalis Yannakakis. Model checking of recursive probabilistic systems. *ACM Transactions on Computational Logic*, 13(2), 2012.

[FKK$^+$00] R. Fagin, A. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons". *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 484–493, 2000.

[Har63] T. E. Harris. *The Theory of Branching Processes.* Springer-Verlag), 1963.

[HJ85] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[HJV05] P. Haccou, P. Jagers, and V. A. Vatutin. *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge U. Press, 2005.

[HKL$^+$11] K. Arnsfelt Hansen, M. Koucký, N. Lauritzen, P. Bro Miltersen, and E. P. Tsigaridas. Exact algorithms for solving stochastic games: extended abstract. *STOC*, 2011. see full Arxiv version, http://arxiv.org/abs/1202.3898.

[IK66] E. Isaacson and H. B. Keller. *Analysis of Numerical Methods*. J. Wiley & Sons, 1966.

[JL91] F. Jelinek and J. D. Lafferty. Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17(3):315–323, 1991.

[KA02] M. Kimmel and D. E. Axelrod. *Branching processes in biology*. Springer, 2002.

[KS47] A. N. Kolmogorov and B. A. Sevastyanov. The calculation of final probabilities for branching random processes. *Doklady*, 56:783–786, 1947. (Russian).

[LR99] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM series on statistics and applied probability, 1999.

[LT85] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. Academic Press, 2nd edition, 1985.

[Neu81] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models:an algorithmic approach*. Johns Hopkins U. Press, 1981.

[NS06] Mark-Jan Nederhof and Giorgio Satta. Estimation of consistent probabilistic context-free grammars. *HLT-NAACL*, 2006.

[NS08] M.-J. Nederhof and G. Satta. Probabilistic parsing. *New Developments in Formal Languages and Applications*, 113:229–258, 2008.

[NS09] Mark-Jan Nederhof and Giorgio Satta. Computing partition functions of pcfgs. *Research on Language and Computation*, 6(2):139–162, 2009.

[NS11] Mark-Jan Nederhof and Giorgio Satta. Computation of infix probabilities for probabilistic context-free grammars. *EMNLP 2011*, pages 1213–1221, July 2011.

[Pli76] S. Pliska. Optimization of multitype branching processes. *Management Science*, 23(2):117–124, 1976.

[PP08] I. Pazsit and L. Pal. *Nuclear Fluctuations: a treatise on the physics of branching processes*. Elsevier science, 2008.

[Put94] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[RBCN13] J. G. Reiter, I. Bozic, K. Chatterjee, and M. A. Nowak. Ttp: Tool for tumor progression. In *CAV*, pages 101–106, 2013.

[RW82] U. Rothblum and P. Whittle. Growth optimality for branching markov decision chains. *Math. Oper. Res.*, 7(4):582–601, 1982.

[SB97] J. Sánchez and J.-M. Benedí. Consistency of stochastic context-free grammars from probabilistic estimation based on growth transformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1052–1055, 1997.

[Sto95] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):167–201, 1995.

[WE07] D. Wojtczak and K. Etessami. Premo: an analyzer for probabilistic recursive models. *Proc. 13th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 66–71, 2007.