The Computational Content of Abstract Algebra and Analysis

Franziskus Wolfgang Josef Wiesnet

iNδAM

Trento, München, Verona 2021

The Computational Content of Abstract Algebra and Analysis

Franziskus Wolfgang Josef Wiesnet

PhD Thesis
prepared at
Fakultät für Mathematik, Informatik und Statistik
Ludwig–Maximilians–Universität München
and
Dipartimento di Matematica
Università degli Studi di Trento
and
Dipartimento di Informatica
Università degli Studi di Verona

submitted by
Franziskus Wolfgang Josef Wiesnet
from Landshut, Deutschland

Munich, May 22, 2021

Supervisors: Prof. Dr. Peter Schuster
Prof. Dr. Helmut Schwichtenberg

Statutory declaration, Eidesstattliche Versicherung

I herewith formally declare that I have written the submitted dissertation independently without any unauthorized assistance.

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig ohne unerlaubte Beihilfe angefertigt ist.

München, 22.05.2021
Place, Date

Franziskus Wiesnet
Signature

Abstract

In this dissertation we discuss several forms of proof interpretation based on examples in algebra and analysis. Our main goal is the construction of algorithms or functions out of proofs – constructive proofs and classical proofs.

At the beginning of this dissertation the constructive handling of Zorn's lemma in proofs of algebra plays a main role. We consider maximal objects and their approximations in various algebraic structures. The approximation is inspired by Gödel's functional interpretation and methods from proof mining. Proof mining is a branch of mathematical logic which analyses classical existence statements to get explicit bounds. We describe a recursive algorithm which constructs a sequence of approximations until they become close enough. This algorithm will be applied to Krull's lemma and its generalisation: the universal Krull-Lindenbaum lemma. In some case studies and applications like the theorem of Gauß-Joyal and Kronecker's theorem we show explicit uses of this algorithm.

By considering Zariski's lemma we present how a manual construction of an algorithm from a constructive proof works. We formulate a constructive proof of Zariski's lemma, and use this proof as inspiration to formulate an algorithm and a computational interpretation of Zariski's lemma. Finally, we prove that the algorithm indeed fulfils the computational interpretation. As an outlook we sketch how this algorithm could be combined with our state-based algorithm from above which constructs approximations of maximal objects, to get an algorithm for Hilbert's Nullstellensatz.

We then take a closer look at an example of an algorithm which is extracted out of a constructive proof by using a computer assistant. In our case we use the proof assistant Minlog. We use coinductively defined predicates to prove that the signed digit representation of real numbers is closed under limits of convergent sequences. From a formal proof in Minlog, the computer generates an algorithm, a soundness theorem and a proof of the soundness theorem corresponding to the convergence theorem out of our formalisation. The convergence theorem is applied to Heron's method to get an algorithm which computes the signed digit representation of the

square root of a non-negative real number out of its signed digit representation. Here we use the programming language Haskell to display the computational content. As second application we show that the signed digit code is closed under multiplication and state the corresponding algorithm.

This dissertation is concluded by an example of proof mining in analysis. We consider one convergence lemma with a classical proof, use techniques of proof mining to get a new lemma with a rate of convergence, and apply the new lemma to various fixed-point theorems for asymptotically weakly contractive maps and their variants. In the last step, beside extracting quantitative information – in our case rates of convergence – from proofs, we also discuss the usage of proof mining to generalise or combine theorems.

Zusammenfassung

In dieser Dissertation diskutieren wir verschiedene Formen der Beweisinterpretation anhand von Beispielen aus der Algebra und Analysis. Unser Ziel ist dabei immer die Konstruktion von Algorithmen oder Funktionen aus Beweisen – sowohl klassischen wie auch konstruktiven Beweisen.

Am Anfang der Dissertation gehen wir auf eine konstruktive Behandlung von Zorns Lemma in Beweisen der Algebra ein. Insbesondere betrachten wir maximale Objekte und deren Approximationen in verschiedenen algebraischen Strukturen. Die Approximation ist dabei inspiriert durch Gödels Funktionalinterpretation und Methoden des Proofminings. Proofmining ist ein Teilgebiet der mathematischen Logik, welches klassische Existenzbeweise analysiert, um explizite Schranken zu extrahieren. Wir geben einen rekursiven Algorithmus an, welcher eine Folge von Approximationen konstruiert bis diese hinreichend genau sind. Dieser Algorithmus wird anschließend auf Krulls Lemma und seine Verallgemeinerung angewandt: dem universellen Krull-Lindenbaum-Lemma. In einigen Fallstudien und Beispielen wie dem Satz von Gauß-Joyal und Kroneckers Lemma zeigen wir ganz explizite Anwendungen des Algorithmus.

Anhand von Zariskis Lemma zeigen wir, wie man einen Algorithmus von Hand ohne Computerhilfe aus einem konstruktiven Beweis extrahieren kann. Hierbei geben wir zunächst einen konstruktiven Beweis von Zariskis Lemma an. Diesen Beweis nutzen wir als Inspiration, um einen Algorithmus und eine rechnerische Interpretation von Zariskis Lemma zu formulieren. Abschließend beweisen wir, dass der Algorithmus tatsächlich die rechnerische Interpretation erfüllt. Als Ausblick skizzieren wir, wie dieser Algorithmus angewandt werden kann, um einen Algorithmus für Hilberts Nullstellensatz zu erhalten. Dabei verwenden wir auch den zuvor entwickelten Algorithmus, welcher Approximationen von maximalen Objekten konstruiert.

Nach dieser manuellen Extraktion eines Algorithmus aus einem konstruktiven Beweis kommen wir zu einem Beispiel automatischer Beweisextraktion mit Hilfe von Computerunterstützung. In unserem Fall verwenden wir den Beweisassisten-

ten Minlog. Unter Verwendung von coinduktiven Prädikaten zeigen wir, dass die Binärdarstellung mit Vorzeichen abgeschlossen unter der Bildung von Grenzwerten konvergenter Folgen ist. Nach der Formalisierung in Minlog erzeugt der Computer einen Algorithmus, eine Korrektheitsaussage und einen Beweis dieser Aussage für unser Konvergenztheorem. Dieses Konvergenztheorem wird anschließend auf das Heron-Verfahren angewandt, um einen Algorithmus zu erhalten, welcher die Binärdarstellung mit Vorzeichen einer nicht-negativen reeller Zahl zu der Binärdarstellung mit Vorzeichen ihrer Wurzel transformiert. Um diesen Algorithmus anzuwenden, verwenden wir die Programmiersprache Haskell. Als zweite Anwendung zeigen wir, dass die Binärdarstellung mit Vorzeichen abgeschlossen unter Multiplikation ist, und geben dazu einen Algorithmus an.

Im letzten Kapitel diskutieren wir ein Beispiel von Proofmining in der Analysis. Hierbei betrachten wir ein Konvergenzlemma und verwenden Techniken des Proofminings, um eine Konvergenzrate zu erhalten. Wir wenden das neue Lemma auf verschiedene Fixpunktsätze für asymptotisch schwach kontraktive Abbildungen und deren Varianten an. Bei dem letzten Schritt diskutieren wir nicht nur die Extraktion von quantitativen Informationen – in unserem Fall Konvergenzraten – aus Beweisen, sondern auch die Anwendung von Proofmining, um Sätze zu verallgemeinern und zu kombinieren.

Acknowledgements

I would like to thank the following people for supporting me in writing this thesis.

Helmut Schwichtenberg: He was my supervisor at the Ludwig-Maximilians University and was always ready to answer my questions. During my PhD studies we have published one paper together. He has proofread one of my papers and a part of this thesis.

Peter Schuster: He was my supervisor at the Universities of Verona and Trento. He already assisted me in writing the application for my scholarship at the Istituto Nazionale di Alta Matematica "Francesco Severi". Together with Thomas Powell we have written two papers which form the basis of Chapter 3. He has also proofread some of my other papers and a part of my thesis.

Thomas Powell: During my PhD study we have written three papers which form the basis of Chapter 3 and Chapter 6. I had the honour to visit him at the Technical University of Darmstadt.

Daniel Wessel: We had many discussions about mathematical logic and he answered many of my academic questions. He has proofread one of my papers and a part of this thesis.

Nils Köpp: During my PhD studies Nils was a helpful colleague. Together we have written a paper where among other things the results of Chapter 5 are presented. He has proofread a part of this thesis.

Christoph-Simon Senjak †: In the first year of my PhD study and the years before he supported me as a friend with many discussions about mathematical logic.

Finja Ehlers, Miriam Mahler, Tilman Ritschl, Moritz Scherrmann: They have proofread a part of my thesis.

Finally, I thank the **Istituto Nazionale di Alta Matematica "Francesco Severi"** and the **European Commission** for my Marie Skłodowska-Curie fellowship.

Contents

Chapter 1

Introduction

1.1 Who is this dissertation for

By using various techniques of proof interpretations, we have managed to get new research results in the field of constructive algebra and analysis. We develop a good many new algorithms and vitalize old theorems by adding computational content or even generalizing them.

However, this thesis does not just present new research results. The author of this thesis has chosen mathematical logic as specialty to explore the fine structure of mathematics, and while writing he had in mind to share this insight with the reader. In the following chapters we study structures of individual proofs. We see that even non-constructive principles can become computationally relevant. We even decompose some proofs and let them read by a computer program to generate algorithms and new proofs. On the other hand, we also keep an eye on the big picture by examining the connection between proofs with the goal to combine and generalize them.

With this in mind, also mathematicians whose main subject is not mathematical logic benefit from reading this thesis. The concepts we are presenting will be useful for the reader while writing their own proofs.

1.2 Proof interpretations

In this dissertation, "proof interpretation" means a process which takes one or more mathematical proofs together with their proven theorems as input and returns one or more proofs with their proven theorems as output. This description is deliberately vague. In other works like [127, Definition 1.1] "proof interpretation" is defined

more formally, whereas in this thesis it is rather a heuristic notation. The reason for this is that every mathematical example which interprets proofs in this thesis shall answer the question "What can we learn from a mathematical proof?" by providing an insight into the versatility of the work with proofs and their theorems. Despite all the achievements in the last decades like all the tools of proof interpretation such as Gödel's functional interpretation [77], Friedman's A-Translation [74] or program extraction from constructive proofs [159], to only name a few, there is still a strong need for research.

Our main purpose of proof interpretations is the extraction of computational content, which can be a term or an algorithm. The second purpose is the transformation of some given proofs to proofs of other theorems. This is familiar to every mathematician who has tried to generalize a given theorem. Both aspects are covered in this thesis, albeit the attention is on the first aspect as it is characteristic for constructive mathematics.

Proof interpretations are originally motivated by Hilbert's program [72, 81, 106, 141, 164, 185] which was founded by David Hilbert in the early part of the 20^{th} century. His goal was to prove the consistency of the axiom systems in mathematics. Although Kurt Gödel showed that Hilbert's program is not generally realisable [76, 149], there are many examples in algebra where Hilbert's program works [58]. By his pioneering work on the unwinding of proofs [104, 105], it was Kreisel who founded the field of proof interpretations in modern mathematics. There is a good many applications of his program in e.g. algebra [62], combinatorics [18] and number theory [117]. In the last decades, the application of proof interpretations has become a major topic in proof theory, and today encompasses both proof mining [89, 90, 92], which focuses on obtaining quantitative information primarily from proofs in areas of mathematical analysis, and the mechanized synthesis of programs from proofs, which has found many concrete applications in discrete mathematics and computer science [24, 25, 158].

1.3 Constructive algebra

Constructive algebra is often defined as algebra done within intuitionistic logic [44, 169]. In this work we will go a step further. Our aim is an algorithmic formulation of existence statements. In particular, we describe algorithms and prove that these algorithms fulfil certain properties. The proof itself can be done in the context of classical logic. Our view is inspired by Ihsen Yengui's book "Constructive Commutative Algebra" [184] where constructive algebra is seen as an abstract version of computer algebra. The other two important books on constructive al-

gebra are "Commutative Algebra: Constructive Methods" by Henri Lombardi and Claude Quitté [115] and "A Course in Constructive Algebra" by Ray Mines, Fred Richman and Wim Ruitenburg [121]. These three books are the main references of the material on constructive algebra in this thesis.

In classical algebra non-effective methods like the law of excluded middle, Zorn's lemma, the existence of maximal ideals or some other version of the axiom of choice are used to prove concrete statements mainly in first order logic. Hence, in constructive algebra one puts the emphasis more on structures which are definable in first-order logic and one tries to replace those non-effective methods by finite methods. There are many papers considering this issue written by Sami Barhoumi, Thierry Coquand, Lionel Ducos, Stefan Neuwirth, Hervé Perdry, Marie-Françoise Roy, Peter Schuster, Daniel Wessel and the persons mentioned above [15–17, 35, 41–43, 46–55, 57, 65, 66, 110, 112–115, 126, 130, 131, 174, 183].

Kreisel already discussed the use of proof theoretic techniques to extract quantitative information from proofs in abstract algebra [107], specifically Hilbert's 17^{th} problem together with his Nullstellensatz. Today there are comparatively few formal applications of proof interpretation in algebra, and the computational analysis of it is largely done on a case by case basis. This typically involves replacing semantic conservation theorems with appropriate syntactic counterparts both sufficient for proofs of elementary statements and provable by elementary means. This method has proved possible in numerous different settings [35, 36, 124, 125, 146, 175], and in the context of commutative algebra the so-called dynamical method is especially dominant [57,60,115,183,184]. In dynamical algebra one deals with a supposed ideal object (such as maximal ideals) only by means of concrete, finitary approximations (such as finitely generated ideals, or rather the finite set of generators), where the latter provide partial but sufficiently complete information about the former.

The idea of replacing ideal objects with suitable finitary approximations is already implicit in Kreisel's unwinding program and is captured by his famous no-counterexample interpretation [104, 105]. In our cases, it is like Gödel's functional interpretation and corresponds to the notion of metastability [88, 93, 96], which has been made popular by Tao [167] and has featured in higher order computability theory [148]. Although metastability was originally introduced in the context of analysis, in Chapter 3 we use this approach to define the notion of approximate explicit maximal objects.

The algebraic results of Chapter 3 and 4 can also be seen in the context of the proof mining program, which mainly focuses on the area of analysis, but in the last few years some proof mining has become an interesting topic in algebra, too [163].

1.4 Constructive analysis

Analogously to algebra, constructive analysis can also be defined as analysis done within intuitionistic logic. In Chapter 5 we adopt this approach, which in analysis is even more popular as it is in algebra. One reason for this is that in analysis we often consider concrete objects like real and complex numbers whereas in algebra theorems deal with abstract objects like fields, rings and groups. Two important book which considers analysis in the context of intuitionistic logic are "Foundations of constructive analysis" by Errett Bishop [29] and "Constructive Analysis" by Errett Bishop and Douglas Bridges [30]. In particular, the representation of real numbers plays a main role in analysis with intuitionistic logic. In the last twenty years there have been written many papers on exact real arithmetic by Ulrich Berger, Alberto Ciaffaglione, Pietro Di Gianantonio, Kenji Miyamoto, Monika Seisenberger, Helmut Schwichtenberg and Hideki Tsuiki [21, 26, 38, 63, 123, 170]. In this case intuitionistic logic is used to formalise and prove constructive existential statements, and computer support is used to extract the computational content and prove its soundness. However, we use this method also as heuristic to get algorithms out of constructive proofs at the end of Chapter 5 and even in the context of algebra in Chapter 4. Other very typical examples of analysis with intuitionistic logic are the intermediate value theorem, the mean value theorem and the fundamental theorem of algebra [79, 84, 152, 155, 156], which are existence statements of real numbers. In these cases a concrete constructive representation, for instance of continuous or differentiable functions, is important as, in contrast to the classical analysis, continuity and differentiability come with a modulus. This is why this part of constructive analysis is often called "analysis with witnesses".

Closely related to this topic is the field of computable analysis. This is the study of analysis in the context of computability theory, which goes back to the work of Turing about computable numbers [171]. An introduction to computable analysis is given in [173] by Klaus Weihrauch. In this dissertation computability theory works mainly in the background, i.e. we state algorithms as program code or in natural language. How this code is in detail implemented, for instance as a Turing machine, is not an issue of this dissertation.

Apart from that, the computational meaning of non-constructive proofs in analysis lies also at the heart of mathematical logic. A typical concept in this direction is the concept of proof mining, which is also based on Kreisel's unwinding program. The term "proof mining" goes back to Dana Scott. However, there is no formal definition as proof mining is a quite enormous field. In Chapter 6 we use the techniques of proof mining to extract rates of convergence in fixed point theorems. Our approach is mostly inspired by some papers of Ulrich Kohlenbach, Laurenţiu

Leuştean, Paulo Oliva Thomas Powell and Andrei Sipoş [87, 94, 95, 98–101]. This part of proof mining is motivated by new techniques like the monotone functional interpretation [91] and the refinement of already existing proofs [22, 27]. For a general overview we refer to Kohlenbach's book "Applied Proof Theory: Proof Interpretation and their Use in Mathematics" [90]. In contrast to other techniques of proof interpretation, the end result of proof mining in analysis is again a purely mathematical theorem with a proof involving no notions of mathematical logic. Some tools we use, e.g. Gödel's functional interpretation or formal program extraction from proofs, are only used as an inspiration.

1.5 Overview and results of the dissertation

In the following we give an outline of this dissertation chapter by chapter. The new research results are emphasised by the typeface and we mark all collaborations during the research process.

Chapter 2

In this chapter we give an overview of the theories we use in the dissertation. At this point there are no new results presented and we mainly use [159] as source. The main topics of this chapter are the theory of computable functionals, program extraction from proofs and Gödel's functional translation. For the last two we also state the corresponding soundness theorems. As a short application of Gödel's functional translation and bridge to the next chapter we prove a theorem about metastability in analysis, which is a special case of [95, Proposition 6.4].

Chapter 3

This chapter is an application of proof theory in commutative algebra. We introduce the so-called coverings and use proof theoretic methods to give a computational interpretation of a general maximality principle, which in particular is a generalisation of the existence of maximal ideals in commutative rings. By using ideas of Gödel's functional interpretation and metastability we *develop a notion of an approximation to explicit maximal objects* in the countable case and *describe a state based algorithm*, which computes such approximate explicit maximal objects. We apply our theory to the universal Krull-Lindenbaum lemma – a generalisation of Krull's lemma – and *formulate an algorithmic universal Krull-Lindenbaum lemma*. In several case studies the *algorithmic universal Krull-Lindenbaum lemma is specialized to*

certain areas like radical ideals or filters in commutative rings, valuation rings, ordered fields, complete theories and distributive lattices. The specialization to radical ideals in commutative rings leads to the proper Krull lemma, and the specialization to complete theories leads to Lindenbaum's lemma. In the case of radical ideals, we *give an algorithm for the theorem of Gauß-Joyal and an algorithm for a theorem of nilpotent coefficients of invertible polynomials.* In the case of valuation rings we *get an algorithm for Kronecker's theorem and Dedekind's Prague theorem.* This is the result of joint work with Peter Schuster and Thomas Powell, published in [136, 137].

Chapter 4

On the basis of Zariski's lemma we present a typical manual approach in constructive mathematics to extract algorithms out of constructive proofs, which is quite similar to the automatic program extraction from proofs presented in Chapter 2. We take Zariski's lemma and *formulate a constructive proof of it.* Using this proof as foundation, we *formulate an algorithm together with an algorithm version of Zariski's lemma.* Afterwards we *prove that the algorithm indeed fulfils the algorithmic version.* We conclude with an outlook where we combine the results in this chapter with the results in Chapter 3 to *get a possible algorithmic version of Hilbert's Nullstellensatz.* A part of the work in this chapter is published in [180].

Chapter 5

In contrast to the Chapter 4, this chapter deals with the automatic program extraction from proofs and the exact representation of real numbers by signed digit streams. The signed digit representation of real numbers is basically the binary representation with the additional digit -1. The property of having a signed digit representation is formalised by using a coinductively defined predicate, and *generate an algorithm which takes a sequence of signed digit streams of real numbers with modulus of convergence and returns a signed digit stream of the limit.* Here we use the proof assistant Minlog to *extract a program from our constructive proof* and the programming language Haskell to display the extracted algorithm in a more readable way. As an application of the extracted algorithm we give an *algorithm which converts the signed digit representation of a non-negative real number to the signed digit representation of its square root* by using Heron's method. The computational part of the application is done manually without Minlog and displayed in Haskell notation. As the result of a collaboration with Nils Köpp [181] we finally show how the *convergence theorem can be used to get a function which takes the signed digit stream of two real numbers and returns the signed digit stream of their product.*

Chapter 6

We use typical tools of proof mining on existence theorems about fixed points of variants of asymptotically contractive maps on normed spaces. We study Krasnoselskii-Mann sequences for approximating fixed points of asymptotically weakly contractive maps on normed spaces. As starting point we analyse a convergence lemma with a recursive inequality. We *generalise this lemma and extract a rate of convergence*. The analysed lemma is contained in the proof of the fixed points theorems. We *define a new notion of being asymptotically weakly contractive with modulus*. In several case studies we *formulate a series of abstract convergence theorems which generalise, unify and quantify known results from the literature*. This chapter is a result of joint work with Thomas Powell [138].

1.6 List of publications

The following publications contain some of the research reported in this dissertation

[**136**] Thomas Powell, Peter Schuster, and Franziskus Wiesnet. An algorithmic approach to the existence of ideal objects in commutative algebra. In R. Iemhoff, M. Moortgat, and R. de Queiroz, editors, *Logic, Language, Information and Computation*, volume 11541 of *Lectures Notes in Computer Science*. Springer-Verlag, July 2019.

[**137**] Thomas Powell, Peter Schuster, and Franziskus Wiesnet. A universal algorithm for Krull's theorem. *Information and Computation*, 2021. To appear.

[**138**] Thomas Powell and Franziskus Wiesnet. Rates of convergence for asymptotically weakly contractive mappings in normed spaces. *arXiv preprint arXiv:2104.14495*, 2021. submitted.

[**180**] Franziskus Wiesnet. An algorithmic version of Zariski's lemma. In L. De Mol, F. Manea, and A. Weiermann, editors, *Computability in Europe*, Lecture Notes in Computer Science. Springer, 2021. To appear.

[**181**] Franziskus Wiesnet and Nils Köpp. Limits of real numbers in the binary signed digit representation. *arXiv preprint arXiv:2103.15702*, 2021. submitted.

Chapter 2

Logical background

Motivation 2.0.1. We start by introducing the tools of proof interpretation which are used in this dissertation. These are the formal program extraction from proof, the double-negation translation and Gödel's functional interpretation. The first part of this section is about the formal program extraction from proofs and presents the theory of computable functional which will be the logical framework for program extraction. We go shortly into Heyting and Peano arithmetic in all finite types because Peano arithmetic in all finite types can be used as metatheory of this dissertation except Chapter 5. The second section of this chapter deals with local operators, of which double-negation translation is a special case, and Gödel's functional interpretation. At the end we give a proof about metastability in analysis where the double-negation translation and Gödel's functional interpretation are combined. This is a preparation for the next chapter.

2.1 The theory of computable functionals

Motivation 2.1.1. For a formal program extraction from proofs, we need a logical framework. As such a system we pick the theory of computable functionals **TCF**, a version of constructive arithmetic in finite types where algebras are admitted as base types. It is also used in the proof assistant Minlog as metatheory and one advantage is that inductively and coinductively defined predicates are available, which we use together with there computational content to prove statements about infinite data.

This section is mainly based on the newest research by Helmut Schwichtenberg given in [157]. Other important sources are [159, 178].

The partial continuous functionals in the sense of Scott [161] and Ershov [70] are viewed as the intended (standard) model of **TCF**. As this is not relevant for us,

we do not go into details about the model of **TCF** and just refer to the first part of [157].

We want to emphasise the following: the theory in this section is very formal and in Chapter 5 we see a typical application of it. But notions like program extraction, reliability and soundness can also be seen as informal heuristic to develop algorithms out of proofs. An example of this is given in Chapter 4.

2.1.1 Algebras and types

Motivation 2.1.2. This subsection is about the definition of types. In Definition 2.1.4 constructor types are defined by using types, and in Definition 2.1.6 we define types by using constructor types. This seems to be circular argument but in fact it is not as we formally use recursion over the syntactical structure of the types.

The definitions in this section should be seen syntactically, i.e. types are nothing else than strings.

Notation 2.1.3. By α, β and ξ we denote *type variables*. We write $\vec{\alpha}$ for $\alpha_0, \ldots, \alpha_{k-1}$ and some implicit $k \in \mathbb{N}$.

Definition 2.1.4. We call a type κ a *constructor type* if it has the form

$$\vec{\alpha} \to (\vec{\beta_i} \to \xi)_{i<n} \to \xi$$

where the α_i, β_{ij} are types (defined below) which do not contain ξ. The $\vec{\alpha}$ are called *parameter types* and the $\vec{\beta_i} \to \xi$ are called *recursive types*. If $\vec{\kappa} := \kappa_0, \ldots, \kappa_{k-1}$ for $k > 0$ are constructor types, we call

$$\iota := \mu_\xi(\vec{\kappa})$$

an *algebra*.

Example 2.1.5. In the following we give some important algebras, where we include the name of the constructors in the notion.

$$\begin{aligned}
&\mathbb{U} := \mu_\xi(\mathrm{Dummy} : \xi) && \text{(unit)},\\
&\mathbb{B} := \mu_\xi(\mathrm{tt} : \xi,\ \mathrm{ff} : \xi) && \text{(booleans)},\\
&\mathbb{D} := \mu_\xi(\mathrm{SdR} : \xi,\ \mathrm{SdM} : \xi,\ \mathrm{SdL} : \xi) && \text{(signed digits, for 1, 0, -1)},\\
&\mathbb{S} := \mu_\xi(\mathrm{C} : \mathbb{D} \to \xi \to \xi) && \text{(siged digit streams)},\\
&\mathbb{N} := \mu_\xi(0 : \xi,\ \mathrm{S} : \xi \to \xi) && \text{(natural numbers)},\\
&\mathbb{P} := \mu_\xi(1 : \xi,\ \mathrm{S}_0 : \xi \to \xi,\ \mathrm{S}_1 : \xi \to \xi) && \text{(binary positive numbers)},\\
&\mathbb{Z} := \mu_\xi(0 : \mathbb{Z},\ + : \mathbb{P} \to \xi,\ - : \mathbb{P} \to \xi) && \text{(integers)}.
\end{aligned}$$

Positive numbers are also denoted by $\mathbb{Z}^+$. Algebras with type parameter are

$$\begin{aligned}
\alpha \times \beta &:= \mu_\xi(\langle .,. \rangle : \alpha \to \beta \to \xi) && \text{(type product)},\\
\alpha + \beta &:= \mu_\xi(\mathrm{in}_0 : \alpha \to \xi,\ \mathrm{in}_1 : \beta \to \xi) && \text{(type sum)},\\
\mathbb{L}(\alpha) &:= \mu_\xi([] : \xi,\ \mathrm{Cons} : \alpha \to \xi \to \xi) && \text{(lists of } \alpha),
\end{aligned}$$

The list type $\mathbb{L}(\alpha)$ will also be denoted by α^*. The rational numbers $\mathbb{Q}$ are defined by $\mathbb{Q} := \mathbb{Z} \times \mathbb{P}$.

Definition 2.1.6. We define *types* recursively as follows:

- Each type variable is a type.
- If $\kappa_0, \ldots, \kappa_{k-1}$ are of constructor types with $k > 0$, $\mu_\xi(\vec{\kappa})$ is a type.
- If τ and σ are types, $\tau \to \sigma$ is a type.

2.1.2 Terms

Motivation 2.1.7. In **TCF** each term t comes together with its type τ. One can write t^τ to make the type explicit. Again, a term is also just a string (or a pair of two strings to include the type).

Similar to the definition of terms, the definition of terms in **TCF** is done by recursion over the syntactical structure of their type.

Definition 2.1.8. Let an algebra $\iota = \mu_\xi(\vec{\kappa})$ with k constructor types $\vec{\kappa} := \kappa_0(\xi), \ldots, \kappa_{k-1}(\xi)$ for $k > 0$ be given. For each constructor type $\kappa_i(\xi)$ we define a *constructor* C_i with type $\kappa_i(\iota)$.

Definition 2.1.9. *Terms* are defined recursively and each term comes with a type.

- Each typed term variable is a term.
- Each constructor given in Definition 2.1.8 is a term with the type given in this definition.
- If t is a term of type σ and x is a variable of type τ then $\lambda_x t$ is a term of type $\tau \to \sigma$.
- If t is a term of type $\tau \to \sigma$ and s is a term of type τ then ts is a term of type σ.

- Each program constant (given in the next definition) is a term with the corresponding type.

If t is a term of type σ, s is a term of type τ and x is a variable of type τ, we identify the terms $(\lambda_x t(x))s$ and $t(s)$. If furthermore r is a term of type $\tau \to \sigma$, we identify $\lambda_x(rx)$ and r.

Definition 2.1.10. A *program constant* D (also called *defined constant*) is given by its type and a list of computational rules. For the formal definition of a program constant we refer to [159, Section 6.2.4] and [178, Definiton 1.2.16] and [157]. For each computational rule $D\vec{t} := s$, we identify the term $D\vec{t}$ with the term s.

The program constants we mainly use in this dissertation are the following: For each algebra ι there is the *recursion operator* $\mathcal{R}^\tau_\iota$ in a type τ, the *destructor* $\mathcal{D}_\iota$ and the *corecursion operator* ${}^{co}\mathcal{R}^\tau_\iota$ from a type τ. Concrete definitions of these program constants are given in [157, Section 2.1] and [159, 178].

Example 2.1.11. For the algebra $\mathbb{S}$ of signed digit streams the recursion operator $\mathcal{R}^\tau_{\mathcal{S}}$ is given by

$$\begin{aligned}\mathcal{R}^\tau_{\mathbb{S}} &: \mathbb{S} \to (\mathbb{D} \to \mathbb{S} \to \tau \to \tau) \to \tau\\ \mathcal{R}^\tau_{\mathbb{S}}(\mathrm{C}dv)f &:= fdv(\mathcal{R}^\tau_{\mathbb{S}}vf).\end{aligned}$$

Its destructor is given by

$$\begin{aligned}\mathcal{D}_{\mathbb{S}} &: \mathbb{S} \to \mathbb{D} \times \mathbb{S}\\ \mathcal{D}_{\mathbb{S}}(\mathrm{C}dv) &:= \langle d, v\rangle.\end{aligned}$$

Its corecursion operator is given by

$$\begin{aligned}{}^{co}\mathcal{R}^\tau_{\mathbb{S}} &: \tau \to (\tau \to \mathbb{D} \times (\mathbb{S} + \tau)) \to \mathbb{S}\\ {}^{co}\mathcal{R}^\tau_{\mathbb{S}}xf &:= \mathrm{C}(\pi_0(fx))([\mathrm{id}, \lambda_y {}^{co}\mathcal{R}^\tau_{\mathbb{S}}yf](\pi_1(fx)))\end{aligned}$$

where $\mathrm{id} := \lambda_x x$ is the identity function, $\pi_i : \alpha_0 \times \alpha_1 \to \alpha_i$ is the *projection* on the i-th component with the computation rule $\pi_i\langle a_0, a_1\rangle := a_i$, and $[F_0, F_1]\mathrm{in}_i T := F_i T$.

This example is used in Chapter 5.

2.1.3 Predicates and formulas

Definition 2.1.12. In **TCF** *predicate* are either (co)inductively defined predicates (given below) or of the form $\{\vec{x} \mid A(\vec{x})\}$, where $A(\vec{x})$ is a formula. We identify $\{\vec{x} \mid A(\vec{x})\}\vec{t}$ with $A(\vec{t})$. *Formulas* are given by atomic formulas $P\vec{t}$, where P is a predicate and $\vec{t}$ are well-typed terms, and given by $\forall_x A$ and $A \to B$, where A and B are already formulas.

Definition 2.1.13. An *inductively defined predicate* I is given by clauses of the form

$$I_i^+ : \forall_{\vec{x_i}}((A_{ij}(I))_{j<n_i} \to I\vec{t_i}),$$

for $i \in \{0, \ldots, k-1\}$, $k > 0$ and $n_i \geq 0$, where I only occurs strictly positive in each $A_{ij}(I)$ and $\vec{x_i}$ contains all free variables of $A_{ij}(I)$ and $\vec{t_i}$. The clauses are the introduction axioms of I. Intuitively, an inductively defined predicate is the smallest predicate w.r.t. $\subseteq$, which fulfils its clauses. Formally, this is expressed by its elimination axiom

$$I^- : (\forall_{\vec{x_i}}((A_{ij}(I \cap X))_{j<n_i} \to X\vec{t_i}))_{i<k} \to I \subseteq X,$$

where X can be any predicate with the same arity as I. This axiom is also called *induction axiom*.

Example 2.1.14. For each type τ we have the *Leibniz equality* $\equiv$ given by the single introduction rule $\forall_x x \equiv x$. The elimination rule of the Leibniz equality is

$$\forall_{x,y}(x \equiv y \to \forall_x(Pxx) \to Pxy).$$

Given two formulas A and B, the formula $A \wedge B$ is the inductively defined predicate given by the clause $\wedge^+ : A \to B \to A \wedge B$ and $A \vee B$ is the inductively defined predicate given by the two clauses $\vee_0^+ : A \to A \vee B$ and $\vee_1^+ : B \to A \vee B$. Another important example is the existential quantifier $\exists_x A(x)$ which is given by the clause $\exists^+ : \forall_x(A(x) \to \exists_x A(x))$. For these examples we have the following elimination rules:

$$\begin{aligned} \wedge^- &: A \wedge B \to (A \to B \to C) \to C \\ \vee^- &: A \vee B \to (A \to C) \to (B \to C) \to C \\ \exists_x A &: \exists_x A \to \forall_x(A \to B) \to B \end{aligned}$$

Example 2.1.15. A further important class of predicates are the *totality predicates*. For each type τ there exists the totality predicate $\mathbf{T}_\tau$. For a formal definition we refer to [159, Section 7.1.6.] and [178, Definition 1.5.3] (There it is called "absolute totality"). Here we just give the examples which we need in Chapter 5. The totality predicate $\mathbf{T}_\mathbb{N}$ of the natural numbers has the following clauses:

$$\begin{aligned} (\mathbf{T}_\mathbb{N})_0^+ &: \mathbf{T}_\mathbb{N}0 \\ (\mathbf{T}_\mathbb{N})_1^+ &: \forall_n(\mathbf{T}_\mathbb{N}n \to \mathbf{T}_\mathbb{N}\mathrm{S}n) \end{aligned}$$

Note that the elimination axiom of the totality for natural numbers is exactly the well-known induction axiom over natural numbers. The totality of the positive binary numbers is defined similarly:

$$(\mathbf{T}_{\mathbb{P}})^+_0 : \mathbf{T}_{\mathbb{P}}0$$
$$(\mathbf{T}_{\mathbb{P}})^+_1 : \forall_p(\mathbf{T}_{\mathbb{P}}p \to \mathbf{T}_{\mathbb{P}}\mathrm{S}_0 p)$$
$$(\mathbf{T}_{\mathbb{P}})^+_2 : \forall_p(\mathbf{T}_{\mathbb{P}}p \to \mathbf{T}_{\mathbb{P}}\mathrm{S}_1 p)$$

As last example we give the clause for the totality of the function type $\mathbb{P} \to \mathbb{N}$:

$$(\mathbf{T}_{\mathbb{P}\to\mathbb{N}})^+ : \forall_f(\forall_p(\mathbf{T}_{\mathbb{P}}p \to \mathbf{T}_{\mathbb{N}}fp) \to \mathbf{T}_{\mathbb{P}\to\mathbb{N}}f)$$

Definition 2.1.16. To each inductively defined predicate I given by its clauses $(I_i)_{i<k}$, there exists the associated *coinductively defined predicate* ${}^{co}I$. The elimination axiom ${}^{co}I^-$, also called *closure axiom*, of the coinductively defined predicate says that if ${}^{co}I\vec{x}$ holds, it comes from one of the clauses:

$${}^{co}I^- : \forall_{\vec{x}}\Big({}^{co}I\vec{x} \to \bigvee_{i<k} \exists_{\vec{x_i}}\Big(\bigwedge_{j<n_i} A_{ij}({}^{co}I) \land \vec{x} \equiv \vec{t_i}\Big)\Big).$$

Here $\vec{a} \equiv \vec{b}$ is meant as $a_0 \equiv b_0 \land \cdots \land a_{l-1} \equiv b_{l-1}$. In other words, if ${}^{co}I\vec{x}$ holds, there is at least one clause I_i^+ whose premises $(A_{ij}({}^{co}I))_{j<n_i}$ (with ${}^{co}I$ instead of I) are fulfilled and whose conclusion is ${}^{co}I\vec{x}$ up to Leibniz equality. The introduction axiom ${}^{co}I^+$ of ${}^{co}I$ says that ${}^{co}I$ is the greatest predicate, which fulfils this elimination property:

$${}^{co}I^+ : \forall_{\vec{x}}\Big(X\vec{x} \to \bigvee_{i<k} \exists_{\vec{x_i}}\Big(\bigwedge_{j<n_i} A_{ij}({}^{co}I \cup X) \land \vec{x} \equiv \vec{t_i}\Big)\Big) \to X \subseteq {}^{co}I$$

It is also called *coinduction axiom* or *greatest-fixed-point axiom*.

2.1.4 Computational content

Motivation 2.1.17. In **TCF** the computational content is expected to be a term of a certain type. By introducing non-computational and computationally relevant formulas, we determine for which formulas we compute a computational content. The extracted term of a computational relevant formula is defined by recursion over its proof and can be interpreted as the computational content.

Definition 2.1.18. To each predicate variable X, inductively defined predicate I and coinductively defined predicate ${}^{co}I$ we define the non-computational version X^{nc}, I^{nc} and ${}^{co}I^{nc}$. The axioms of I^{nc} and ${}^{co}I^{nc}$ are the same with the exception that the competitor predicate in I^{nc-} and ${}^{co}I^{nc+}$ must be non-computational.

A formula is *non-computational* (n.c.) if its final conclusion is non-computational, otherwise we call it *computationally relevant* (c.r.).

Definition 2.1.19. We define the type of a c.r. formula as follows:

The clauses of a c.r. inductively defined predicate I determine an algebra ι_I by

$$\iota_I := \mu_{\tau(I)}(\tau(I_0^+), \dots, \tau(I_{k-1}^+)),$$

where $I_0^+, \dots, I_{k-1}^+$ are exactly the clauses of I. The notation $\mu_{\tau(I)}$ means that we replace each occurrence of $\tau(I)$ in the formal definition of the type of a formula by the type variable ξ and after this we build the algebra over the variable ξ.

For a formula A built from atomic formulas by $\to$, $\forall$ whose final conclusion is c.r. we define its *type* $\tau(A)$ by

$$\begin{aligned}
\tau(P\vec{t}) &:= \tau(P), \\
\tau(A \to B) &:= \begin{cases} \tau(A) \to \tau(B) & \text{if } A \text{ is c.r.} \\ \tau(B) & \text{if } A \text{ is n.c.} \end{cases} \\
\tau(\forall_z A) &:= \tau(A), \\
\tau(I) &:= \tau({}^{co}I) := \iota_I
\end{aligned}$$

To define $\tau(X)$ for a c.r. predicate variable X we need a type variable α_X uniquely associated to X. Then we define $\tau(X) := \alpha_X$. Furthermore, we define $\tau(\{\vec{x} \mid A\}) := \tau(A)$.

Example 2.1.20. We continue Example 2.1.14:

The Leibniz equality is defined as non-computational.

Let A and B be c.r. formulas. The type of $A \wedge B$ is the algebra given by the constructor $C : \tau(A) \to \tau(B) \to \tau(A \wedge B)$, which is the type product $\tau(A) \times \tau(B)$.

The type of $A \vee B$ is the algebra given by the constructors $C_0 : \tau(A) \to \tau(A \vee B)$ and $C_1 : \tau(B) \to \tau(A \vee B)$, which is the type sum $\tau(A) + \tau(B)$. The constructors are denoted by in_0 and in_1.

The type of $\exists_x A(x)$ is the algebra given by the constructor $C : \tau(A) \to \tau(\exists_x A)$. This algebra is the identity algebra of $\tau(A)$ and we identify it with $\tau(A)$ itself and

the term Cx is identified with x. Altogether, we have

$$\begin{aligned}\tau(A \wedge B) &= \tau(A) \times \tau(B),\\ \tau(A \vee B) &= \tau(A) + \tau(B),\\ \tau(\exists_x A) &= \tau(A)\end{aligned}$$

for c.r. formulas A and B.

Example 2.1.21. For the totality predicate $\mathbf{T}_\tau$ from Example 2.1.15 there are the computational relevant version $\mathbf{T}_\tau$ and the the non-computational version $\mathbf{T}_\tau^{nc}$. The type of the computational relevant totality predicate is the underlying type, i.e. $\tau(\mathbf{T}_\tau) = \tau$. By using the computational relevant totality predicate, one can transfer a variable to the extracted term. We use this method in the formulation of Theorem 5.3.12.

Definition 2.1.22. Let M be a proof in **TCF** of a c.r. formula A. We define its *extracted term* $\mathrm{et}(M)$ of type $\tau(A)$ as follows: To define $\mathrm{et}(u^A)$ for an assumption u of the formula A we need a variable z_u of type $\tau(A)$ uniquely associated to u, then $\mathrm{et}(u^A) := z_u$. For the rule of $\forall$ and $\to$ we define the extracted term as follows:

$$\begin{aligned}\mathrm{et}((\lambda_{u^A} M^B)^{A\to B}) &:= \begin{cases}\lambda_{z_u}\mathrm{et}(M) & \text{if } A \text{ is c.r.}\\ \mathrm{et}(M) & \text{if } A \text{ is n.c.}\end{cases}\\ \mathrm{et}((M^{A\to B} N^A)^B) &:= \begin{cases}\mathrm{et}(M)\mathrm{et}(N) & \text{if } A \text{ is c.r.}\\ \mathrm{et}(M) & \text{if } A \text{ is n.c.}\end{cases}\\ \mathrm{et}((\lambda_x M^A)^{\forall_x A}) &:= \mathrm{et}(M)\\ \mathrm{et}((M^{\forall_x A(x)} t)^{A(t)}) &:= \mathrm{et}(M).\end{aligned}$$

It remains to define extracted terms for the axioms. Let I be a c.r. inductively defined predicate. The extracted term $\mathrm{et}(I_i^+)$ of the i-th clause is the i-th constructor C_i of the algebra ι_I. The extracted term of I^- is given by the recursion operator $\mathcal{R}_{\iota_I}^{\tau(P)}$, where $\tau(P)$ is the type of the competitor predicate. The extracted term of ${}^{co}I^-$ is the destructor $\mathcal{D}_{\iota_I}$, and the extracted term of ${}^{co}I^-$ is the corecursion operator ${}^{co}\mathcal{R}_{\iota_I}^{\tau(P)}$, where $\tau(P)$ is the type of the competitor predicate.

2.1.5 Soundness of program extraction

Motivation 2.1.23. Using the definition above, we are able to extract program (i.e. terms) from proofs in **TCF**. It remains to give a meaning to the extracted

term. For this we use the realisability predicate and then we formulate the soundness theorem, which says that the extracted term of a proof is indeed a realiser of the proven formula. In this proof we need the *invariance axiom*.

Definition 2.1.24. To each c.r. predicate P there exists a *realisability predicate* $P^{\mathbf{r}}$ with the same arity plus an additional argument of type $\tau(P)$. For a formal definition of the realisability predicate we refer to [157, Section 4.1].

We call a formula **r**-free if it does not contain any realisability predicates, and we call a proof **r**-free if it contains **r**-free formulas only.

Definition 2.1.25. For each c.r. formula A we define the following *invariance axiom*

$$\exists_z(z \ \mathbf{r} \ A) \leftrightarrow A.$$

In particular, it says that a c.r. formula is true if and only if it can be realised.

Theorem 2.1.26 (Soundness theorem of program extraction). Let M be an **r**-free proof of a formula A from assumptions $(u_i : C_i)_{i<n}$. Then

$$\begin{cases} \mathrm{et}(M) \ \mathbf{r} \ A & \text{if } A \text{ is c.r.} \\ A & \text{if } A \text{ is n.c.} \end{cases}$$

is derivable from the assumptions

$$\begin{cases} z_{u_i} \ \mathbf{r} \ C_i & \text{if } C_i \text{ is c.r.} \\ C_i & \text{if } C_i \text{ is n.c.} \end{cases}$$

for $i < n$ and the invariance axiom.

Proof. We refer to [157, Theorem 4.10]. □

2.1.6 Heyting and Peano arithmetic in all finite types

Motivation 2.1.27. We have introduced the theory of computational functional and in Chapter 5 we use it as meta theory. However, **TCF** is quite comprehensive and therefore often not necessary. In this short section we introduce Heyting and Peano arithmetic in all finite type, and in all the other chapters it suffices to consider Peano arithmetic in finite types as meta theory.

We also use Heyting and Peano arithmetic in all finite types to define Gödel's functional interpretation in the next section, which operates on formulas of Heyting arithmetic in all finite types (which are identical to the formulas of Peano arithmetic in all finite types).

Definition 2.1.28. The theory of *Heyting arithmetic* $\mathbf{HA}^\omega$ in all finite type is a fragment of **TCF**:

- The types in $\mathbf{HA}^\omega$ are $\mathbb{N}$, $\mathbb{B}$, $\mathbb{U}$, list types, product types, sum types and function types.
- The predicates in $\mathbf{HA}^\omega$ are totality, Leibniz equality, the existential quantifier, the conjunction and the disjunction, where the last three predicates are seen as logical operators.

Peano arithmetic $\mathbf{PA}^\omega$ in all finite types is $\mathbf{HA}^\omega$ plus the law of excluded middle, i.e. $A \vee \neg A$ for all formulas A. In Heyting and Peano arithmetic all objects are implicitly total.

In particular, case distinction of Boolean terms and induction over natural numbers are allowed in $\mathbf{HA}^\omega$ and $\mathbf{PA}^\omega$.

Definition 2.1.29. We define the *degree of a type* as follows:

$$\begin{aligned} \deg(\mathbb{N}) := \deg(\mathbb{B}) := \deg(\mathbb{U}) &:= 0 \\ \deg(\tau^*) &:= \deg(\tau) \\ \deg(\tau + \rho) := \deg(\tau \times \rho) &:= \max\{\deg(\tau), \deg(\rho)\} \\ \deg(\tau \to \rho) &:= \max\{\deg(\tau) + 1, \deg(\rho)\} \end{aligned}$$

Recall that τ^* denotes the list type over τ. A type τ with $\deg(\tau) = 0$ is called *base type*.

Remark 2.1.30. Note that a base type can be coded as natural number. In particular, a base type can be seen as a type which can be embedded into the natural numbers.

2.2 Tools of proof interpretation

Motivation 2.2.1. In Definition 2.1.22 we have introduced the extracted term which is used to get constructive content out of proofs which are formulated in **TCF**. In this section we introduce local operators and Gödel's functional interpretation, which are also tools to get more information out of proofs. In a Nutshell, all the following chapters are showing how to (formally or informally) use these tools to get constructive content out of several classes of proofs.

We want to compare our work with natural sciences: Whereas natural scientists are studying the nature and extract information out of nature's behaviour, in this

thesis we are extracting information out of proofs occurring in mathematics. Here we consider many types of proofs, even proofs with non-constructive moments like Zorn's Lemma or the law of excluded middle.

In the last subsection we present a short example where the two new tools are used to get quantitative information out of a classical theorem which is a foretaste of the next chapter.

2.2.1 Local operators

Motivation 2.2.2. Inspired by [31] we define local operators and in particular the double-negation translation, which are operating on the set of formulas in a given language. Afterwards we prove some properties of local operators. To give a formal proof, one has to fix a proof calculus like the calculus of natural deduction. To stay more general and short, we do not introduce a certain proof calculus and instead give informal proofs.

However, we use the notation $\Gamma \vdash A$ for "A is derivable from finitely many formulas of Γ". For a formal definition in terms of the calculus of natural deduction we refer to [159, Section 1.1]. We write $\Gamma \vdash_c A$ for $\Gamma \cup \mathrm{Stab} \vdash A$, where Stab consists of all formulas of the form $\neg\neg A \to A$.

Definition 2.2.3. A *local operator* is a map ∇ between formulas with the following properties

$$\begin{aligned} A &\to \nabla A, \\ \nabla(\nabla A) &\to \nabla A, \\ \nabla(A \wedge B) &\leftrightarrow \nabla A \wedge \nabla B, \\ (A \leftrightarrow B) &\to (\nabla A \leftrightarrow \nabla B) \end{aligned}$$

for all formulas A and B. Furthermore we require that ∇ does not change the free variables and commutes with substitution, i.e. $(\nabla A)[t/x]$ and $\nabla(A[t/x])$ are the same formulas, and $FV(\nabla A) \subseteq FV(A)$.

Lemma 2.2.4. The map ∇ on formulas given by $\nabla A := \neg\neg A$ is a local operator.

Proof. The first property follows trivially as $\neg\neg A := (A \to \bot) \to \bot$. The second property is a special case of $\neg\neg\neg A \to \neg A$.
For the third property we give an informal proof (note that a formal proof depends on the underlying proof calculus):

$\to$: Assume $\neg\neg(A \wedge B)$. Our goal is $\neg\neg A$ and $\neg\neg B$. We show only $\neg\neg A$ as $\neg\neg B$ is analogously shown. Therefore, assume $\neg A$ i.e. $A \to \bot$. This leads directly to $A \wedge B \to \bot$ and together with $\neg\neg(A \wedge B)$ we have $\bot$.

$\leftarrow$: From the assumptions $\neg\neg A$, $\neg\neg B$ and $\neg(A \wedge B)$ we have to prove $\bot$: Assume A and B, then $\bot$ by $\neg(A \wedge B)$. Therefore, $\neg B$ under the assumption A. By $\neg\neg B$, it follows $\bot$ under the assumption A. Therefore, $\neg A$, and by $\neg\neg A$, we have $\bot$.

The fourth property follows directly since each logical operator conserves equivalence.
The two requirements are also obviously fulfilled. □

Example 2.2.5. In addition to the double negation from the lemma above there are of course many more examples. The trivial examples are $\nabla A := A$ and $\nabla A := \top$. Another example is $\nabla A := B \vee A$ for a fixed closed formula B. Here it is simple to verify the axioms of a local operator.

Definition 2.2.6. Let ∇ be a local operator. The ∇-*translation* A^∇ of a formula A is recursively defined as follows:

$$\begin{aligned}
(P\vec{t})^\nabla &:= \nabla(P\vec{t}) \\
(A \circ B)^\nabla &:= (A^\nabla \circ B^\nabla) \quad \text{for } \circ \in \{\wedge, \to\} \\
(A \vee B)^\nabla &:= \nabla(A^\nabla \vee B^\nabla) \\
(\forall_x A)^\nabla &:= \forall_x A^\nabla \\
(\exists_x A)^\nabla &:= \nabla(\exists_x A^\nabla)
\end{aligned}$$

Here A, B are formulas, P is a predicate, x is a variable and $\vec{t}$ are terms. In the case $\nabla = \neg\neg$ we call $A^G := A^\nabla$ the *Gödel-Gentzen translation* or the *Double-negation translation*.

If $\nabla = B\vee$ for some fix formula B, the ∇-translation can be called *Friedman translation* as it goes back to Friedman [74].

If Γ is a set of formulas, we define $\Gamma^\nabla := \{A^\nabla \mid A \in \Gamma\}$

Lemma 2.2.7. Let ∇ be a local operator and A,B are formulas. Then

$$(A \to B) \to (\nabla A \to \nabla B)$$

and even

$$\nabla(A \to B) \to (\nabla A \to \nabla B)$$

are derivable.

Proof. We use that $(A \to B) \leftrightarrow (A \wedge B \leftrightarrow A)$ is derivable. Therefore, the first formula follows by using the third and the fourth property of local operators.
To prove the second formula, we assume $\nabla(A \to B)$ and ∇A. By the third property we get $\nabla((A \to B) \wedge A)$. As $((A \to B) \wedge A) \to B$ it follows ∇B by the first formula. □

Lemma 2.2.8. Let ∇ be a local operator and A a formula, then $\nabla A^\nabla \to A^\nabla$ is derivable.

Proof. Induction over the formula A. In the case that A is an existential statement, a disjunction or a prime formula the claim follows by the second property of a local operator.

Let $A = B \wedge C$. By the third axiom we have $\nabla(B^\nabla \wedge C^\nabla) \leftrightarrow \nabla B^\nabla \wedge \nabla C^\nabla$, and by using the induction hypothesis on B and C, we are done.

Let $A = B \to C$. We assume $\nabla(B^\nabla \to C^\nabla)$ and B^∇, and the goal is C^∇. By the induction hypothesis it is enough to show ∇C^∇. From B^∇ we get ∇B^∇ by the first property of a local operator, and by the third property we have

$$\nabla(B^\nabla \to C^\nabla) \wedge \nabla B^\nabla \leftrightarrow \nabla((B^\nabla \to C^\nabla) \wedge B^\nabla).$$

Furthermore, by Lemma 2.2.7 we have

$$\nabla((B^\nabla \to C^\nabla) \wedge B^\nabla \to C^\nabla) \to \nabla((B^\nabla \to C^\nabla) \wedge B^\nabla) \to \nabla C^\nabla.$$

As $\nabla((B^\nabla \to C^\nabla) \wedge B^\nabla \to C^\nabla)$ is true by the first property of a local operator, it follows ∇C^∇.

For the last case let $A = \forall_x B$, so we have $\nabla A^\nabla := \nabla(\forall_x B^\nabla)$. Given a fixed variable x we get ∇B^∇ by Lemma 2.2.7. Using the induction hypothesis this leads to B^∇. Since x was arbitrary, the proof is finished. □

Theorem 2.2.9. Let A be a formula and Γ be a set of formulas with $\Gamma \vdash A$. Furthermore, let ∇ be a local operator. Then

$$\Gamma^\nabla \vdash A^\nabla.$$

Proof. From a derivation M of a formula A with assumption set Γ one has to construct a derivation M' of the formula A^∇ with assumption set Γ^∇. This is done by induction on the underlying proof calculus. We refer to [159, Section 1.1.9] where the statement is proved in the case of the calculus of natural deduction and ∇ being the double-negation translation. Since the proof uses the properties from Definition 2.2.3 only, it can be easily generalised for arbitrary ∇. □

Corollary 2.2.10. Let A be a formula and Γ be a set of formulas with $\Gamma \vdash_c A$, then $\Gamma^G \vdash A^G$.

Proof. By assumption we have $\Gamma \cup \mathrm{Stab} \vdash A$. By the theorem above, we get $\Gamma^G \cup \mathrm{Stab}^G \vdash A^G$. Using induction over formulas, one can prove that $(\neg\neg A \to A)^G$ is derivable in minimal logic. Therefore, it follows $\Gamma^G \vdash A^G$. □

Remark 2.2.11. This corollary provides a non-trivial way to transform classical proofs into intuitionistic proofs. In particular, we are able to transform proofs in Peano arithmetic into proofs in Heyting arithmetic. The next sections shows how one can get some computational content out of a proof in Heyting arithmetic. Therefore, by using the Gödel-Gentzen translation we even get some information out of a classical proof.

2.2.2 Gödel's functional interpretation

Motivation 2.2.12. An important tool we will use is Gödel's Dialectica interpretation. It was introduced in [77]. We will use the formulation which is given in [159]. The Dialectica interpretation is a map which assigns a formula A to a formula of the form $\exists_x \forall_y A_0(x,y)$, where A_0 is quantifier-free and therefore decidable. We will see that the Dialectica interpretation of a formula is equivalent to the formula itself under certain assumption, which are as non-constructive as possible. First, we start with the type of the variables x and y in $\exists_x \forall_y A_0(x,y)$, where $\tau^+(A)$ shall be the type of x and $\tau^-(A)$ shall be the type of y:

Definition 2.2.13. To each formula A in $\mathbf{HA}^\omega$ we define the *positive type* $\tau^+(A)$ and the *negative type* $\tau^-(A)$ by recursion as follows:

$$\begin{aligned}
\tau^+(P\vec{t}) &:= \mathbb{U} & \tau^-(P\vec{t}) &:= \mathbb{U} \\
\tau^+(\forall_{x^\rho} A) &:= \rho \to \tau^+(A) & \tau^-(\forall_{x^\rho} A) &:= \rho \times \tau^-(A) \\
\tau^+(A \wedge B) &:= \tau^+(A) \times \tau^+(B) & \tau^-(A \wedge B) &:= \tau^-(A) \times \tau^-(B) \\
\tau^+(A \vee B) &:= \mathbb{B} \times \big(\tau^+(A) \times \tau^+(B)\big) & \tau^-(A \vee B) &:= \tau^-(A) \times \tau^-(B) \\
\tau^+(\exists_{x^\rho} A) &:= \rho \times \tau^+(A) & \tau^-(\exists_{x^\rho} A) &:= \tau^-(A)
\end{aligned}$$

$$\begin{aligned}
\tau^+(A \to B) &:= \big(\tau^+(A) \to \tau^+(B)\big) \times \big(\tau^+(A) \to \tau^-(B) \to \tau^-(A)\big) \\
\tau^-(A \to B) &:= \tau^+(A) \times \tau^-(B)
\end{aligned}$$

Since $\mathbb{U}$ is the unit algebra with the only constructor Dummy : $\mathbb{U}$, we identify $\mathbb{U} \times \tau$, $\tau \times \mathbb{U}$ and $\mathbb{U} \to \tau$ with τ, and we identify $\tau \to \mathbb{U}$ with $\mathbb{U}$.

Example 2.2.14. We will often use the functional interpretation in combination with the double-negation translation. Therefore, we take a look at the negation of a formula A:

$$\begin{aligned}
\tau^+(\neg A) = \tau^+(A \to \bot) &= \big(\tau^+(A) \to \tau^+(\bot)\big) \times \big(\tau^+(A) \to \tau^-(\bot) \to \tau^-(A)\big) \\
&= \tau^+(A) \to \tau^-(A)
\end{aligned}$$

and

$$\tau^-(\neg A) = \tau^+(A) \times \tau^-(\bot) = \tau^+(A)$$

For the double negation we get

$$\begin{aligned}\tau^+(\neg\neg A) &= \big(\tau^+(\neg A) \to \tau^+(\bot)\big) \times \big(\tau^+(\neg A) \to \tau^-(\bot) \to \tau^-(\neg A)\big)\\ &= \big(\tau^+(A) \to \tau^-(A)\big) \to \tau^+(A)\end{aligned}$$

and

$$\tau^-(\neg\neg A) = \tau^+(\neg A) \times \tau^-(\bot) = \tau^+(A) \to \tau^-(A).$$

Definition 2.2.15. For each formula A and terms $r : \tau^+(a)$,$s : \tau^-(A)$ we define the quantifier-free formula $|A|^r_s$ by recursion as follows:

$$\begin{aligned}|P\vec{t}|^r_s &:= P\vec{t}\\ |\forall_x A(x)|^r_s &:= |A(s_0)|^{r(s_0)}_{s_1}\\ |A \wedge B|^r_s &:= |A|^{r_0}_{s_0} \wedge |B|^{r_1}_{s_1}\\ |A \vee B|^r_s &:= \big(r_0 = 0 \to |A|^{r_{10}}_{s_0}\big) \wedge \big(r_0 = 1 \to |A|^{r_{11}}_{s_1}\big)\\ |\exists_x A(x)|^r_s &:= |A(r_0)|^{r_1}_s\\ |A \to B|^r_s &:= |A|^{s_0}_{r_1 s_0 s_1} \to |B|^{r_0 s_0}_{s_1}\end{aligned}$$

Here, for a term t of a product type $\tau \times \rho$, we denote the left component by t_0 and the right component by t_1.

We define $A^D := \exists_x \forall_y |A|^x_y$ as the *Gödel translation* or the *functional interpretation* and $A_D(x, y) := |A|^x_y$ as the *Gödel kernel* of A.

Remark 2.2.16. We can describe this in a more readable way by writing terms of a pair type in pair form:

$$\begin{aligned}|P\vec{t}|^r_s &:= P\vec{t}\\ |\forall_x A(x)|^f_{y,z} &:= |A(y)|^{fy}_z\\ |A \wedge B|^{x,y}_{u,v} &:= |A|^x_u \wedge |B|^y_v\\ |A \vee B|^{b,x,y}_{u,v} &:= (b = 0 \to |A|^x_u) \wedge (b = 1 \to |A|^y_v)\\ |\exists_x A(x)|^{y,z}_s &:= |A(y)|^z_s\\ |A \to B|^{f,g}_{u,v} &:= |A|^u_{guv} \to |B|^{fu}_v\end{aligned}$$

Remark 2.2.17. The functional interpretation transforms each formula in a Σ_2 formula. A well-known visualisation of a Σ_2 formula $\exists_{x^X}\forall_{y^Y}A_0(x,y)$ is to see it as a game between two players:
Player 1 one picks an element x_0 of X then Player 2 picks an element y_0 of Y. If $A_0(x_0,y_0)$ holds, Player 1 wins. If $A_0(x_0,y_0)$ dose not hold, Player 2 wins. Hence, the formula $\exists_{x^X}\forall_{y^Y}A_0(x,y)$ holds if Player 1 has a winning strategy and it does not hold if Player 2 has a winning strategy. Of course, only in a classical setting one can prove that one player has a winning strategy. For more game theoretic interpretations of the functional interpretation we refer to [128].

Example 2.2.18. We continue Example 2.2.14: For an arbitrary formula A with Gödel translation $A^D = \exists_{x^X}\forall_{y^Y}|A|^x_y$, the Gödel translation of $\neg\neg A$ is given by

$$\exists_\varepsilon\forall_p\neg\neg\left(|A|^{\varepsilon p}_{p(\varepsilon p)}\right)$$

with types $\varepsilon : (X\to Y)\to X$ and $p : X\to Y$.
Using the visualisation from Remark 2.2.17 we can compare the Gödel translation of A with the Gödel translation of $\neg\neg A$ as follows: In the game of A^D, Player 2 picks a y_0 depending on the chosen x_0 by Player 1. The strategy of Player 2 is represented by the function $p : X\to Y$. Hence, in the game of $(\neg\neg A)^D$ Player 1 can choose an element of X dependent on the strategy of Player 2, i.e. Player 1 can use the information how Player 2 wants to find a counterexample. Therefore, the game of $(\neg\neg A)^D$ is easier to win for Player 1 as the game of A^D.

Let $A := \exists_x\forall_y(Py\to Px)$ be a version of the drinker paradox for a decidable predicate P. Here A is already in Σ_2 form and therefore the Gödel interpretation of A is again A. The classical way to give a witness of the existence would be

$$x := \begin{cases} y & \text{for a } y \text{ with } Py \\ 0 & \text{if such a } y \text{ does not exists} \end{cases}$$

But in intuitionistic logic this is not well-defined and A is not provable. However, the double negation $\neg\neg A$ is provable and the Gödel translation is

$$\exists_\varepsilon\forall_p(P(p(\varepsilon p))\to P(\varepsilon p)).$$

Here it is able to give a witness: We define $\varepsilon : (X\to Y)\to X$ by

$$\varepsilon(p) := \begin{cases} p0 & \text{if } P(p0) \\ 0 & \text{if } \neg P(p0). \end{cases}$$

Since P is decidable, ε is well-defined and fulfils $\forall_p(P(p(\varepsilon p))\to P(\varepsilon p))$.

Motivation 2.2.19. We first show that the Gödel interpretation of a formula A is not so far away from the original formula. Meaning that if we assume some constructively critical principles, A and A^D are equivalent. The principles are the following:

Definition 2.2.20. The *Markov principle* (MP) is the formula

$$(\forall_x P(x) \to A_0) \to \exists_x (P(x) \to A_0).$$

for each decidable predicate $P(x)$ and each decidable formula A_0.

The *axiom of choice*[1] (AC) is given by

$$\forall_x \exists_y A(x,y) \to \exists_{f^{X\to Y}} \forall_x A(x, fx)$$

for each arbitrary formula $A(x^X, y^Y)$.

The *axiom of the independence of the premise* (IP) is given by the formula

$$(A \to \exists_x B) \to \exists_x (A \to B),$$

where A and B are arbitrary formulas with $x \notin FV(A)$ and $\tau^+(A) = \mathbb{U}$.

Remark 2.2.21. The Markov principle could also be formulated as

$$\neg\forall_x P(x) \to \exists_x \neg P(x).$$

In words this formula says that if Px does not hold for all x then there is a counter example. In intuitionistic logic both formulations are equivalent, but we will use the more general one.

Lemma 2.2.22. For a formula A in $\mathbf{HA}^\omega$ we have

$$\mathrm{MP} + \mathrm{AC} + \mathrm{IP} \vdash A \leftrightarrow \exists_x \forall_y |A|_y^x.$$

Proof. The proof is done by induction:
If A is an atomic formula, it is its own Dialectica interpretation and we have nothing to show.
Let $A = B \wedge C$. By induction hypothesis this is equivalent to

$$\exists_x \forall_y |B|_y^x \wedge \exists_u \forall_v |C|_v^u.$$

[1] Despite the fact that this axiom is called "axiom of choice", it is not as inconstructive as the axiom of choice in set theory.

This is without any assumption equivalent to

$$\exists_{x,u}\forall_{y,v}(|B|_y^x \wedge |C|_v^u),$$

which is equivalent to A^D by definition.
Let $A = B \vee C$. By induction hypothesis we have the equivalent formula

$$\exists_x\forall_y|B|_y^x \vee \exists_u\forall_v|C|_v^u.$$

For an element $b \in \mathbb{B}$ we have $b = 0 \vee b = 1$, therefore we have the equivalent formula

$$\exists_b.\left(b = 0 \to \exists_x\forall_y|B|_y^x\right) \wedge (b = 1 \to \exists_u\forall_v|C|_v^u)$$

By using IP, we get

$$\exists_b.\exists_x\left(b = 0 \to \forall_y|B|_y^x\right) \wedge \exists_u\left(b = 1 \to \forall_v|C|_v^u\right).$$

This is with out any assumptions equivalent to

$$\exists_{b,x,u}\forall_{y,v}.\left(b = 0 \to |B|_y^x\right) \wedge \left(b = 1 \to |C|_v^u\right),$$

which is A^D by definition.
Let $A = B \to C$. The following formulas are equivalent:

$$\begin{aligned}
&\exists_x\forall_y|B|_y^x \to \exists_u\forall_v|C|_v^u \text{ by induction hypothesis}\\
&\forall_x.\forall_y|B|_y^x \to \exists_u\forall_v|C|_v^u\\
&\forall_x\exists_u.\forall_y|B|_y^x \to \forall_v|C|_v^u \text{ by IP}\\
&\forall_x\exists_u\forall_v.\forall_y|B|_y^x \to |C|_v^u\\
&\forall_x\exists_u\forall_v\exists_y.|B|_y^x \to |C|_v^u \text{ by MP}\\
&\exists_f\forall_{x,v}\exists_y.|B|_y^x \to |C|_v^{fx} \text{ by AC}\\
&\exists_{f,g}\forall_{x,v}.|B|_{gvx}^x \to |C|_v^{fx} \text{ by AC}
\end{aligned}$$

The last formula is equivalent to $\exists_{f,g}\forall_{x,v}|B \to C|_{x,v}^{f,g}$ by definition.
The case $A = \exists_x B$ is trivial.
Let $A = \forall_x B(x)$. By induction hypothesis we have that A is equivalent to

$$\forall_x\exists_y\forall_z|B(x)|_z^y$$

and by using the axiom of choice, we have

$$\exists_f\forall_{x,z}|B(x)|_z^{fx}.$$

By definition this is $\exists_f\forall_{x,z}|\forall_x B(x)|_{x,z}^f$. □

Theorem 2.2.23 (Soundness of the functional interpretation). Assume we have a derivation M of a formula A in $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP} + \mathrm{MP}$ with assumptions $(u_i : C_i)_{i<n}$. Let $x_i : \tau^+(C_i)$ for each i and $y : \tau^-(A)$ be variables. Then there are terms $et^+(M) =: t$ of type $\tau^+(A)$ and $et_i^-(M) =: r_i$ of type $\tau^-(C_i)$ with $y \notin FV(t)$, such that $|A|_y^t$ is derivable in $\mathbf{HA}^\omega$ with assumptions $\overline{u}_i : |C_i|_{r_i}^{x_i}$.

Proof. The proof is done by induction on M. We refer to [159, Section 7.4.4]. □

2.2.3 Application: metastability

Motivation 2.2.24. In this section, we give an example where the Dialectica interpretation together with the double-negation translation is applied to a general version of the statement that each non-negative and non-increasing sequence of real numbers converges. This statement cannot be proven constructively, but the translated formula can. Our example is presented in [95, Proposition 6.4] in a more general setting. Hence, this section is not new but it is a foretaste to the next chapter, where we present metastability in the context of algebra.

At the end we want to prove the translated formula of being a Cauchy sequence. A sequence $(\lambda_n)_{n\in\mathbb{N}}$ of real numbers is a Cauchy sequence if

$$\forall_k \exists_n \forall_{i,j\geq n} |\lambda_i - \lambda_j| \leq 2^{-k}.$$

Before we apply the Gödel-Gentzen translation followed by the functional interpretation, we reformulate this statement. This is a standard method: the functional interpretation with double-negation shift is applied to an equivalent version of the starting formula. In our case the following formula is appropriate:

$$\forall_k \exists_n \forall_l \forall_{i,j\in[n,n+l]} |\lambda_i - \lambda_j| \leq 2^{-k}$$

The new formula may look more complex but the part $\forall_{i,j\in[n,n+l]} |\lambda_i - \lambda_j| \leq 2^{-k}$ of the formula above can be seen as a decidable predicate (if we assume that the inequality is decidable) as the quantifier is bounded.

The translation of this formula is now given by

$$\exists_\Psi \forall_{k,p} \forall_{i,j\in[\Psi(p,k),\Psi(p,k)+p(\Psi(p,k))]} |\lambda_i - \lambda_j| \leq 2^k,$$

where the types are $\Psi : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}$, $p : \mathbb{N} \to \mathbb{N}$ and $k : \mathbb{N}$. In the following we show an equivalent version of this statement under certain considerations. Note that we do the proof by hand and do not use theorems like Corollary 2.2.10 and Theorem 2.2.23 since it is easier to formulate a new proof instead of going through the processes in these theorems.

Lemma 2.2.25. Let $(\lambda_n)_n, (\kappa_n)_n, (\gamma_n)_n \in (\mathbb{R}_0^+)^{\mathbb{N}}$ with

$$\lambda_{n+1} \leq (1+\kappa_n)\lambda_n + \gamma_n \tag{2.1}$$

for all $n \in \mathbb{N}$. Furthermore, let $K_1, K_2 \in \mathbb{N}$ be given with $\sum_{i=1}^{\infty} \kappa_i \leq K_1$ and $\sum_{i=1}^{\infty} \gamma_i < K_2$ and we have two moduli of convergence $r_1, r_2 : \mathbb{R}^+ \to \mathbb{N}$ such that $\forall_{\varepsilon>0} \sum_{i=r_1(\varepsilon)}^{\infty} \kappa_i \leq \varepsilon$ and $\forall_{\varepsilon>0} \sum_{i=r_2(\varepsilon)}^{\infty} \gamma_i \leq \varepsilon$.

Then there is $\Phi : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ such that

$$\forall_k \forall_g \exists_{n \leq \Phi(g,k)} |\lambda_{n+g(n)} - \lambda_n| < 2^{-k}$$

Proof. By iterating the given inequality, we have

$$\lambda_{n+1} \leq e^{\sum_{i=1}^{\infty} \kappa_i} \left(\lambda_1 + \sum_{i=1}^{\infty} \gamma_i \right) \leq e^{K_1}(\lambda_1 + K_2) < \infty,$$

and hence there is $L \in \mathbb{N}$ such that $e^{\sum_{i=1}^{\infty} \kappa_i}(\lambda_1 + \sum_{i=1}^{\infty} \gamma_i) \leq 2^L$. For all $g : \mathbb{N} \to \mathbb{N}$ we define $\tilde{g} : \mathbb{N} \to \mathbb{N}$ by $\tilde{g}(n) = g(n) + n$ and show

$$\forall_M \forall_k \forall_g \exists_{i \leq 2^{k+L}} (\lambda_{\tilde{g}^i(M)} - \lambda_{\tilde{g}^{i+1}(M)} < 2^{-k}) \tag{2.2}$$

Assume that for some M, k and g

$$\forall_{i \leq 2^{k+L}} \lambda_{\tilde{g}^i(M)} - \lambda_{\tilde{g}^{i+1}(M)} \geq 2^{-k}.$$

Summation over $0 \leq i \leq 2^{k+L}$ leads to

$$\lambda_M - \lambda_{\tilde{g}^{2^{k+1}}(M)} \geq (2^{k+L}+1)2^{-k} > 2^L$$

which is a contradiction to the choice of L, therefore (2.2).

Using induction on (2.1) we get

$$\lambda_{n+m} \leq \prod_{j=m}^{n+m-1} (1+\kappa_j) \left(\lambda_m + \sum_{j=m}^{n+m-1} \gamma_j \right) \leq e^{\sum_{j=m}^{n+m-1} \kappa_j} \left(\lambda_m + \sum_{j=m}^{n+m-1} \gamma_j \right),$$

and hence

$$\lambda_{n+m} - \lambda_m \leq \left(e^{\sum_{j=m}^{\infty} \kappa_j} - 1 \right) 2^L + e^{\sum_{j=m}^{\infty} \kappa_j} \sum_{j=m}^{\infty} \gamma_j$$

for all $m, n \in \mathbb{N}$. We define $R : \mathbb{N} \to \mathbb{N}$ by

$$R(k) := \max\{r_1(\ln(2)), r_1(\ln(1 + 2^{-k-L-1})), r_2(2^{-k-2})\},$$

where L is defined as above. Then for all k, $i > R(k)$ and $j > i$ we have

$$\begin{aligned} \lambda_j - \lambda_i &\leq \left(e^{\sum_{j=m}^{\infty} \kappa_j} - 1\right) 2^L + e^{\sum_{j=m}^{\infty} \kappa_j} \sum_{j=m}^{\infty} \gamma_j \\ &\leq 2^{-k-L-1} 2^L + 2 \cdot 2^{-k-2} \leq 2^{-k}. \end{aligned} \tag{2.3}$$

Then (2.2) and the property of R together with $\forall_m \tilde{g}^i(m) \geq m$ gives

$$\forall_k \forall_g \exists_{i \leq 2^k} |\lambda_{\tilde{g}^i(R(k))} - \lambda_{\tilde{g}^i(R(k)) + g(\tilde{g}^i(R(k)))}| < 2^{-k}.$$

Hence, we define

$$\Phi(g, k) := \tilde{g}^{2^k}(R(k)).$$

□

Theorem 2.2.26. In the situation of Lemma 2.2.25 we furthermore have

$$\forall_k \forall_g \exists_{n \leq \Phi(g,k+2)} \forall_{i,j \in [n, n+g(n)]} |\lambda_i - \lambda_j| \leq 2^{-k}.$$

Proof. Let $k \in \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ be given. By Lemma 2.2.25 there is $n \leq \Phi(g, k+2)$ with

$$|\lambda_{n+g(n)} - \lambda_n| \leq 2^{-k-2}$$

and, similar to (2.3) in the proof of this lemma,

$$\forall_{i>n} \forall_{j>i} \lambda_j - \lambda_i \leq 2^{-k-2}.$$

For given $i, j \in [n, n + g(n)]$ we assume w.l.o.g. $i \leq j$ and have $\lambda_j - \lambda_i \leq 2^{-k-2}$ and

$$\lambda_i - \lambda_j < \lambda_i - \lambda_j + \lambda_{n+g(n)} - \lambda_n + 2^{-k-2} = \lambda_i - \lambda_n + \lambda_{n+g(n)} - \lambda_j + 2^{-k-2} < 2^{-k},$$

hence $|\lambda_i - \lambda_j| \leq 2^{-k}$. □

Remark 2.2.27. The notion "metastability" is used since the sequence above is stable on the interval $[n, n+g(n)]$. One can also say that this is an approximation to the property of being a Cauchy sequence. In Section 3.3 we approximate maximal objects in algebra by using a similar process. As here we have an example about metastability in analysis, Section 3.3 can be seen as an example of metastability in algebra.

Chapter 3

Ideal objects in commutative algebra

This chapter is based on [136, 137] which is a result of a collaboration with Thomas Powell and Peter Schuster. The content of both papers are combined. The case studies in Section 3.7 and Section 3.8 are new and were not considered the two papers.

Motivation 3.0.1. The existence of ideal objects, such as maximal ideals in non-zero rings, plays a crucial role in commutative algebra. These are typically justified by Zorn's lemma, and thus pose a challenge from a computational point of view. Giving a constructive meaning to ideal objects is a problem which dates back to Hilbert's program, and today it is still a central theme in the area of dynamical algebra, which focuses on the elimination of ideal objects via syntactical methods. In this chapter, we take an alternative approach based on Gödel's interpretation with double negation shift and sequential algorithms. First we give a computational interpretation of an abstract maximality principle in the countable setting via an intuitive, state based algorithm.

As a concrete application, we present the computational interpretation to a general and abstract formulation of the so-called universal Krull-Lindebaum lemma as in [144]. We give a number of concrete examples of this phenomenon, including the prime ideal theorem and Kronecker's theorem for valuation rings.

The novelty of our work lies in our use of Gödel's functional interpretation and in our description of its solution as a state based algorithm, inspired by recent work of Thomas Powell [128, 129, 133–135], which focuses on the algorithmic meaning of extracted programs. This form of presentation allows us to bridge the gap between the rigorous extraction of programs from proof as terms in some formal calculus,

and the more algorithmic style of dynamical algebra.

It also enables us to present our results in an entirely self-contained manner, without needing to introduce any heavy proof theoretic machinery. Though behind the scenes at least aspects of our work are influenced by Gödel's functional intepretation, given in Section 2.2.2, and Spector's bar recursion [166]. However, neither of these are necessary to understand the theory presented in this chapter.

3.1 General maximality

Motivation 3.1.1. We use a classical metatheory such as $\mathbf{PA}^\omega$ and we globally assume that for each set S and object x in our algorithms we can decide weather $x \in S$ or $x \notin S$ and that the equality is decidable, i.e. $x = y$ or $x \neq y$ for all x, y. Hence, the power set of a given set X can be seen as the set of all boolean valued functions on X, i.e. $\mathcal{P}(X)$ is identified with 2^X. The underlying set in the algorithms of this chapter will always be countable, and therefore this assumption is not a huge restriction.

We begin by presenting our abstract maximality principle, which forms the main subject of this chapter. This is achieved by abstracting ideals of commutative rings to the context of coverings, and in Section 3.4 we abstract prime ideals by adding binary operators. This is a move in the vein of formal topology [147, 150]. The notion of a covering goes back to Tarski's concept of consequent operator [168] and also the axiom systems of Hertz [80].

Notation 3.1.2. Let X be an arbitrary set. We denote the set of all finite subsets of X by $\mathcal{P}_{fin}(X)$. Variables in $\mathcal{P}_{fin}(X)$ will be typically denoted by A and B. A subset $\rhd \subseteq \mathcal{P}_{fin}(X) \times X$ is called a *covering* on X. In the following a covering will typically be denoted by $\rhd$ and treated as a binary relation with infix notation.

We say that the element x is generated by a finite set A if $A \rhd x$ and extend $\rhd$ to arbitrary $S \subseteq X$ by defining $S \rhd x$ whenever there exists some finite $A \subseteq S$ such that $A \rhd x$.

If it is clear from the context, we use the notations A, x and A, B for $A \cup \{x\}$ and $A \cup B$, respectively.

Definition 3.1.3. A covering $\rhd$ on a set X is called *reflexive* if

$$\{a\} \rhd a$$

for all $a \in X$, and it is called *transitive*, if

$$B \rhd b \wedge A, b \rhd a \Rightarrow A, B \rhd a$$

for all $a, b \in X$ and all finite $A, B \subseteq X$.

Remark 3.1.4. For the proof that Algorithm 3.3.6 indeed produces an approximate maximal object, we do not need reflexivity and transitivity of $\rhd$. But after this we will specialize this algorithm to an algorithm for the universal Krull-Lindenbaum lemma. At that point we need reflexivity and transitivity.

Definition 3.1.5. Let $\rhd$ be a covering of a set X and $S \subseteq X$ a subset. We define the sequence $(S_i)_{i\in\mathbb{N}}$ by

$$S_0 := S \qquad \text{and} \qquad S_{i+1} := \left\{ x \in X \;\middle|\; \bigcup_{j=0}^{i} S_j \rhd x \right\}.$$

With this sequence we define $\langle S\rangle^{\rhd} := \bigcup_{n\in\mathbb{N}} S_i$, the *closure* of S w.r.t. $\rhd$. A subset $I \subseteq X$ with $I = \langle I\rangle^{\rhd}$ is called a $\rhd$-*ideal*.

Remark 3.1.6. The set $\langle S\rangle^{\rhd}$ in the definition above is an $\rhd$-ideal: one easily sees that $\langle S\rangle^{\rhd} \rhd x$ implies $x \in \langle S\rangle^{\rhd}$.

Lemma 3.1.7. If a covering $\rhd$ on a set X is reflexive and transitive then $\langle S\rangle^{\rhd} = \{x \in X \mid S \rhd x\}$ for all $S \subseteq X$.

Proof. By induction and transitivity we have $S_i = S_1$ for all $i \geq 1$, and we have $S_0 \subseteq S_1$ by reflexivity. □

Notation 3.1.8. For a given covering $\rhd$ on a set X, $S \subseteq X$ and $x \in X$, we denote the *closed extension* of S with x by $S \oplus x := \langle S, x\rangle^{\rhd}$.

If furthermore θ is a predicate on X, we write $\theta^{\rhd}(S)$ for $\forall_{x\in\langle S\rangle^{\rhd}}\theta(X)$ and $\theta^{\rhd}_F(S) := \theta^{\rhd}(F \cup S)$.

Motivation 3.1.9. In the following definition we define general maximality. We will use this definition as starting point for our studies about the approximation of maximal objects.

Definition 3.1.10. Let $\rhd$ be a covering and θ a predicate on a set X. We say that $M \subseteq X$ is *maximal* w.r.t. $\rhd$ and θ if

(i) M is closed w.r.t. $\rhd$ (i.e. $M \rhd x \Rightarrow x \in M$),

(ii) $\theta^{\rhd}(M)$,

(iii) $\neg\theta^{\rhd}(M, x)$ for any $x \notin M$.

Motivation 3.1.11. The following theorem shows the existence of such a maximal object by using Zorn's Lemma. Therefore, from the proof of this theorem one cannot construct this maximal object. However, in the next section we analyse the notion of maximality, use some tools to reformulate them and construct an algorithm which provides an approximation to a maximal object.

Theorem 3.1.12. Let a covering $\triangleright$ and a predicate θ on a set X be given. Suppose that $\theta^{\triangleright}(F)$ holds for some $F \subseteq X$. Then there exists some $M \supseteq F$ which is maximal w.r.t. $\triangleright$ and θ.

Proof. We define $\mathcal{S} := \{S \subseteq X \mid F \subseteq S = \langle S \rangle^{\triangleright} \wedge \theta^{\triangleright}(S)\}$. $\langle F \rangle^{\triangleright} \in \mathcal{S}$ follows from $\theta^{\triangleright}(F)$, and we show that $\mathcal{S}$ is chain complete:

Let $\varnothing \neq \mathcal{K} \subseteq \mathcal{S}$ be a chain w.r.t. the order $\subseteq$. We define $S := \bigcup \mathcal{K}$. Then if $A \triangleright x$ for finite $A \subseteq S$ and $x \in X$, we get $P \in \mathcal{K}$ with $A \subseteq P$ because $\mathcal{K}$ is a chain. Since $P = \langle P \rangle^{\triangleright}$, we have $x \in P$ and therefore $x \in S$. Furthermore, we have $\theta^{\triangleright}(S)$ because if $x \in S = \langle S \rangle^{\triangleright}$ then $x \in P$ for some $P \in \mathcal{K}$ and hence $\theta(x)$.

By Zorn's lemma $\mathcal{S}$ has a maximal element M. M is closed and $\theta^{\triangleright}(M)$ holds as $M \in \mathcal{S}$. Given $x \in X \setminus M$, we have $M \subsetneq M \oplus x$ and thus $M \oplus x \notin \mathcal{S}$. But since $M \oplus x$ is closed, $\neg\theta^{\triangleright}(M, x)$ □

3.2 A logical analysis

Notation 3.2.1. In the following we often assume that X is countable. In that case we say "Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable.", which means that there is a set X and a sequence $(x_n)_{n\in\mathbb{N}}$ with values in X such that $X = \{x_n \mid n \in \mathbb{N}\}$.

For $S \subseteq X$, the *initial segment* of S of length n is defined by $[S](n) := \{x_m \mid n < m\} \cap S$ and we further define $\mathrm{dom}(S) := \{n \in \mathbb{N} \mid x_n \in S\}$.

Motivation 3.2.2. The following theorem gives a criterion when a subset $M \subseteq X$ is maximal w.r.t. $\triangleright$ and θ. This criterion is central for our investigations. We adapt a well-known trick from reverse mathematics, see e.g. [165, Lemma III.5.4]. After some technical assumptions about the formulas, we will use this theorem as motivation to define explicit maximal objects. It turns out that from a logical point of view the defining formula of an explicit maximal object is more manageable than the defining formula of a general maximal object. In particular, we can use the double negation shift and Gödel's functional interpretation to approximate explicit maximal objects.

Theorem 3.2.3. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable and $F \subseteq X$ a subset. Suppose $\theta^{\rhd}(F)$ and that $M \subseteq X$ satisfies

$$x_n \in M \Leftrightarrow \theta_F^{\rhd}([M](n), x_n) \tag{3.1}$$

for all $n \in \mathbb{N}$. Then M is a maximal ideal w.r.t. $\rhd$ and θ with $F \subseteq M$.

Proof. We abbreviate $M_n := [M](n)$ and $F_n := [F](n)$. First we show by induction that $\theta_F^{\rhd}(M_n)$ and $F_n \subseteq M_n$ for all $n \in \mathbb{N}$:

For $n = 0$ we have $M_0 = \varnothing = F_0$ and the claims follow by the assumption $\theta^{\rhd}(F)$, i.e. $\theta_F^{\rhd}(\varnothing)$. Now we assume that $\theta_F^{\rhd}(M_n)$ and $F_n \subseteq M_n$ hold for a given $n \in \mathbb{N}$. We have two cases: if $\theta_F^{\rhd}(M_n, x_n)$ holds then $x_n \in M$ and hence $M_{n+1} = M_n \cup \{x_n\}$. Furthermore, we have $F_{n+1} \subseteq F_n \cup \{x_n\} \subseteq M_{n+1}$. On the other hand, if $\neg\theta_F^{\rhd}(M_n, x_n)$ holds, then $x_n \notin M$ and hence $M_{n+1} = M_n$. Therefore, $\theta_F^{\rhd}(M_{n+1})$ follows by the induction hypothesis. To see $F_{n+1} \subseteq M_{n+1} = M_n$, we need to show $x_n \notin F$. Assume $x_n \in F$, then $\theta_F^{\rhd}(M_n, x_n) \Leftrightarrow \theta_F^{\rhd}(M_n)$ and since the latter is true, this would contradict the assumption $\neg\theta_F^{\rhd}(M_n, x_n)$.

Now we prove the three properties of maximality. First we show that M is closed, and therefore we assume that $M \rhd\!\!\rhd x_n$ but $x_n \notin M$ for some given $n \in \mathbb{N}$. Then by definition $\neg\theta_F^{\rhd}(M_n, x_n)$. Since $M \rhd\!\!\rhd x_n$, we have $M_k \rhd\!\!\rhd x_n$ for some $k \in \mathbb{N}$. First assume $k \leq n$. Then $M_k \subseteq M_n$ and thus $M_n \rhd\!\!\rhd x_n$, which implies $x_n \in \langle M_n \rangle^{\rhd}$ and therefore

$$M_n \oplus x_n = \langle M_n \rangle^{\rhd}.$$

But then $\theta_F^{\rhd}(M_n)$ contradicts $\neg\theta_F^{\rhd}(M_n, x_n)$. Now assume $n < k$. Then $M_n \oplus x_n \subseteq M_k \oplus x_n$ and thus $\neg\theta_F^{\rhd}(M_n, x_n)$ implies $\neg\theta_F^{\rhd}(M_k, x_n)$. Because of $M_k \rhd\!\!\rhd x_n$ we have $M_k \oplus x_n = \langle M_k \rangle^{\rhd} \subseteq \langle M_k, F \rangle^{\rhd}$, and therefore $\neg\theta_F^{\rhd}(M_k)$, a contraction to $\theta_F^{\rhd}(M_k)$.

Next we show $\theta^{\rhd}(M)$: Let $x_n \in M = \langle M \rangle^{\rhd}$ be given. Then $x_n \in M_{n+1} \subseteq M$ and thus $\theta(x_n)$ follows from $\theta^{\rhd}(M_{n+1})$.

Finally, we show maximality: If $x_n \notin M$ then $\neg\theta_F^{\rhd}(M_n, x_n)$ and since $M_n \oplus x_n \subseteq M \oplus x_n$ it follows $\neg\theta_F^{\rhd}(M, x_n)$. □

Definition 3.2.4. For a given set X, a covering $\rhd$ on X is called a Σ_1^0-*covering* if for all $A \in \mathcal{P}_{fin}(X)$ and $x \in X$ the formula $A \rhd x$ is a Σ_1^0-formula. In particular, there is a decidable predicate which we will denote by $\rhd_{(.)}$ such that $A \rhd x \Leftrightarrow \exists_t A \rhd_t x$, where t is a variable of a base type. We denote this base type by W and call it the *witness type*. Variables of type W will be denoted by s and t.

Lemma 3.2.5. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. Assume that $\rhd$ is a Σ_1^0-covering and θ is decidable. For $F, S \subseteq X$ and $x \in X$ we have:

1. $x \in \langle S \rangle^{\triangleright}$ is a Σ_1^0-Formula and
2. $\theta^{\triangleright}(S)$ and $\theta_F^{\triangleright}(S)$ are Π_1^0-Formulas.

Proof. For the first statement: we have $x \in \langle A \rangle^{\triangleright}$ iff there exists a finite derivation tree whose root is x, whose leaves are elements of A and whose nodes represent instances of $\triangleright$. Given that $\triangleright$ can be encoded as a Σ_1^0-formula, the existence of a derivation tree can in turn be represented as Σ_1^0-formula via some suitable encoding.

For the second statement: we have $\theta^{\triangleright}(S) \Leftrightarrow \forall_m(x_m \in \langle S \rangle^{\triangleright} \Rightarrow \theta(x_m))$. By the first part $x_m \in S$ is a Σ_1^0-formula. Therefore, $x_m \in \langle S \rangle^{\triangleright} \Leftrightarrow \exists_r G(S, x_m, r)$ for some decidable $G(S, x_m, r)$. It follows:

$$\theta^{\triangleright}(S) \Leftrightarrow \forall_m(\exists_r G(S, x_m, r) \Rightarrow \theta(x_m)) \Leftrightarrow \forall_{m,r}(G(S, x_m, r) \Rightarrow \theta(x_m)) \tag{3.2}$$

Therefore, $\theta^{\triangleright}(S)$ is a Π_1^0-formula. Since $\theta_F^{\triangleright}(S)$ is defined as $\theta^{\triangleright}(F \cup S)$, $\theta_F^{\triangleright}(S)$ is also a Π_1^0-formula. □

Remark 3.2.6. As in [136], Lemma 3.2.5 also works if we assume that $\theta(x)$ is a Π_1^0-formula. But from (3.2) we see that an all-quantified variable in $\theta(x_m)$ can be rewritten as an existence-quantified variable in $S \triangleright x_m$, and hence we do not need that $\theta(x)$ is a Π_1^0-formula.

Notation 3.2.7. By Lemma 3.2.5 $\theta^{\triangleright}(S)$ and $\theta_F^{\triangleright}(S)$ are Π_1^0-formula if $\triangleright$ is a Σ_1^0-covering and θ is decidable. In the following we use the notation $\theta_F^{\triangleright}(A) \Leftrightarrow \forall_p R_A^F(p)$, where $R_A^F(p)$ is a decidable predicate and $R_A(p) := R_A^{\varnothing}(p)$. We denote the base type for which R_A^F is defined by $\mathcal{G}$ and variables of type $\mathcal{G}$ will be denoted by p. Note that this notation ignores that $R_A^F(p)$ also depends on the covering $\triangleright$ and the predicate θ. But we will always denote these objects by $\triangleright$ and θ. Hence there will not be any possibility of confusion.

Remark 3.2.8. In the situation of Lemma 3.2.5, if $\triangleright$ is reflexive and transitive, we have $\langle S \rangle^{\triangleright} = \{x \in X \mid \exists_{A \in \mathcal{P}_{fin}(S)} A \triangleright x\}$ by Lemma 3.1.7. Therefore

$$x \in \langle S \rangle^{\triangleright} \Leftrightarrow \exists_{A,t}(A \in \mathcal{P}_{fin}(S) \wedge A \triangleright_t x)$$

is a representation of $x \in \langle S \rangle^{\triangleright}$ as Σ_1^0-formula and

$$\theta^{\triangleright}(S) \Leftrightarrow \forall_{A,t,x}(A \in \mathcal{P}_{fin}(S) \wedge A \triangleright_t x \Rightarrow \theta(x))$$

is a representation of $\theta^{\triangleright}(S)$ as Π_1^0-formula. Note that since there is a surjection from $\mathbb{N}$ to X, there is also a surjection from $\mathbb{N}$ to $\mathcal{P}_{fin}(X)$ and therefore $\mathcal{P}_{fin}(X)$ can be

seen as base type. In particular, in this case one can choose $\mathcal{G} = \mathcal{P}_{fin}(X) \times W \times X$ and

$$R^F_S(A, t, x) \Leftrightarrow (F \cup S \supseteq A \rhd_t x \Rightarrow \theta(x)).$$

We will use this later to develop an algorithm for the universal Krull-Lindenbaum lemma.

Corollary 3.2.9. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. Given a Σ^0_1-covering $\rhd$, a decidable predicate θ on X and $F \subseteq X$, such that $\theta^{\rhd}(F)$ and $M \subseteq X$ satisfies

$$x_n \in M \Leftrightarrow \forall_p R^F_{[M](n)\cup\{x_n\}}(p) \tag{3.3}$$

for all $n \in \mathbb{N}$. Then M is maximal w.r.t. $\rhd$, θ and $F \subseteq M$.

Proof. This follows directly from Theorem 3.2.3 and Notation 3.2.7. □

Motivation 3.2.10. By writing out the equivalence, the existence of some M satisfying (3.3) unfolds to

$$\exists_M \forall_n. \left(x_n \in M \Rightarrow \forall_p R^F_{[M](n)\cup\{x_n\}}(p)\right) \wedge \left(x_n \notin M \Rightarrow \exists_q \neg R^F_{[M](n)\cup\{x_n\}}(q)\right).$$

and in Skolem normal form

$$\exists_{M,f} \forall_{n,p}. \left(x_n \in M \Rightarrow R^F_{[M](n)\cup\{x_n\}}(p)\right) \wedge \left(x_n \notin M \Rightarrow \neg R^F_{[M](n)\cup\{x_n\}}(f(n))\right). \tag{3.4}$$

This motivates our final version of maximality, which is now in a form where we can directly apply Gödel's functional interpretation together with the double-negation shift.

Definition 3.2.11. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. An *explicit maximal object* w.r.t. a Σ^0_1-covering $\rhd$ on X, a decidable predicate θ on X and $F \subseteq X$ is a set $M \subseteq X$ together with a function $f : \mathrm{dom}(X \setminus M) \to \mathcal{G}$ such that

- $x_n \in M \Rightarrow R^F_{[M](n)\cup\{x_n\}}(p)$
- $x_n \notin M \Rightarrow \neg R^F_{[M](n)\cup\{x_n\}}(f(n))$

for all $n \in \mathbb{N}$ and $p \in \mathcal{G}$.

Remark 3.2.12. The idea here is that the function f provides concrete evidence for why x_n is excluded in the maximal structure M. In particular, it encodes a derivation tree T such that the leaves of T are in $[M](n) \oplus x_n$, the node of T are $\rhd$-rules and $\neg\theta(r)$, where r is the root of the tree T.

3.3 An approximating algorithm for maximal objects

Motivation 3.3.1. In general, it is impossible to effectively compute a set M together with an f satisfying Definition 3.2.11. However, we will demonstrate an approximate, or metastable, formulation of maximality in the spirit of Gödel's functional interpretation with double negation shift, which can be directly witnessed via an intuitive stateful procedure.

In Algorithm 3.3.6 we give a sequential algorithm (in the sense of Gurevich [78, Definition 6.1]) whose states evolves step by step until they terminate in some final state which represents the solution of the functional interpretation with double negation shift of Theorem 3.2.3. Each step in this process represents an improvement to our construction of an approximate ideal object, and hence can also be viewed as a learning procedure in the style of [11, 134]. In particular, we replace the maximality principle in the countable setting by a form of update recursion, which has already be studied in the context of open induction [19, 40, 140].

Formula (3.4) is already its Gödel translation. As in the first part of Example 2.2.18 we translate it by using the double negation shift to get an approximation to an explicit maximal object. This can also be seen as a metastable version.

Definition 3.3.2. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. Given a Σ^0_1-covering $\rhd$ and a decidable predicate θ on X, let ω, ϕ be two functionals which take as input $M \in 2^X$ and $f : \mathrm{dom}(X \setminus M) \to \mathcal{G}$ and return an integer and an element in $\mathcal{G}$, respectively. An *approximate explicit maximal object* w.r.t. $\rhd, \theta, \omega$ and ϕ is a set $M \subseteq X$ together with a function f, such that

- $x_n \in M \Rightarrow R^F_{[M](n)\cup\{x_n\}}(p)$
- $x_n \notin M \Rightarrow \neg R^F_{[M](n)\cup\{x_n\}}(f(n))$

for $n \leq \omega(M, f)$ and $p = \phi(M, f)$.

Remark 3.3.3. Note that the definition above is slightly stronger than the functional interpretation with double negation shift, because we require $n \leq \omega(M, f)$ instead of $n = \omega(M, f)$ and $\omega(M, f)$ can also be a negative integer. It turns out that this is more comfortable. For instance, we use that M and f form always an approximate explicit maximal object if $\omega(M, f) < 0$.

Definition 3.3.4. A *state* π is a function of type $\mathbb{N} \to \mathbb{U} + \mathcal{G}$. We denote $\mathrm{in}_0\mathrm{Dummy} \in \mathbb{U} + \mathcal{G}$ by ε, and we identify elements of the form $\mathrm{in}_1 p \in \mathbb{U} + \mathcal{G}$ with $p \in \mathcal{G}$ itself.

Furthermore, if a countable set $X = \{x_n \mid n \in \mathbb{N}\}$ is given then for a state π, we define the set $M[\pi] \subseteq X$ as

$$M[\pi] := \{x_n \in X \mid \pi(n) = \varepsilon\}$$

and the function $f[\pi] : \mathrm{dom}(X \setminus M[\pi]) \to \mathcal{G}$ by

$$f[\pi](n) := \pi(n),$$

where $\pi(n) \in \mathcal{G}$ follows from the assumption that $x_n \in M[\pi]$. For the two functionals ω and ϕ from Definition 3.3.2, we define

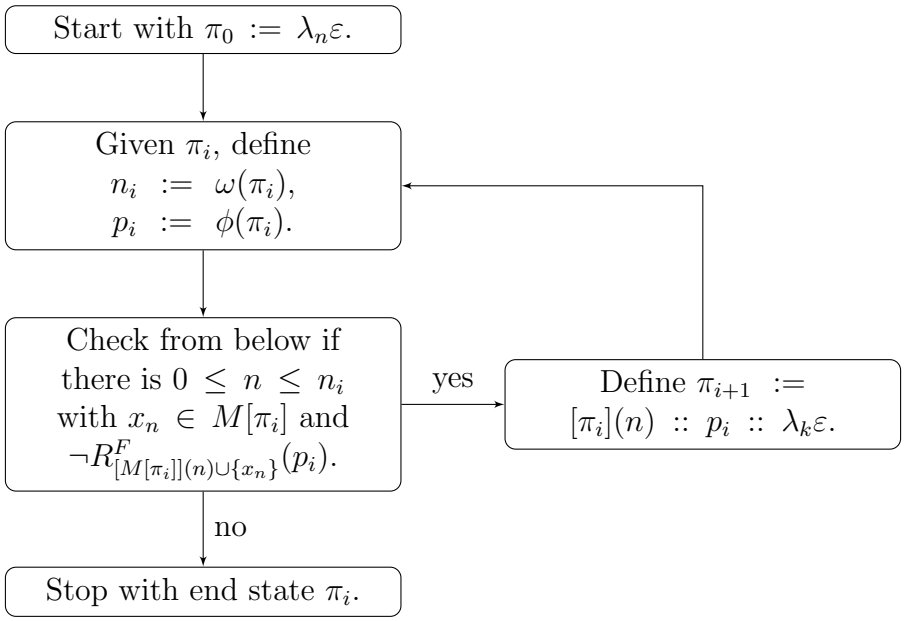
$$\omega(\pi) := \omega(M[\pi], f[\pi]) \qquad \text{and} \qquad \phi(\pi) := \phi(M[\pi], f[\pi]).$$

A state π is called *approximate explicit maximal object* if $(M[\pi], f[\pi])$ is an approximate explicit maximal object.

Notation 3.3.5. For any function f which maps $\mathbb{N}$ into some arbitrary image set (for example if f is a state) and any $n \in \mathbb{N}$ the n-th initial segment $[f](n)$ of f is the list of the first n values of f, namely

$$[f](n) := [f(0), \ldots, f(n-1)].$$

Algorithm 3.3.6. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable and $F \subseteq X$. For $\rhd, \theta, \omega$ and ϕ as in Definition 3.3.2 we define a finite sequence $(\pi_i)_{i \le L}$ for some $L \in \mathbb{N}$ or an infinite sequence $(\pi_i)_{i \in \mathbb{N}}$, of states recursively as follows:



Here :: denotes the list concatenation. In particular,

$$M[\pi_{i+1}] = [M[\pi_i]](n) \cup \{x_k \in X \mid k > n\}.$$

Usually we just write $(\pi_i)_i$ to include the finite and the infinite case.

Lemma 3.3.7. In the situation of Algorithm 3.3.6 let $(\pi_i)_i$ be the output. Then for each state π_i and $n \in \mathbb{N}$ we have

$$x_n \notin M[\pi_i] \Rightarrow \neg R^F_{[M[\pi_i]](n)\cup\{x_n\}}(f[\pi_i](n)).$$

Informally spoken, if an element is excluded from the set of a state then the state provides a concrete evidence why this element is excluded.

Proof. We use induction on i. We have $M[\pi_0] = X$ and therefore, the statement is trivial for $i = 0$. For the induction step suppose that the statement is true for some i. If π_i is an end state, we are done. Otherwise, π_{i+1} exists and we assume $x_n \notin M[\pi_{i+1}]$ for some $n \in \mathbb{N}$. Our goal is $\neg R^F_{[M[\pi_{i+1}]](n)\cup\{x_n\}}(f[\pi_{i+1}](n))$. By Algorithm 3.3.6

$$\pi_{i+1} := [\pi_i](n') :: p_i :: \lambda_k\varepsilon,$$

for some $n' \leq n_i = \omega(\pi_i)$ and $p_i = \phi(\pi_i)$ with

$$x_{n'} \in M[\pi_i] \qquad \text{and} \qquad \neg R^F_{[M[\pi_i]](n')\cup\{x_{n'}\}}(p_i).$$

There are two possibilities: Either $n < n'$ then $x_n \notin M[\pi_i]$ and so the result follows by the induction hypothesis since $f[\pi_{i+1}](n) = \pi_{i+1}(n) = \pi_i(n) = f[\pi_i](n)$ and $[M[\pi_{i+1}]](n) = [M[\pi_i]](n)$, or $n = n'$ and so $f[\pi_{i+1}](n) = p_i = \phi(\pi_i)$ which is defined to satisfy $\neg R^F_{[M[\pi_i]](n)\cup\{x_n\}}(p_i)$, and thus the result follows since $[M[\pi_{i+1}]](n) = [M[\pi_i]](n)$. Note that $n > n'$ is not possible due to $x_n \notin M[\pi_{i+1}]$ □

Theorem 3.3.8. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. Given a Σ^0_1-covering $\rhd$ on X, a decidable predicate θ on X, $F \subseteq X$ and ω, ϕ as in Definition 3.3.2, assume that Algorithm 3.3.6 terminates in an end state π_k for some $k \in \mathbb{N}$. Then π_k is an approximate explicit maximal object w.r.t. $\rhd$, θ, ω and ϕ.

Proof. By Algorithm 3.3.6, π_k is an end state if and only if $x_n \in M[\pi_k]$ implies $R^F_{[M[\pi_k]](n)\cup\{x_n\}}(\phi(\pi_k))$ for all $n \leq n_k = \omega(\pi_k)$. Furthermore, by Lemma 3.3.7 if $x_n \notin M[\pi_k]$ then $\neg R^F_{[M[\pi_k]](n)\cup\{x_n\}}(f[\pi_k](n))$. Therefore, the two properties of an approximate explicit maximal object are fulfilled. □

Motivation 3.3.9. In the following we show that Algorithm 3.3.6 terminates for pleasant input. For this we need an additional assumption, namely that the functionals ω and ϕ are continuous in the following sense:

Definition 3.3.10. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. We say that (ω, ϕ) as in Definition 3.3.2 are continuous if for all states $\pi : \mathbb{N} \to \mathbb{U} + \mathcal{G}$ there exists some $L \in \mathbb{N}$ such that for any other input state π', if $[\pi](L) = [\pi'](L)$ then

$$(\omega(\pi), \phi(\pi)) = (\omega(\pi'), \phi(\pi')).$$

Remark 3.3.11. Note that whenever ω and ϕ are instantiated by computable functionals, they will automatically be continuous. Therefore, continuity is not a restriction for the purposes of our applications.

Theorem 3.3.12. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable, $\rhd$ a Σ^0_1-covering and θ a decidable predicate on X. Whenever Algorithm 3.3.6 is evaluated on continuous input ω and ϕ, it will terminate.

Proof. By contradiction we assume that the algorithm does not terminate and so the algorithm defines an infinite sequence $(\pi_i)_{i\in\mathbb{N}}$. We recursively define an increasing sequence $(m_i)_{i\in\mathbb{N}}$ satisfying

$$\forall_n \forall_{i \geq m_n} [\pi_i](n) = [\pi_{m_n}](n) \tag{3.5}$$

as follows:

For the start, we define $m_0 := 0$. Now, assume that m_n is already defined. We have two cases: either there is some $m \geq m_n$ such that $x_n \notin M[\pi_m]$, in which case we define m_{n+1} as the minimum of $\{m \geq m_n \mid x_n \notin M[\pi_m]\}$, or $x_n \in M[\pi_m]$ for all $m \geq m_n$, in this case we set $m_{n+1} := m_n$.

We prove that this construction satisfies (3.5) using induction on n. The base case is trivial. For the step case let n be fixed. By the induction hypothesis and $m_{n+1} \geq m_n$ we have $[\pi_i](n) = [\pi_{m_{n+1}}](n)$ for all $i \geq m_{n+1}$ and so we only need to check $\pi_i(n) = \pi_{m_{n+1}}(n)$ for all $i \geq m_{n+1}$. Now, in the case $x_n \in M[\pi_i]$ for all $i \geq m_n = m_{n+1}$ we are done because this means that $\pi_{m_{n+1}}(n) = \pi_i(n) = \varepsilon$ for all $i \geq m_{n+1}$. In the other case, if $x_n \notin M[\pi_{m_{n+1}}]$ then $\pi_{m_{n+1}}(n) = p \in \mathcal{G}$. By the structure of the algorithm, the only way that $\pi_i(n) \neq \pi_{m_{n+1}}(n)$ is possible for some $i > m_{n+1}$ is if there is some $m_{n+1} \leq k < i$ with $\pi_{k+1} = [\pi_k](n') :: \phi(\pi_k) :: \lambda_l \varepsilon$ for some $n' < n$. But this contradicts the induction hypothesis, namely that $[\pi_k](n) = [\pi_{k+1}](n)$.

We now define π_∞ by $\pi_\infty(n) := \pi_{m_{n+1}}(n)$ for all $n \in \mathbb{N}$. Since (ω, ϕ) are continuous, there is $L \in \mathbb{N}$ such that for all states π with $[\pi](L) = [\pi_\infty](L)$ it follows

$(\omega(\pi), \phi(\pi)) = (\omega(\pi_\infty)), \phi(\pi_\infty))$. We define $N := \max\{L, \omega(\pi_\infty)+1\}$ and claim that $\pi_m = \pi_{m+1}$ for $m \geq m_N$:

Since $[\pi_\infty](L) = [\pi_m](L)$ it follows

$$\omega(\pi_m) = \omega(\pi_\infty) < N.$$

Since Algorithm 3.3.6 does not terminate, there is some $n \leq \omega(\pi_m)$ with $x_n \in M[\pi_m]$ but $x_n \notin M[\pi_{m+1}]$. But by definition of N and $n < N$ we have $x_n \in M[\pi_m] \Rightarrow x_n \in M[\pi_{m+1}]$, a contradiction. □

3.4 The universal Krull-Lindenbaum lemma

Motivation 3.4.1. The following section is based on the work in [144]. We show an application of the developed algorithm: The universal Krull-Lindenbaum lemma is used in several parts of mathematics. The most known form of it is that the intersection of all prime ideals in a commutative ring is the nilradical. The universal Krull-Lindenbaum lemma is also considered in [145, 151, 153, 154] and a historical view to it can be found in [126].

First, we will generalize the notion of a prime ideal to formalize the universal Krull-Lindenbaum lemma. Then, we will develop a computational formulation and use Algorithm 3.3.6 to get a realiser of this computational formulation. Finally, we will consider the instantiations of the universal Krull-Lindenbaum lemma given in Section 4 of [144] from the computational point of view.

Definition 3.4.2. Let a set X and a covering $\rhd$ on X be given. A *multiplication operator* (or just *operator*) w.r.t. the covering $\rhd$ is a binary function $\circ : X \times X \to X$ which satisfies the following condition, which is called *encoding*:

$$A, a \rhd c \text{ and } B, b \rhd c \text{ implies } A \cup B, a \circ b \rhd c$$

for all $a, b, c \in X$ and $A, B \in \mathcal{P}_{fin}(X)$.

We say that an $\rhd$-ideal $I \subseteq X$ is *prime* (or a *prime $\rhd$-ideal*)[1] if it satisfies

$$a \circ b \in I \Rightarrow a \in I \vee b \in I$$

for all $a, b \in X$.

[1] In contrast to the ordinary definition of prime ideals in commutative rings, with our definition the whole set X is a prime ideal. Therefore, the intersection in the Krull-Lindenbaum lemma is always well-defined.

Theorem 3.4.3 (Universal Krull-Lindenbaum lemma). Let X be a set, $F \subseteq X$ and assume that reflexivity, transitivity and encoding for a covering $\rhd$ and an operator $\circ$ on X are fulfilled. Then

$$\bigcap\{P \subseteq X \mid F \subseteq P \text{ and } P \text{ is a prime } \rhd\text{-ideal } \} \subseteq \langle F\rangle^{\rhd}$$

Proof. We show that for any $r \notin \langle F\rangle^{\rhd}$, there is a prime $\rhd$-ideal $P \subseteq X$ with $F \subseteq P$ and $r \notin P$. Therefore, we fix $r \in X \setminus \langle F\rangle^{\rhd}$ and apply Theorem 3.1.12 with $\theta(x) := (x \neq r)$. Then there exists some ideal $M \supseteq F$ with $r \notin M$ but $r \in \langle M, a\rangle^{\rhd}$ for all $a \in X \setminus M$. We show that this M is prime:

Suppose $a, b \notin M$ and so $r \in \langle M, a\rangle^{\rhd} \cap \langle M, b\rangle^{\rhd}$. Since $\rhd$ is reflexive and transitive, we apply Lemma 3.1.7 and get $A' \rhd r$ and $B' \rhd r$ for some finite subsets $A' \subseteq M, a$ and $B' \subseteq M, b$. As $C \rhd r$ cannot be true for all finte $C \subseteq M$, we get $A, B \subseteq M$ with $A, a = A'$ and $B, b = B'$, i.e. $A, a \rhd r$ and $B, b \rhd r$. By encoding it follows $A \cup B, a \circ b \rhd r$, and hence $r \in \langle M, a \circ b\rangle^{\rhd}$. Therefore, $a \circ b \notin M$ and so M is prime. □

Example 3.4.4. The best known version of the universal Krull-Lindenbaum lemma is the following:

> The intersection of all prime ideals P in a commutative ring R containing a fixed subset $F \subseteq R$ is the radical ideal $\sqrt{(F)}$ generated by the set F.

Here X is instantiated as the commutative ring R, $\rhd$ is given by

$$\{a_1, \ldots, a_k\} \rhd x \Leftrightarrow \exists_{x_1,\ldots,x_k \in R, e \in \mathbb{N}}(a_1x_1 + \cdots + a_kx_k = x^e)$$

and $\circ$ is the ring multiplication. In Section 3.5 we will give a precise definition of $A \rhd x$ and prove that reflexivity, transitivity and encoding are fulfilled.

3.4.1 A computational formulation of the universal Krull-Lindenbaum lemma

Motivation 3.4.5. For some fixed $r \in X$ and $F \subseteq X$, the assumptions of the universal Krull-Lindenbaum lemma are

- $\rhd$ is a covering and $\circ$ is an operator on X such that reflexivity, transitivity and encoding are fulfilled.
- r is an element of each prime ideal P with $F \subseteq P$.

The conclusion of the universal Krull-Lindenbaum lemma is $r \in \langle F \rangle^{\triangleright}$. Assuming that $\triangleright$ is a Σ^0_1-covering this says that there is a finite $A \subseteq F$ and $t \in W$ such that $A \triangleright_t r$.

The first assumption is that reflexivity, transitivity and encoding are fulfilled. The conclusion of each of these properties has the form $A \triangleright x$. Hence, we need three functions which give a witness t with $A \triangleright_t x$. We will call these three functions a "cover structure".

The second assumption has the following equivalent formulation:

$$\forall_{P \subseteq X}. F \not\subseteq P \vee r \in P \vee (P \text{ is not an } \triangleright\text{-ideal}) \vee (P \text{ is not prime}).$$

A computational interpretation of this formula is a functional ψ which takes a subset $P \subseteq X$ as input and returns as output evidence that at least one of the disjuncts holds. Further the four cases in the disjunction again need witnesses. In the first case, we need some element $a \in F$ with $a \notin P$. The second case does not need any witness since it just says $r \in P$, which is decidable. The third case needs as witness some finite $A \subseteq P$, $t \in W$ and $x \in X$ such that $A \triangleright_t x$ but $x \notin P$. The fourth case needs some $a, b, c \in X$ with $a \circ b = c \in P$ but $a, b \notin P$. Putting things together, our functional $\psi(P)$ returns two things: a marker which informs which case holds, and the corresponding evidence for this. We call such a function a "Krull functional".

In the following two definitions we formalize the discussed concepts of a cover structure and a Krull functional.

Definition 3.4.6 (Cover structure). Let X be a set, $\triangleright$ a Σ^0_1-covering and $\circ$ an operator on X. A *cover structure* for $\triangleright$ and $\circ$ is a triple of functions (ι, τ, η) with values in the witness type W satisfying the following properties:

- $\forall_{x \in X}(\{x\} \triangleright_{\iota(x)} x)$
- $\forall_{A,B \in \mathcal{P}_{fin}(X)} \forall_{x,y \in X} \forall_{s,t \in W} (A \triangleright_s x \wedge B, x \triangleright_t y \Rightarrow A \cup B \triangleright_{\tau(A,B,x,y,s,t)} y)$
- $\forall_{A,B \in \mathcal{P}_{fin}(X)} \forall_{x,y,z \in X} \forall_{s,t \in W} (A, x \triangleright_s z \wedge B, x \triangleright_t z \Rightarrow A \cup B, a \circ b \triangleright_{\eta(A,B,x,y,z,s,t)} z)$

In particular, the existence of a cover structure implies reflexivity, transitivity and encoding for $\triangleright$ and $\circ$.

In concrete case studies, if the functions ι, τ and η only depend on some of their arguments, we simply drop arguments that are not needed.

Definition 3.4.7 (Krull functional). Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. Fixing a Σ^0_1-covering $\triangleright$, an operator $\circ$, $F \subseteq X$ and $r \in X$, a *Krull functional*

$\psi : 2^X \to \{0,1,2,3\} \times \mathbb{N}$ is a function which for any input $P \in 2^X$ satisfies

$$\begin{aligned}
&\psi(P) = (0,x) \Rightarrow x \in F \wedge x \notin P\\
&\psi(P) = (1,0) \Rightarrow r \in P\\
&\psi(P) = (2,[\![A,t,x]\!]) \Rightarrow A \subseteq P \wedge A \rhd_t x \wedge x \notin P\\
&\psi(P) = (3,[\![x,y,z]\!]) \Rightarrow x \circ y = z \in P \wedge x,y \notin P.
\end{aligned}$$

Here $[\![x_1,\dots,x_n]\!] \in \mathbb{N}$ denotes some coding of $x_1,\dots,x_n$ as a single natural number and we implicitly associate elements $x,y,z \in X$ with induces representing their encoding in $\mathbb{N}$. Note that W is a base type and can therefore also be embedded into $\mathbb{N}$.

Motivation 3.4.8. The computational interpretation of the universal Krull-Lindenbaum lemma takes a cover structure and a Krull functional as input. Our aim is to get $A \subseteq F$ and $t \in W$ (where W is the witness type) with $A \rhd_t r$ by using Algorithm 3.3.6. Hence, we will convert the cover structure and the Krull functional to some functionals ω and ϕ, which we then pass to Algorithm 3.3.6. The functional defined in the next definition is the main step.

Notation 3.4.9. If an arbitrary function h maps into a product $A_0 \times \cdots \times A_{k-1}$, we denote h combined with the i-th projection by h_i. In particular, h_i maps into A_i. If h is of the form $h = f[\pi]$ for some state π then $f_i[\pi] := h_i$.

Definition 3.4.10. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable. For a given Σ^0_1-covering $\rhd$ and a given operator $\circ$ on X let (ι,τ,η) be a cover structure and for given $F \subseteq X$ and $r \in X$ let ψ be a Krull functional. We define the functional $g_{\iota,\tau,\eta,\psi}$ with input $M \subseteq X$ and $f : \mathrm{dom}(X \setminus M) \to \mathcal{P}_{fin}(X) \times W \times X$ and output in $\mathcal{P}_{fin}(X) \times W$ as follows:

$$g_{\iota,\tau,\eta,\psi}(M,f) :=$$
$$\begin{cases}
(\varnothing; w_0) & \text{if } \psi(M) = (0,a)\\
(\{r\}; \iota(r)) & \text{if } \psi(M) = (1,0)\\
\big(A \cup (f_0(i) \setminus \{x_i\}); \tau(A, f_0(i) \setminus \{x_i\}, x_i, r, t, f_1(i))\big) & \text{if } \psi(M) = (2,[\![A,t,x_i]\!])\\
\big((f_0(i) \setminus \{x_i\}) \cup (f_0(j) \setminus \{x_j\}) \cup \{z\};\\
\eta\,(f_0(i) \setminus \{x_i\}, f_0(j) \setminus \{x_j\}, x_i, x_j, z, f_1(i), f_1(j))\big) & \text{if } \psi(M) = (3,[\![x_i,x_j,z]\!])
\end{cases}$$

where $w_0 \in W$ is some canonical element.

As $g_{\iota,\tau,\eta,\psi}(M,f)$ does not depend on the last component of f, we also use the definition above for $f : \mathrm{dom}(X \setminus M) \to \mathcal{P}_{fin}(X) \times W$.

Lemma 3.4.11. In the situation of Definition 3.4.10 let $M \subseteq X$ and $f : \mathrm{dom}(X \setminus M) \to \mathcal{P}_{fin}(X) \times W \times X$ be given such that $F \subseteq M$ and

$$x_n \notin M \Rightarrow F \cup [M](n) \cup \{x_n\} \supseteq f_0(n) \rhd_{f_1(n)} r \wedge x_n \in f_0(n) \tag{3.6}$$

for all n. We set $(A, t) := g_{\iota,\tau,\eta,\psi}(M, f)$. Then $M \supseteq A \rhd_t r$.

Proof. As ψ is a Krull functional, we use case distinction on $\psi(M)$. First note that $\psi(M) = (0, a)$ is not possible because of $F \subseteq M$. Therefore, there are three cases left:

- If $\psi(M) = (1, 0)$ then $r \in M$ and $(A, t) = (\{r\}, \iota(r))$ and indeed $\{r\} \rhd_{\iota(r)} r$ by the property of ι.
- If $\psi(M) = (2, [\![B, s, x_i]\!])$ then we have $B \rhd_s x_i$ for some finite $B \subseteq M$ and $x_i \notin M$. By (3.6) it follows $F \cup [M](i) \cup \{x_i\} \supseteq f_0(i) \rhd_{f_1(i)} r$ and $x_i \in f_0(i)$. Using the property of τ in the definition of cover structure, we get

  $$(f_0(i) \setminus \{x_i\}) \cup B \rhd_{\tau(B, f_0(i)\setminus\{x_i\}, x_i, r, s, f_1(i))} r,$$

  which is $A \rhd_t r$, and $(f_0(i) \setminus \{x_i\}) \cup B \subseteq M$ holds because of $f_0(i) \subseteq F \cup [M](i) \cup \{x_i\}$, $F \subseteq M$ and $B \subseteq M$.
- If $\psi(M) = (3, [\![x_i, x_j, z]\!])$ then $x_i, x_j \notin M$ and $x_i \circ x_j = z \in M$. By (3.6) we have $F \cup [M](i) \cup \{x_i\} \supseteq f_0(i) \rhd_{f_1(i)} r$, $F \cup [M](j) \cup \{x_j\} \supseteq f_0(j) \rhd_{f_1(j)} r$ and $x_i \in f_0(i)$, $x_j \in f_0(j)$. Therefore, by encoding

  $$(f_0(i) \setminus \{x_i\}) \cup (f_0(j) \setminus \{x_j\}), z \rhd_{\eta(f_0(i)\setminus\{x_i\}, f_0(j)\setminus\{x_j\}, x_i, x_j, z, f_1(i), f_1(j))} r,$$

  which is $A \rhd_t r$ in this case. That $A \subseteq M$ holds, follows from $F \cup [M](i) \cup \{x_i\} \supseteq f_0(i)$, $F \cup [M](j) \cup \{x_j\} \supseteq f_0(j)$, $F \subseteq M$ and $z \in M$.

This finishes the proof. □

Lemma 3.4.12. In the situation of Definition 3.4.10 we define the two functionals ω and ϕ for all $S \subseteq X$ and $h : \mathrm{dom}(X \setminus S) \to \mathcal{P}_{fin}(X) \times W \times X$ by[2]

$$\omega(S, h) := \begin{cases} \max\{i \in \mathbb{N} \mid x_i \in A \setminus F\} & \text{if } A \setminus F \neq \varnothing \\ -1 & \text{else,} \end{cases}$$
$$\phi(S, h) := (A, t, r)\,,$$

[2]Note that A is finite and so $A \setminus F = \varnothing$ is decidable and ω is computable.

where A and t are given by

$$(A, t) := g_{\iota,\tau,\eta,\psi_{F,r}}(S, h).$$

Running Algorithm 3.3.6 with these ω and ϕ and $\theta(x) := (x \neq r)$ we get a sequence $(\pi_k)_k$. Then for each k we have $F \subseteq M[\pi_k]$ and

$$x_n \notin M[\pi_k] \Rightarrow F \cup [M[\pi_k]](n) \cup \{x_n\} \supseteq f_0[\pi_k](n) \rhd_{f_1[\pi_k](n)} r \wedge x_n \in f_0[\pi_k](n)$$

for all $n \in \mathbb{N}$.

Proof. Algorithm 3.3.6 updates an state by an output of the functional ϕ. Therefore, $f_2[\pi_k]$ is constantly r for all k since ϕ_2 is constantly r. This together with Remark 3.2.8 and $\theta(x) := (x \neq r)$ gives that

$$\forall_k \forall_i . x_i \notin M[\pi_k] \Rightarrow F \cup [M[\pi_k]](i) \cup \{x_i\} \supseteq f_0[\pi_k](i) \rhd_{f_1[\pi_k](i)} r \tag{3.7}$$

is an instance of Lemma 3.3.7. Therefore, it remains to show for all $k \in \mathbb{N}$,

$$F \subseteq M[\pi_k] \text{ and } \forall_i (x_i \notin M[\pi_k] \Rightarrow x_i \in f_0[\pi_k](i)). \tag{3.8}$$

We prove both assertions by induction on k. If $k = 0$, we have $M[\pi_0] = X$ and there is nothing to show. In the induction step, we assume that k fulfils (3.8). If the algorithm stops with end state π_k, we are done. Otherwise, we have

$$\pi_{k+1} = [\pi_k](n) :: (A, t, r) :: \lambda_n \varepsilon$$

for $(A, t) := g_{\iota,\tau,\eta,\psi_{F,r}}(M[\pi_k], f[\pi_k])$ and some $n \in \mathbb{N}$, and

$$\omega(\pi_k) = \max\{i \in \mathbb{N} \mid x_i \in A \setminus F\}.$$

We claim that $n = \omega(\pi_k)$, where $A \setminus F \neq \varnothing$ as π_k is not an end state.[3] Proof of the claim: If this is not the case then $n < \omega(\pi_k)$, since Algorithm 3.3.6 only checks up to $\omega(\pi_k)$. But then $x_{\omega(\pi_k)} \notin F \cup [M[\pi_{k+1}]](n) \cup \{x_n\}$ and therefore $A \not\subseteq F \cup [M[\pi_{k+1}]](n) \cup \{x_n\}$. By the already proven formula (3.7) and $f_0[\pi_{k+1}](n) = A$, we must have $x_n \in M[\pi_{k+1}]$. This is a contradiction to the choice of n and therefore $n = \omega(\pi_k)$ is proven.

From this claim and the induction hypothesis $F \subseteq M[\pi_k]$, it follows directly that $F \subseteq M[\pi_{k+1}]$, because $x_{\omega(\pi_k)} \notin F$ by definition of ω.

Finally, Let $i \in \mathbb{N}$ with $x_i \notin M[\pi_{k+1}]$ be given. Then either $i = n$ and by definition $x_i = x_{\omega(\pi_k)} \in A = f_0[\pi_{k+1}](\omega(\pi_k))$, or $i < n$, then $f_0[\pi_{k+1}](i) = f_0[\pi_k](i)$ and $x_i \in f_0[\pi_k](i)$ by the induction hypothesis. □

[3] We will use this claim also in Theorem 3.4.15.

Theorem 3.4.13. Assume we are in the situation of Definition 3.4.10 and ω, ϕ, θ are defined as in Lemma 3.4.12. Let π_K be the end state of Algorithm 3.3.6, which exists by Theorem 3.3.12, and $g_{\iota,\tau,\eta,\psi}(M[\pi_K], f[\pi_K]) = (A, t)$, then $F \supseteq A \rhd_t r$.

Proof. By Lemma 3.4.12 we have $F \subseteq M[\pi_K]$ and

$$x_n \notin M[\pi_K] \Rightarrow F \cup [M[\pi_K]](n) \cup \{x_n\} \supseteq f_0[\pi_K](n) \rhd_{f_1[\pi_K](n)} r \wedge x_n \in f_0[\pi_K](n)$$

for all $n \in \mathbb{N}$. Using Lemma 3.4.11 it follows $M[\pi_K] \supseteq A \rhd_t r$, and it remains to show $A \subseteq F$. By Theorem 3.3.8 the end state π_K is an approximate explicit maximal object. In particular,

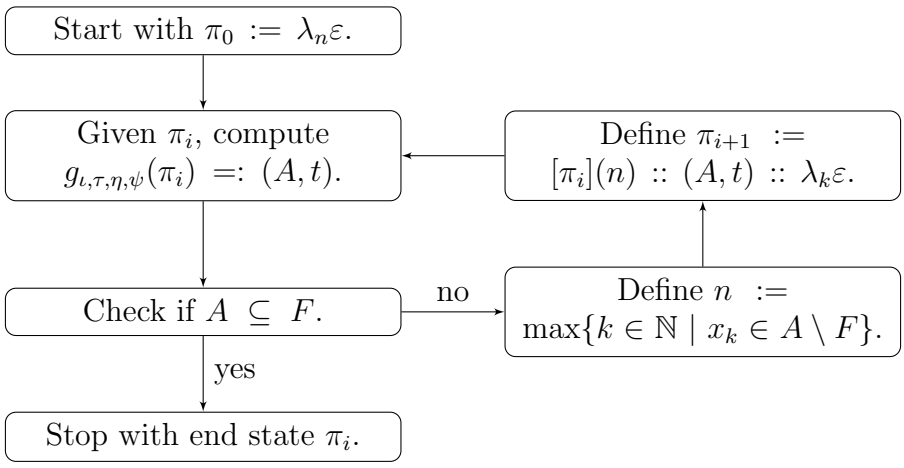
$$x_n \in M[\pi_K] \Rightarrow \neg(F \cup [M[\pi_K]](n) \cup \{x_n\} \supseteq A \rhd_t r) \tag{3.9}$$

for all $n \leq \omega(\pi_K)$. If $A \not\subseteq F$ then $\omega(\pi_K) = \max\{i \in \mathbb{N} \mid x_i \in A \setminus F\} \in \mathbb{N}$ and since $A \rhd_t r$ and $x_{\omega(\pi_K)} \in A \subseteq M[\pi_K]$ together with (3.9) it follows

$$A \not\subseteq F \cup [M[\pi_K]](\omega(\pi_K)) \cup \{x_{\omega(\pi_K)}\}.$$

But this contradicts $A \subseteq M[\pi_K]$ and $\omega(\pi_K) = \max\{i \in \mathbb{N} \mid x_i \in A \setminus F\}$. □

Algorithm 3.4.14. Let $X = \{x_n \mid n \in \mathbb{N}\}$ be countable, $F \subseteq X$, $r \in X$, $\rhd$ be a Σ^0_1-covering on X and $\circ$ be an operator on X. Furthermore, we assume that (ι, τ, η) is a cover structure for $\rhd$ and $\circ$, and ψ is a Krull functional w.r.t. $\rhd$, $\circ$, F and r. We define a sequence of states $(\pi_i)_{i\in\mathbb{N}}$ with $\pi_i : \mathbb{N} \to \mathbb{U} + \mathcal{P}_{fin}(X) \times W$ as follows:



Here $g_{\iota,\tau,\eta,\psi}(\pi_i) := g_{\iota,\tau,\eta,\psi}(M[\pi_i], f[\pi_i])$ is defined as in Definition 3.4.10.

Theorem 3.4.15 (Algorithmic universal Krull-Lindenbaum lemma). In the situation of Algorithm 3.4.14, the algorithm terminates in an end state π_K such that $F \supseteq A \rhd_t r$ for $g_{\iota,\tau,\eta,\psi}(M[\pi_K], f[\pi_K]) = (A, t)$.

Proof. Algorithm 3.4.14 is an almost rewritten form of Algorithm 3.3.6 in the situation of Theorem 3.4.13. There are two difference: The first difference is that a state π maps into $\mathbb{U} + \mathcal{P}_{fin}(X) \times W$ instead of $\mathbb{U} + \mathcal{P}_{fin}(X) \times W \times X$. As we have already observed in the proof of Lemma 3.4.12, the last component is always r, therefore we can drop it. The second difference is that if $\omega(\pi_i) \geq 0$ (i.e. $A \not\subseteq F$), we directly define $n := \max\{k \mid x_k \in A \setminus F\} =: \omega(\pi_i)$ and do not search for $n \leq \omega(\pi_i)$ with $M[\pi_i] \supseteq A \rhd_t r$. That $n = \omega(\pi_i)$ has this property follows by Lemma 3.4.11 and $n < \omega(\pi_i)$ is not possible, which can be proven similar to what we have seen in the proof of Lemma 3.4.12. Therefore, we can directly define $n := \omega(\pi_i)$. Putting things together, the statement follows by Theorem 3.4.13. □

Motivation 3.4.16. In the next sections we discuss several versions of the universal Krull-Lindenbaum lemma from [144, Section 4]. Some of them also appear in [145, Section 4.2]. In Example 3.4.4 we have given the best known version of the universal Krull-Lindebaum lemma.

We obtain the various versions by specifying the set X, the covering $\rhd$ and the operator $\circ$, which we have also done in Example 3.4.4. From a computational point of view this corresponds to the construction of a cover structure.

In each case study we first develop a special version of the universal Krull-Lindenbaum lemma. Then we will define a covering, together with its witness type and an operator, and finally we show that this covering is indeed a Σ_1^0-covering and there is a computable cover structure for this covering and operator. This gives us an algorithmic version of this specialisation of the universal Krull-Lindenbaum lemma. At that point we do not have to define a concrete Krull functional.

The construction of a Krull functional is needed when we use such a version of the Krull-Lindenbaum lemma to prove another statement. We will also give some examples for this.

3.5 Case study: radical ideals in commutative rings

Motivation 3.5.1. The following theorem is the best known version of the universal Krull-Lindenbaum lemma. It is given in many algebra books like Proposition 1.8 of [12]. This version is often called Krull's lemma. There are already many constructive considerations of Krull's lemma done by Peter Schuster, Davide Rinaldi and Daniel Wessel [145, 151].

Here, we are in the setting of commutative rings and we assume that the reader knows the fundamental definition of the theory of communicative rings. Hence, we start directly by formulating the theorem and give a classical proof:

Theorem 3.5.2. Let R be a commutative ring and $F \subseteq R$, then

$$\bigcap \{P \subseteq R \mid F \subseteq P \text{ and } P \text{ is a prime ideal}\} \subseteq \sqrt{(F)}.$$

Here "P is a prime ideal" is meant in the sense of commutative rings, with the exception that R itself shall be a prime ideal, and furthermore $\sqrt{(F)}$ is the radical ideal generated by F.

Proof. This is a special case of Theorem 3.4.3: The covering $\rhd$ is given by $A \rhd x$ as $x \in \sqrt{(A)}$ and $\circ$ as the ring multiplication. That reflexivity holds is trivial.

For transitivity let $x \in \sqrt{(A,y)}$ and $y \in \sqrt{(B)}$ be given. Then there is $e_1, e_2 \in \mathbb{N}$ and $r_0, \ldots, r_n, r'_1, \ldots, r'_m \in R$, $a_1, \ldots, a_n \in A$ and $b_1, \ldots, b_m \in B$ with

$$x^{e_1} = r_0 y + r_1 a_1 + \cdots + r_n a_n \text{ and } y^{e_2} = r'_1 b_1 + \cdots + r'_m b_m.$$

It follows $x^{e_1 e_2} = r_0^{e_2} y^{e_2} + \alpha$, where α is a linear combination of the $a_1, \ldots, a_n$. Using $y^{e_2} = r'_1 b_1 + \cdots + r'_m b_m$ we get that $x^{e_1 e_2}$ is a linear combination of $a_1, \ldots, a_n, b_1, \ldots, b_m$, and hence $x \in \sqrt{(A \cup B)}$.

For encoding, we assume $z \in \sqrt{(A,x)} \cap \sqrt{(B,y)}$. Then there is $e_1, e_2 \in \mathbb{N}$ and $r_0, \ldots, r_n, r'_0, \ldots, r'_m \in R$, $a_1, \ldots, a_n \in A$ and $b_1, \ldots, b_m \in B$ such that $z^{e_1} = r_0 x + r_1 a_1 + \cdots + r_n a_n$ and $z^{e_2} = r'_0 y + r'_1 b_1 + \cdots + r'_m b_m$. Therefore, it follows

$$\begin{aligned} z^{e_1+e_2} =& (r_0 x + r_1 a_1 + \cdots + r_n a_n)(r'_0 y + r'_1 b_1 + \cdots + r'_m b_m) \\ =& (r_0 x + r_1 a_1 + \cdots + r_n a_n)(r'_1 b_1 + \cdots + r'_m b_m) \\ & + (r_1 a_1 + \cdots + r_n a_n) r'_0 y + r_0 r'_0 xy, \end{aligned}$$

and thus $z_{e_1+e_2}$ is a linear combination of the $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$ and xy and hence $z \in \sqrt{(A \cup B, xy)}$. □

Motivation 3.5.3. We will now use the proof as an inspiration to get a cover structure and thereby a computational version of the theorem above.

Definition 3.5.4. Let R be a countable commutative ring whose ring operators can be represented by computable functions.

In this section we define the witness type W by

$$W := R^* \times R^* \times \mathbb{N}.$$

To make it clear, we write $\vec{a}, \vec{b}$ for lists in R^*, components of a list $\vec{a}$ will be denoted by a_i, and $|\vec{a}|$ denotes the length of $\vec{a}$. In particular, $\vec{a} = [a_0, \ldots, a_{|\vec{a}|-1}]$. We define the covering $\rhd$ by

$$A \rhd_{(\vec{a},\vec{p},e)} x :\Leftrightarrow (|\vec{a}| = |\vec{p}| \wedge \vec{a} \in A^* \wedge \vec{p} \cdot \vec{a} = x^e),$$

where $\vec{p} \cdot \vec{a} := \sum_{i=0}^{|\vec{a}|-1} p_i a_i$ (with $p_i := 0$ for $i \geq |\vec{p}|$). Furthermore, we define the operator $\circ$ as the ring multiplication in R.

Definition 3.5.5. Let $x \in R$ and $\vec{a}, \vec{p} \in R^*$. We define a computable function $\sigma(x, \vec{a}, \vec{p})$ as follows:

Let $\vec{b} \in R^*$ be the list which is build from the list $\vec{a}$ by removing exactly those components which are equal to x. Let $\vec{q}$ be the list which is build from $\vec{p}$ by removing all the components which have been removed in $\vec{a}$ to get $\vec{b}$, i.e. if we have removed the n-th component from $\vec{a}$ to get $\vec{b}$, we now remove the n-th component from $\vec{p}$ to get $\vec{q}$. Furthermore, let s be the sum of all components which are removed from $\vec{p}$ to get $\vec{q}$. Then $\vec{p} \cdot \vec{a} = \vec{q} \cdot \vec{b} + sx$ and we define

$$\sigma(x, \vec{a}, \vec{p}) := (\vec{b}, \vec{q}, s).$$

Lemma 3.5.6. In the situation of Definition 3.5.4, the relation $\rhd$ is a Σ_1^0-covering and has together with $\circ$ a computable cover structure (ι, τ, η).

Proof. It is obvious that W is a base type and therefore $\rhd$ is a Σ_1^0-covering. To show the existence of a computable cover structure, we deal with each property in turn:

- Since $x 1_X = x^1$ we have $\{x\} \rhd_{\iota(x)} x$ for $\iota(x) = ([x], [1_X], 1)$.[4]

- Assume $A \rhd_{(\vec{a},\vec{p},e_1)} x$ and $B, a \rhd_{\vec{b},\vec{q},e_2} y$ and let $(\vec{b'}, \vec{q'}, s) := \sigma(x, \vec{b}, \vec{p})$, in particular $\vec{a} \in A^*$ and $\vec{b'} \in B^*$. We define $n = |\vec{a}|$, $m = |\vec{b'}|$, $q'_m := s$ and $b'_m := x$, then

$$\sum_{i=0}^{n-1} p_i a_i = x^{e_1} \qquad \text{and} \qquad \sum_{i=0}^{m} q'_i b'_i = y^{e_2}.$$

[4] Here 1_X denotes the unit element in X.

Using the multinomial theorem and multinomial coefficients we get

$$\begin{aligned}
y^{e_1e_2} &= \left(\sum_{i=0}^{m} q'_i b'_i\right)^{e_1} = \sum_{k_0+\cdots+k_m=e_1} \binom{e_1}{k_0,\ldots,k_m} \prod_{i=0}^{m} {b'}_i^{k_i} {q'}_i^{k_i} \\
&= \sum_{j=0}^{m} \sum_{\substack{k_j+\cdots+k_m=e_1 \\ k_j\neq 0 \\ k_0=\cdots=k_{j-1}=0}} \binom{e_1}{k_0,\ldots,k_m} \prod_{i=j}^{m} {b'}_i^{k_i} {q'}_i^{k_i} \\
&= \sum_{j=0}^{m-1} \left(\sum_{\substack{k_j+\cdots+k_m=e_1 \\ k_j\neq 0 \\ k_0=\cdots=k_{j-1}=0}} \binom{e_1}{k_0,\ldots,k_m} {b'}_j^{k_j-1} {q'}_j^{k_j} \prod_{i=j+1}^{m} {b'}_i^{k_i} {q'}_i^{k_i} \right) b'_j + x^{e_1} s^{e_1} \\
&= \sum_{j=0}^{m-1} \left(\sum_{\substack{k_j+\cdots+k_m=e_1 \\ k_j\neq 0 \\ k_0=\cdots=k_{j-1}=0}} \binom{e_1}{k_0,\ldots,k_m} {b'}_j^{k_j-1} {q'}_j^{k_j} \prod_{i=j+1}^{m} {b'}_i^{k_i} {q'}_i^{k_i} \right) b'_j + \sum_{i=0}^{n-1} p_i s^{e_1} a_i
\end{aligned}$$

Therefore, we set

$$\tau(x,(\vec{a},\vec{p},e_1),(\vec{b},\vec{q},e_2)) := (\vec{a} :: \vec{b'}, \vec{l_1} :: \vec{l_2}, e_1e_2),$$

where

$$\vec{l_1} := (p_i s^{e_1})_{i<|\vec{a}|}$$

$$\vec{l_2} := \left(\sum_{\substack{k_j+\cdots+k_{|\vec{b'}|}=e_1 \\ k_j\neq 0 \\ k_0=\cdots=k_{j-1}=0}} \binom{e_1}{k_0,\ldots,k_{|\vec{b'}|}} {b'}_j^{k_j-1} {q'}_j^{k_j} \prod_{i=j+1}^{|\vec{b'}|} {b'}_i^{k_i} {q'}_i^{k_i} \right)_{j<|\vec{b'}|}$$

$$(\vec{b'},\vec{q'},{q'}_{|\vec{b'}|}) := \sigma(x,\vec{b},\vec{p}) \qquad \text{and} \qquad {b'}_{|\vec{b'}|} := x.$$

– Assume that $A, x \rhd_{(\vec{a},\vec{p},e_1)} z$ and $B, y \rhd_{(\vec{b},\vec{q},e_2)} z$. We define $(\vec{a'},\vec{p'},s_1) := \sigma(x,\vec{a},\vec{p})$ and $(\vec{b'},\vec{q'},s_2) := \sigma(y,\vec{b},\vec{q})$. Defining $n := |\vec{a'}|$ and $m := |\vec{b'}|$, we have

$$\sum_{i=0}^{n-1} p'_i {a'}_i + s_1 x = z^{e_1} \qquad \text{and} \qquad \sum_{i=0}^{m-1} q'_i {b'}_i + s_2 y = z^{e_2}.$$

Therefore, it follows

$$\begin{aligned} z^{e_1+e_2} &= \left(\sum_{i=0}^{n-1} p'_i {a'}_i + s_1 x\right)\left(\sum_{j=0}^{m-1} q'_j {b'}_j + s_2 y\right) \\ &= \sum_{i=0}^{n-1} p'_i \left(\sum_{j=0}^{m-1} q'_j {b'}_j + s_2 y\right) {a'}_i + \sum_{i=0}^{m-1} q'_i s_1 x {b'}_i + s_1 s_2 xy. \end{aligned}$$

Hence, we set

$$\eta(x, y, (\vec{a}, \vec{p}, e_1), (\vec{b}, \vec{q}, e_2)) = (\vec{a'} :: \vec{b'} :: xy, \vec{l_1} :: \vec{l_2} :: s_1 s_2, e_1 + e_2),$$

where

$$\begin{aligned} (\vec{a'}, \vec{p'}, s_1) &:= \sigma(x, \vec{a}, \vec{p}) \\ (\vec{b'}, \vec{q'}, s_2) &:= \sigma(y, \vec{b}, \vec{q}) \\ \vec{l_1} &:= \left(\sum_{j=0}^{|b'|-1} p'_i q'_j {b'}_j + p'_i s_2 y\right)_{i<|\vec{a'}|} \\ \vec{l_2} &:= (q'_i s_1 x)_{i<|\vec{b'}|}. \end{aligned}$$

This finishes the proof. □

Corollary 3.5.7. Assume we are in the situation of Definition 3.5.4. Furthermore, suppose that for any $F \subseteq X$ and $r \in X$ there is a computable Krull functional ψ w.r.t. $\rhd$ and $\circ$. Then instantiating Algorithm 3.4.14 on the cover structure of Lemma 3.5.6 and the Krull functional ψ, the algorithm terminates in some state π_k. Let $g_{\iota,\tau,\eta,\psi}(\pi_k) = (A, (l, e))$ then $A \subseteq F$, $|A| = |l|$ and $A \cdot l = r^e$.

Proof. Follows directly from Theorem 3.4.15 and Lemma 3.5.6. □

3.5.1 Nilpotent coefficients of invertible polynomials

Motivation 3.5.8. The first example is very typical and was already studied in [132, 143, 151] and more subtly in [182]. We first give the classical proof following [151], then we show how this proof can be converted into an algorithm describing a Krull functional.

Proposition 3.5.9. Let R be a commutative ring and $f = \sum_{i=0}^{n} a_i X^i$ be an unit in $R[X]$. Then each a_i with $i > 0$ is nilpotent.

Proof. Let $g = \sum_{j=0}^{m} b_j X^j$ be the inverse of f, in particular $1 = fg = \sum_{k=0}^{n+m} c_k X^k$ for $c_k = \sum_{i+j=k} a_j b_{k-j}$. Fixing some $i > 0$, our goal is to show that a_i is nilpotent. By using Theorem 3.5.2 with $F = \varnothing$, we show that $r := a_i \in P$ for each prime ideal $P \subseteq R$ (which are also prime $\rhd$-ideals). Hence, we assume that $a_i \notin P$ for some fixed prime ideal P. Our goal is a contradiction.

We have $0 = c_i = \sum_{j=0}^{i-1} a_j b_{i-j} + a_i b_0$. Since $1 = c_0 = b_0 a_0$, we get

$$a_i = -a_0 \sum_{j=0}^{i-1} a_j b_{i-j}. \tag{3.10}$$

As P is a prime ideal and $a_i \notin P$, there must be $1 \leq j \leq i$ with $b_j \notin P$. Therefore, we may take $1 \leq k \leq n$ and $1 \leq l \leq m$ maximal such that $a_k, b_l \notin P$ and consider

$$0 = c_{k+l} = a_k b_l + \sum_{\substack{p+q=k+l \\ p>k}} a_p b_q + \sum_{\substack{p+q=k+l \\ q>l}} a_p b_q. \tag{3.11}$$

Since P is a prime ideal and $a_k, b_l \notin P$, we have $a_k b_l \notin P$. But, by the choice of k and l, we have $a_p \in P$ for all $p > k$ and $b_q \in P$ for all $q > l$, and therefore

$$a_k b_l = \sum_{\substack{p+q=k+l \\ p>k}} (-a_p) b_q + \sum_{\substack{p+q=k+l \\ q>l}} (-a_p) b_q \in P, \tag{3.12}$$

a contradiction. □

Algorithm 3.5.10. Let R be a countable ring whose ring operations can be represented by computable functions. Let furthermore $f = \sum_{i=0}^{n} a_i X^i \in R[X]$ and $g = \sum_{j=0}^{m} b_j X^j \in R[X]$ with $fg = \sum_{k=0}^{n+m} c_k = 1$ be given. For given a_i with $i > 0$, we define a function ψ with takes any $P \in 2^R$ and returns an element in $\{0, 1, 2, 3\} \times \mathbb{N}$. For simplicity, we use the countability of R to consider an element x in R as an element in $\mathbb{N}$.

1. Check if $0_R \in P$. If not, return $(2, [\![\varnothing, ([], [], 1), 0]\!])$.
2. Check if $a_i \in P$. If not, return $(1, 0)$.
3. Check if $b_1, \ldots, b_i \in P$. If yes, return

$$(2, [\![\{b_1, \ldots, b_i\}, ([b_1, \ldots, b_i], [a_{i-1}, \ldots, a_0], a_i), 1]\!])$$

4. Otherwise, take k, l maximal with $a_k \notin P$ and $b_l \notin P$, and check if $a_k b_l \in P$. If yes, return $(3, [\![a_k, b_l, a_k b_l]\!])$.

5. Else, return

$$\left(2, [\![\{\vec{d}\}, ([\vec{d}], [-b_{l-1}, \ldots, -b_0, -a_{k-1}, \ldots, a_0], 1), a_k b_l]\!]\right),$$

where $\vec{d}$ is a syntactical abbreviation for $a_{k+1}, \ldots, a_{k+l}, b_{l+1}, \ldots, b_{l+k}$.

Lemma 3.5.11. In the situation of Algorithm 3.5.10 and $\rhd$ and $\circ$ are given as in Definition 3.5.4, the defined function ψ is a Krull functional for $a_i \in R$ and $\varnothing \subseteq R$.

Proof. This is a straightforward case distinction, checking the definition of a Krull functional and following the second part of the proof of Proposition 3.5.9. The first and the second case are clear. For the first case, note that the empty sum is equal to zero by definition. For the third case note that (analogously to the proof of Proposition 3.5.9) we get (3.10) and so if $b_1, \ldots, b_i \in P$ but (by the second case) $a_i \notin P$, P is not a prime ideal and the output in this case indeed gives the concrete evidence. The fourth case follows directly since $a_k, b_l \notin P$ and $a_k b_l \in P$ are evidence that P is not prime. The fifth case corresponds to the Formula 3.12 in the proof of Proposition 3.5.9. Note that $a_{k+1}, \ldots, a_{k+l}, b_{l+1}, \ldots, b_{l+k} \in P$ by maximality of k and l. □

Corollary 3.5.12. Let R be a countable commutative ring. Suppose that $f = \sum_{i=0}^{n} a_i X^i \in R[X]$ has an inverse $g = \sum_{i=0}^{m} b_i X^i \in R[X]$ and take $i > 0$. Let Algorithm 3.4.14 be instantiated on the cover structure (ι, τ, η) from Lemma 3.5.6 and the Krull functional ψ defined in Algorithm 3.5.10. Then the algorithm terminates in some state π_K and for $g_{\iota,\tau,\eta,\psi}(\pi_K) = (\varnothing, ([], [], e))$ and $a_i^e = 0_R$.

Proof. This follows directly by Corollary 3.5.7 and Lemma 3.5.11. □

Remark 3.5.13. In the corollary above, if R is an arbitrary (not necessarily countable) ring, we can consider the ring which is generated by $a_1, \ldots, a_n, b_1, \ldots, b_m$. Therefore, we can apply the corollary above to this ring if it is countable, which is at least true in classical logic. Hence, countability is not a major restriction.

3.5.2 The theorem of Gauß-Joyal

Motivation 3.5.14. We consider a second construction, this time arising from the following result, commonly called the Gauss-Joyal theorem. The constructive meaning of it was already considered constructively in many papers like [8, 14, 43, 45, 71, 120]. We proceed similarly to the last section:

Proposition 3.5.15 (Theorem of Gauß-Joyal). Let R be a commutative ring and $f = \sum_{i=0}^{n} a_i X^i$ and $g = \sum_{i=0}^{m} b_i X^i$ two polynomials in $R[X]$ with $fg = \sum_{i=0}^{n+m} c_i X^i$. Then

$$a_i b_j \in \sqrt{(c_0, \ldots, c_{i+j})}$$

for all $i \in \{0, \ldots, n\}$ and $j = \{0, \ldots, m\}$.

Proof. Let $i \in \{0, \ldots, n\}$ and $j = \{0, \ldots, m\}$ be given and define $F := \{c_0, \ldots, c_{i+j}\}$. Using Theorem 3.5.2 we show that $a_i b_j \in P$ for each prime ideal P with $F \subseteq P$. Therefore, we assume $a_i b_j \notin P$ for such P. Since P is prime and $a_i b_j \notin P$, we have $a_i, b_j \notin P$ and so we define k, l to be minimal such that $a_k, b_l \notin P$. Then also $a_k b_l \notin P$ since P is prime. We have

$$c_{k+l} = \sum_{i+j=k+l} a_i b_j = a_k b_l + \sum_{\substack{i+j=k+l \\ i<k}} a_i b_j + \sum_{\substack{i+j=k+l \\ j<l}} a_i b_j \tag{3.13}$$

and $k + l \leq i + j$ and so $c_{k+l} \in F$. Therefore, by minimality k, l we have that $a_i \in P$ for $i < k$ and $b_j \in P$ for $j < l$. By (3.13) it follows $a_k b_l \in P$, a contradiction. □

Algorithm 3.5.16. Let R be a countable commutative ring. Suppose that $f = \sum_{i=0}^{n} a_i X^i$ and $g = \sum_{i=0}^{m} b_i X^i$ are polynomials and $fg = \sum_{i=0}^{n+m} c_i X^i$. Furthermore, let $i \in \{0, \ldots, n\}$ and $j \in \{0, \ldots, m\}$ be given. We define a function ψ which takes $P \in 2^R$ and return an element in $\{0, 1, 2, 3\} \times \mathbb{N}$. As before, for simplicity we use the countability of R to consider an element in R as an element in $\mathbb{N}$.

1. Check if $c_k \notin P$ for some $k \in \{0, \ldots, i + j\}$. If yes, return $(0, c_k)$.

2. Check if $a_i b_j \in P$. If yes, return $(1, 0)$.

3. Check if either $a_i \in P$ or $b_j \in P$. If yes, return $(2, [\{a_i\}, ([a_i], [b_j], 1), a_i b_j])$ or $(2, [\{b_j\}, ([b_j], [a_i], 1), a_i b_j])$, respectively.

4. Compute k, l minimal with $a_k, b_l \notin P$ and check if $a_k b_l \in P$. If yes, return $(3, [a_k, b_l, a_k b_l])$.

5. Else, return $(2, [\{\vec{d}\}, ([\vec{d}], [1_S, -b_{k+l}, \ldots, -b_{l+1}, -a_{k+1}, \ldots, -a_{k+l}], 1), a_k b_l])$ where $\vec{d}$ is a syntactical abbreviation for $c_{k+l}, a_0, \ldots, a_{k-1}, b_{l-1}, \ldots, b_0$.

Lemma 3.5.17. In the situation of Algorithm 3.5.16 and $\rhd, \circ$ given as in Definition 3.5.4, the defined function ψ is a Krull functional for $a_i \in R$ and $F = \{c_0, \ldots, c_{i+j}\}$.

Proof. This follows straightforward by using case distinction and the definition of a Krull functional. The first four cases are clear and for the last case note that $\{c_{k+l}, a_0, \ldots, a_{k-1}, b_{l-1}, \ldots, b_0\} \subseteq P$ follows by minimality of k and l and the fact that $k+l \leq i+j$ and the failure of the first case. Note that the fifth case corresponds to Formula (3.13) in the proof of Proposition 3.5.9. □

Corollary 3.5.18 (Algorithmic version of the theorem of Gauß-Joyal). Let R be a countable commutative ring and let $f = \sum_{i=0}^{n} a_i X^i$ and $g = \sum_{i=0}^{m} b_i X^i$ in $R[X]$ with $fg = \sum_{i=0}^{n+m} c_i X^i$ be given and assume $0 \leq i \leq n$ and $0 \leq j \leq m$. Suppose that Algorithm 3.4.14 is instantiated on the cover structure (ι, τ, η) from Lemma 3.5.6 and the Krull functional defined in Algorithm 3.5.16. Then the algorithm terminates in some state π_K and for $g_{\iota,\tau,\eta,\psi}(\pi_K) = (A, (\vec{a}, \vec{p}, e))$ we have $\{\vec{a}\} \subseteq A \subseteq \{c_0, \ldots, c_{i+j}\}$ and $\vec{a} \cdot \vec{p} = (a_i b_j)^e$

Proof. Follows by Corollary 3.5.7 and Lemma 3.5.17. □

Remark 3.5.19. Similar to the remark after Corollary 3.5.12, it suffices to consider the ring which is generated by $a_1, \ldots, a_n, b_1, \ldots, b_m$, and apply the corollary above to this ring. Therefore, the assumption that R is countable can be reduced to the assumption that the generated ring is countable.

3.6 Case study: valuation rings and integral closures

Motivation 3.6.1. The next example is a well-known statement of commutative algebra. It was considered constructively in [56] and [111], which we mainly use a source of this section.

Definition 3.6.2. Let a ring K and a subring $R \subseteq K$ be given. An element $x \in K$ is called *integral* over R if there is $n \in \mathbb{N}$ and $a_0, \ldots, a_{n-1} \in R$ with

$$x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = 0.$$

In other word: there is a monic polynomial $p \in R[X]$ such that $p(x) = 0$.

The *integral closure* of the ring R w.r.t. the field K is the set of all $x \in K$ which are integral over R and is denoted by $\overline{R}$. If $R = \overline{R}$, we say the R is *integrally closed*.

Lemma 3.6.3. Let a ring K and a subring $R \subseteq K$ be given. The integral closure of R is a subring of K.

Proof. Corollary 5.3 of [12]. □

Definition 3.6.4. Let K be a field. A subring $R \subseteq K$ is called a *valuation ring* if for each $x \in K \setminus \{0\}$ either $x \in R$ or $x^{-1} \in R$.

Lemma 3.6.5. Let K be a field and $R \subseteq K$ be a valuation ring of K. Then R is integrally closed.

Proof. Let $x \in K$ be integral over R. Then there are $n \in \mathbb{N}$ and $a_{n-1}, \ldots, a_0 \in R$ with

$$x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = 0$$

If $x = 0$, we directly have $x \in R$. Hence, we assume $x \neq 0$ and therefore $n \geq 1$. Since R is a valuation ring, either $x \in R$ or $x^{-1} \in R$. In the first case we are done. In the second case we multiply the equation above with $x^{-(n-1)}$ and compute

$$x = -a_{n-1} - \cdots + a_1 x^{-n+2} - a_0 x^{-n+1} \in R.$$

□

Theorem 3.6.6. Let K be a field, $R \subseteq K$ be a subring, then

$$\bigcap \{S \subseteq K \mid R \subseteq S \text{ and } S \text{ is a valuation ring of K}\} \subseteq \overline{R}.$$

Proof. We use Theorem 3.4.3 with $\rhd$ and $\circ$ defined by; $A \rhd x$ iff x is integral over $R[A]$, and $x \circ y := xy$. It remains to show reflexivity, transitivity and encoding. The proofs of the three properties are quite similar to the proof of Lemma 3.6.12. Hence, we do not give details here. □

Notation 3.6.7. Let R be any ring, $\vec{x} \in K^*$ and $I \in \mathbb{N}$ be given with $|I| = |\vec{x}| =: n$. We set $\vec{x}^I := x_0^{I_0} \ldots x_{n-1}^{I_{n-1}}$ and $\|I\| := \max\{I_0, \ldots, I_{n-1}\}$.

Definition 3.6.8. Let K be a countable field whose field operators are computable and let $E \subseteq K$ be a subring of K. We define the witness type by

$$W := K^* \times (\mathbb{N} \times E^{\mathbb{N}^*})^*$$

and the covering by

$$A \rhd_{\vec{a},((n_0,f_0),\ldots,(n_{l-1},f_{l-1}))} x :\Leftrightarrow \vec{a} \in A^* \wedge x^l + \sum_{i=0}^{l-1} x^i \sum_{\substack{I \in \mathbb{N}^* \\ |I|=|\vec{a}| \\ \|I\| \leq n_i}} f_{i,I}\vec{a}^I = 0$$

Furthermore, we define the operator $\circ$ as the multiplication in K

Remark 3.6.9. The definition above is very formal. For simplicity, given

$$\vec{a}, [(n_0, f_0), \ldots, (n_{l-1}, f_{l-1})] \in K \times (\mathbb{N} \times E^{\mathbb{N}^*})^*,$$

we consider the list $[(n_0, f_0), \ldots, (n_{l-1}, f_{l-1})]$ as a monic polynomial $p \in E[\vec{a}][X]$ by

$$p := X^l + \sum_{i=0}^{l-1} X^i \sum_{\substack{I \in \mathbb{N}^* \\ |I| = |\vec{a}| \\ \|I\| \leq n_i}} f_{i,I} \vec{a}^I.$$

Hence, we write $A \rhd_{\vec{a},p} x$ if $\vec{a} \in A^*$, $p \in E[\vec{a}][X]$, p is monic and $p(x) = 0$. Note that the arithmetic operations on $E[\vec{a}][X]$ can be coded as operations on $(\mathbb{N} \times E^{N^*})^*$.

Proposition 3.6.10. Assume we are in the situation of Definition 3.6.8 and let $U \subseteq K$ be any set, then $\langle U \rangle^{\rhd} = \overline{E[U]}$. In particular, the $\rhd$-ideals are exactly the integrally closed subrings of K which contain E. Furthermore, the prime $\rhd$-ideals are exactly the valuation rings of K which contain E.

Proof. We have $x \in \overline{E[U]}$ if and only if there is $a_0, \ldots, a_{n-1} \in E[U]$ such that $x^n + a_{n-1}x^{n-1} + \cdots + a_1 x - a_0 = 0$. Since $r_i \in E[U]$ for each i, there is $u_{i1}, \ldots, u_{ik_i}$ and $f \in E[X_1, \ldots, X_{k_i}]$ with $a_i = f(u_{i1}, \ldots, u_{ik_i})$. Therefore, we have $a_i \in E[A]$ for

$$A := \{u_{ij} \mid i \in \{0, \ldots, n-1\}, j \in \{1, \ldots, k_i\}\}.$$

Hence, we conclude $x \in \overline{E[U]}$ if and only if $x \in E[A]$ for some finite $A \subseteq U$, i.e. $a \in \langle U \rangle^{\rhd}$.

It remains to show that the valuation rings are exactly the prime $\rhd$-ideals. By Lemma 3.6.5 and the first part of this lemma, it follows that a valuation ring is a $\rhd$-ideal. Therefore, it suffices to show that a $\rhd$-ideal $P \subseteq K$ is a valuation ring if and only if it fulfils the prime property. Assume that P is a valuation ring and let $a, b \in K$ with $a \circ b = ab \in K$ be given. If $a \notin P$ then $a \neq 0$ and $a^{-1} \in P$. Since P is a subring of K, it follows $b = aba^{-1} \in P$. Hence, $a \in P$ or $b \in P$. That each prime $\rhd$-ideal P is a valuation follows directly by $1 \in P$ and $1 = aa^{-1}$ for all $a \in K \setminus \{0\}$. □

Notation 3.6.11. Let $\vec{a}$ be any list and S be any set. With $\vec{a} \setminus S$ we denote the list which is computed from $\vec{a}$ by removing each component which is in S. Formally, we set

$$[] \setminus S := [],$$
$$(\vec{a} :: y) \setminus S := \begin{cases} \vec{a} \setminus S & \text{if } y \in S \\ (\vec{a} \setminus S) :: y & \text{else.} \end{cases}$$

Lemma 3.6.12. Assume we are in the situation of Definition 3.6.8, then the relation $\rhd$ is a Σ^0_1-covering and has a computable cover structure (ι, τ, η) together with the operator $\circ$.

Proof. Since W in Definition 3.6.8 is a base type, $\rhd$ is indeed a Σ^0_1-covering. For the proof that there exists a computable cover structure, we deal with each property in turn and we use the simplification from Remark 3.6.9.

- We obviously have $\{x\} \rhd_{[x],p} x$ for $p = X - x \in E[x]$. Hence, we define $\iota(x) := ([x], X - x)$.
- Let $A \rhd_{\vec{a},p} x$ and $B, x \rhd_{\vec{b},q} y$ be given. We define $n := \deg(p)$ and $m := \deg(q)$. Then we have

  $$p(x) = x^n + p_{n-1}x^{n-1} + \cdots + p_1 x + p_0 = 0,$$

  where each $p_i \in E[\vec{a}]$. Let $\vec{b'} := \vec{b} \setminus \{x\}$, from q we can compute some $l \in \mathbb{N}$ and $q_0, \ldots, q_l \in E[\vec{b'}][X]$ with $\deg(q_i) < m$ for all i such that

  $$P_2(x) := q_l(y)x^l + \cdots + q_1(y)x + y^m + q_0(y) = 0,$$

  where $P_2 \in E[\vec{b'}, y][X]$ is a polynomial with $\deg(P_2) = l$ (i.e. we choose l such that $q_l(y) \neq 0$). Then the resultant $\mathrm{res}(p, P_2) = 0$ because p and P_2 have the common zero x (Follows from [32, Satz 6 on page 175]). As defined in [32], the resultant is the determinant of the Sylvester matrix:

  $$\left.\begin{pmatrix} 1 & p_{n-1} & \cdots & p_0 & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & 1 & p_{n-1} & \cdots & p_0 \\ q_l(y) & \cdots & q_1(y) & y^m + q_0(y) & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & q_l(y) & q_{l-1}(y) & \cdots & y^m + q_0(y) \end{pmatrix}\right\} \begin{matrix} \left.\vphantom{\begin{matrix}a\\b\\c\\d\end{matrix}}\right\} l \text{ rows} \\ \left.\vphantom{\begin{matrix}a\\b\\c\\d\end{matrix}}\right\} n \text{ rows} \end{matrix}$$

  Considering the resultant as a polynomial in y, we obtain a polynomial $Q \in E[\vec{a}, \vec{b'}][X]$ with $Q(y) = 0$. To see that this polynomial is monic, observe that each non-zero summand of the determinant of the Sylvester matrix selects l elements from $\{1, p_{n-1}, \ldots, p_0\}$ and n elements from $\{q_l(y), \ldots, q_1(y), y^m + q_0(y)\}$. The p_i are constant in y and the degree of the q_i is smaller then m. So each non-zero summand has degree smaller than mn, except for the product

of the diagonal entries which is given by $1^l(y^m + q_0(y))^n$. Since $\deg(q_0) < m$, this is a monic polynomial in y, and therefore Q is monic. Hence,

$$\tau(A, B, (\vec{a}, p), (\vec{b}, q), x, y) := (\vec{a} :: (\vec{b} \setminus \{x\}), Q)$$

for Q given as above.

– Suppose that $A, x \rhd_{\vec{a},p} z$ and $B, y \rhd_{\vec{b},q} z$. We set $\vec{a}' := \vec{a} \setminus \{x\}$, $\vec{b}' := \vec{b} \setminus \{y\}$, $n := \deg(p)$ and $m := \deg(q)$. Hence, from p we can compute some $k \in \mathbb{N}$ and $p_0, \ldots, p_k \in E[\vec{a}'][X]$ such that

$$p_k(z)x^k + \cdots + p_1(z)x + z^n + p_0(z) = 0$$

and $\deg(p_i) < n$ for all i. Similarly, from q we can compute some $l \in \mathbb{N}$ and $q_0, \ldots, q_l \in E[\vec{b}']$ such that

$$P_2(y) := q_l(z)y^l + \cdots + q_1(z)y + z^n + q_0(z) = 0$$

Now, we multiply the first equation with y^k and define $\overline{p_i}(z) := p_i(z)(xy)^i$ for $i \in \{1, \ldots, k\}$, where $\overline{p_i} \in E[\vec{a}', xy][X]$ with $\deg(\overline{p}_i) < n$, to obtain

$$P_1(y) := (z^n + p_0(z))y^k + \overline{p}_1(z)y^{k-1} + \cdots + \overline{p}_{k-1}(z)y + \overline{p}_k(z) = 0.$$

Since $P_1(y) = P_2(y) = 0$, the resultant $\mathrm{res}(P_1, P_2) \in E[\vec{a}', \vec{b}', xy][z]$ is equal to zero. From this resultant we can compute a polynomial $Q \in E[\vec{a}', \vec{b}', xy][X]$ with $Q(z) = 0$. It remains to show that Q is monic: the Sylvester matrix is given by

$$\begin{pmatrix}
z^n + p_0(z) & \overline{p}_1(z) & \ldots & \overline{p}_k(z) & & 0 \\
 & \ddots & \ddots & & \ddots & \\
0 & & z^n + p_0(z) & \overline{p}_1(z) & \ldots & \overline{p}_k(z) \\
q_l(z) & \ldots & q_1(z) & z^m + q_0(z) & & 0 \\
 & \ddots & & \ddots & \ddots & \\
0 & & q_l(z) & \ldots & q_1(z) & z^m + q_0(z)
\end{pmatrix}$$

Each non-zero summand of the determinant of this matrix involves the multiplication of l elements in $\{z^n + p_0(z), \overline{p}_1(z), \ldots, \overline{p}_k(z)\}$ and k elements in $\{q_l(z), \ldots, q_1(z), z^m + q_0(z)\}$ and thus the degree in z for each summand is bounded by $nl + mk$. But since $\deg(\overline{p}_i) < n$ and $\deg(q_j) < m$ for all i and

j, this degree is actually smaller than $nl + mk$, except for the product of the diagonal entries which is given by

$$(z^n + p_0(z))^l (z^m + q_0(z))^k.$$

This is monic because $\deg(p_0) < n$ and $\deg(q_0) < m$. Therefore, we define

$$\eta(A, B, x, y, z, (\vec{a}, p), (\vec{b}, q)) := ((\vec{a} \setminus \{x\}) :: (\vec{b} \setminus \{y\}), Q),$$

where Q is given as above.

This finishes the proof. □

Corollary 3.6.13. Assume we are in the situation of Definition 3.6.8 and suppose that for $F \subseteq K$ and $r \in K$ we have a Krull functional ψ w.r.t. $\rhd$ and $\circ$. Then instantiating Algorithm 3.4.14 on the cover structure of Lemma 3.6.12 and the Krull functional ψ, the algorithm terminates in some state π_k. Let $g_{\iota,\tau,\eta,\psi}(\pi_k) = (A, (\vec{a}, p))$, then $A \subseteq F$, $\vec{a} \in A^*$, p can be seen as monic polynomial in $E[\vec{a}][X]$ and $p(x) = 0$.

Proof. Follows directly from Theorem 3.4.15 and Lemma 3.6.12. □

3.6.1 Kronecker's theorem

Motivation 3.6.14. In this section we show how we can derive a Krull functional from a proof of Kronecker's theorem, which can classically be proven by using Theorem 3.6.6. We first give a lemma and then the proof of Kronecker's theorem, where we follow the proofs given in [56, 111]. Kronecker's theorem and Dedekind's Prague theorem, which is a special case, are also considered constructively in [68, 109]. We use the classical proof as starting point to define a computable Krull functional in three steps. Hence, at the end we will get an algorithmic version of Kronecker's theorem.

The names "Kronecker's theorem" and "Dedekind's Prague theorem" have different meanings in the literature. We use the same names as in sources above, i.e. Kronecker's theorem is the statement in Proposition 3.6.16, and Dedekind's Prague theorem is the special case of Kronecker's theorem where $E := \mathbb{Z}$ and K is the algebraic closure of $\mathbb{Q}$.

Lemma 3.6.15. Let a field K and a valuation ring P of K be given. Then for any $u_1, \ldots, u_n \in K$ with $(u_1 + \cdots + u_n)^{-1} \in P$ there exists at least one i with $u_i^{-1} \in P$.[5]

[5]When we use the inverse of some element x, we implicitly assume that x is invertible. In particular, we write $x^{-1} \in P$ as an abbreviation for $x \neq 0 \wedge x^{-1} \in P$.

Proof. We use induction over n: for $n = 0$ we have by definition $u_1 + \cdots + u_n = 0$ and therefore the premise is always false. For the induction step suppose that $w := (u_1 + \cdots + u_{n+1})^{-1} \in P$. If $u_{n+1} = 0$, we are done by the induction hypothesis. Otherwise, u_{n+1} is invertible and we have $(u_1 + \cdots + u_n)w + u_{n+1}w = 1$ and thus

$$u_{n+1}^{-1}w^{-1} = u_{n+1}^{-1}v + 1,$$

where $v := u_1 + \cdots + u_n$. If $v = 0$, we directly have $u_{n+1}^{-1} = w \in P$. If $v \neq 0$ then $v^{-1}w^{-1} - 1 = v^{-1}u_{n+1}$, and thus

$$(v^{-1}w^{-1} - 1)(u_{n+1}^{-1}w^{-1} - 1) = v^{-1}w^{-1} - v^{-1}u_{n+1} = 1 \in P.$$

Since P is a valuation ring, either $(v^{-1}w^{-1} - 1) \in P$ and thus $v^{-1} \in P$, in which case we are done by the induction hypothesis, or $(u_{n+1}^{-1}w^{-1} - 1) \in P$ and thus $u_{n+1} \in P$. □

Proposition 3.6.16 (Kronecker's Theorem). Let E be a subring of a field K and $a_0, \ldots, a_m, b_0, \ldots, b_n \in K \setminus \{0\}$ and $c_k := \sum_{i+j=k} a_i b_j$. Then $a_i b_j$ is integral over $E[c_1, \ldots, c_{m+n}]$ for any i, j.

Proof. Let $a_k b_l$ be given. We use Theorem 3.6.6 with $R = E[c_0, \ldots, c_{m+n}]$. Therefore, it suffices to show that $a_k b_l \in P$ for any valuation ring of K which contains E and $c_0, \ldots, c_{n+m}$. We define the relation $\leq_P$ on $K \setminus \{0\}$ by

$$x \leq_P y :\Leftrightarrow x^{-1}y \in P.$$

Since P is a subring, the relation $\leq_P$ is reflexive and transitive, i.e. a preorder. It is even a total preorder, i.e. $x \leq y$ or $y \leq x$ for all $x, y \in K \setminus \{0\}$, because P is a valuation ring and therefore either $x^{-1}y \in P$ or $xy^{-1} \in P$ for all $x, y \in K \setminus \{0\}$.

Using reflexivity, transitivity and totality, it follows that there are i_0 and j_0 with $a_{i_0} \leq_P a_i$ and $b_{j_0} \leq_P b_j$ for all i, j. We take i_0 and j_0 maximal with this property. As $x \leq_P y$ and $x' \leq_P y'$ imply $xx' \leq_P yy'$, we have $a_{i_0}b_{j_0} \leq a_k b_l$. As $a_k b_l \in P$ follows from $a_{i_0}b_{j_0} \in P$, it suffices to show $a_{i_0}b_{j_0} \in P$: By the definition of $c_{i_0+j_0}$ we have

$$a_{i_0}b_{j_0} = c_{i_0+j_0} - \sum_{\substack{i+j=i_0+j_0 \\ i \neq i_0}} a_i b_j.$$

Since $a_{i_0}b_{j_0} \neq 0$, it follows

$$c_{i_0+j_0}a_{i_0}^{-1}b_{j_0}^{-1} - \sum_{\substack{i+j=i_0+j_0 \\ i>i_0}} a_i b_j a_{i_0}^{-1}b_{j_0}^{-1} - \sum_{\substack{i+j=i_0+j_0 \\ j>j_0}} a_i b_j a_{i_0}^{-1}b_{j_0}^{-1} = 1$$

In particular, also the inverse of the sum above is equal 1, and therefore in P. By Lemma 3.6.15 we have three cases:

– If $c_{i_0+j_0}^{-1} a_{i_0} b_{j_0} \in P$ then $a_{i_0} b_{j_0} \in P$ and we are done.

– If $a_i^{-1} b_j^{-1} a_{i_0} b_{j_0} \in P$ for some $i > i_0$ then by $b_{j_0} \leq_P b_j$ we have $a_i \leq_P a_{i_0}$ but this is not possible by the maximality of i_0 and the transitivity of $\leq_P$.

– If $a_i^{-1} b_j^{-1} a_{i_0} b_{j_0} \in P$ for some $j > j_0$ then by $a_{i_0} \leq_P a_i$ we have $b_j \leq_P b_{j_0}$ but this is not possible by the maximality of j_0 and the transitivity of $\leq_P$.

This finishes the proof. □

Definition 3.6.17. In the situation of Definition 3.4.7, where a Krull functional is defined, a *partial Krull functional* has the same properties as a Krull functional with the exception that it is defined on a subset of 2^X.

Algorithm 3.6.18. Let a countable field K and $[u_1, \ldots, u_n] \in K^*$ be given. We define a function $\phi^1_{[u_1,\ldots,u_n]}(P)$ on all $P \in 2^K$ with $1 \in P$, $(u_1 + \cdots + u_n)^{-1} \in P$ and $u_i^{-1} \notin P$ for all i as follows by recursion:

1. If $u_n = 0$, return $\phi^1_{[u_1,\ldots,u_{n-1}]}(P)$.

2. Define $w := (u_1 + \cdots + u_n)^{-1}$ and $v = u_1 + \cdots + u_{n-1}$ and note that $v \neq 0$ because otherwise $P \ni w = u_n^{-1} \notin P$. Hence, check if $u_n^{-1} w^{-1} - 1 \in P$. If yes, return

$$(2, [\![\{u_n^{-1} w^{-1} - 1, w\}, ([u_n^{-1} w^{-1} - 1, w], X - (u_n^{-1} w^{-1} - 1)w - w), u_n^{-1}]\!]).$$

3. Check if $v^{-1} w^{-1} - 1 \in P$. If yes, consider two cases:

   (a) If $v^{-1} \notin P$, return

$$(2, [\![\{v^{-1} w^{-1} - 1, w\}, ([v^{-1} w^{-1} - 1, w], X - (v^{-1} w^{-1} - 1)w - w), v^{-1}]\!]).$$

   (b) If $v^{-1} \in P$, return $\phi^1_{[u_1,\ldots,u_{n-1}]}(P)$.

4. Else, return $(3, [\![v^{-1} w^{-1} - 1, u_n^{-1} w^{-1} - 1, 1]\!])$.

Lemma 3.6.19. In the situation of Algorithm 3.6.18, the function $\phi^1_{[u_1,\ldots,u_n]}$ is a partial Krull functional (w.r.t. any $F \subseteq K$ and $r \in K$) defined on all P with $1 \in P$, $(u_1 + \cdots + u_n)^{-1} \in P$ and $u_i^{-1} \notin P$ for all i.

Proof. The proof is done by induction over n and checking each case in the algorithm. Let P be given and assume $1 \in P$, $(u_1 + \cdots + u_n)^{-1} \in P$ and $u_i \notin P$ for all i. We have to check that $\phi^1_{[u_1,\ldots,u_n]}(P)$ fulfils the properties in Definition 3.4.7. By Step 1 and the induction hypothesis, we may assume $u_n \neq 0$. As mentioned in Step 2 we can assume that $n \geq 2$ since if $n = 1$, we would have $v = 0$, and $n = 0$ is not possible because $u_1 + \cdots + u_n$ is invertible by assumption. If $u_n^{-1}w^{-1} - 1 \in P$ then $\{u_n^{-1}w^{-1} - 1, w\} \subseteq P$ and $X - (u_n^{-1}w^{-1} - 1)w + w = X - u_n^{-1}$ and so the out put is justified. Analogously, the output in Case 3a is justified. In Step 3b we can directly use the induction hypothesis. Finally, in Step 4 we have

$$(u_n^{-1}w^{-1} - 1)(u^{-1}w^{-1} - 1) = 1$$

as in the proof of Lemma 3.6.15 and $1 \in P$ by assumption. But, by the failure of Step 2 and 3, none of the factors are in P. Therefore, in this case $\phi^1_{[u_1,\ldots,u_n]}$ also fulfils the property of a partial Krull functional. □

Definition 3.6.20. Let K be a countable field. For given $P \subseteq 2^K$ we define the binary relation $\leq_P$ on $K \setminus \{0\}$ by $x \leq_P y :\Leftrightarrow x^{-1}y \in P$.

Algorithm 3.6.21. Let K be a countable field and $[u_1, \ldots, u_n] \in (K \setminus \{0\})^*$. We define the function $\phi^2_{[u_1,\ldots,u_n]}(P) \in \{0, 1, 2, 3\} \times \mathbb{N}$ for all $P \subseteq K$ with $1 \in P$ and the property $\forall_i \exists_j u_i \not\leq_P u_j$:

1. Search for some $i \leq n$ such that $u_i \leq_P u_j$ for all $j < n$. If such an i does not exists, return $\phi^2_{[u_1,\ldots,u_{n-1}]}(P)$. Otherwise, take this i.

2. Check if $u_n \not\leq_P u_i$. If yes, return $(3, [\![u_i u_n^{-1}, u_n u_i^{-1}, 1]\!])$.

3. Else, take k with $u_n \not\leq_P u_k$ and return

$$(2, [\![\{u_i u_n^{-1}, u_k u_i^{-1}\}, ([u_i u_n^{-1}, u_k u_i^{-1}], X - u_k u_i^{-1} u_i u_n^{-1}), u_k u_n^{-1}]\!])$$

Lemma 3.6.22. Assume we are in the situation of Algorithm 3.6.21, then the function $\phi^2_{[u_1,\ldots,u_n]}$ is a partial Krull functional (w.r.t. any $F \subseteq K$ and $r \in K$) defined on all P with $1 \in P$ and $\forall_{i \leq n} \exists_{j \leq n} u_i \not\leq_P u_j$.

Proof. Let $u_1, \ldots, u_n$ and P be given as above. We show that the property of a Krull functional for P is fulfilled. The proof is done by induction over n and following the steps of Algorithm 3.6.21. If the property of Step 1 is fulfilled, we directly use the induction hypothesis. Note that the property of Step 1 P is in the domain of $\phi^2_{[u_1,\ldots,u_{n-1}]}$. For the next steps let i be given with $u_i \leq_P u_j$ for all $j < n$. By the

property of P we must have $u_i \not\leq_P u_n$. If in Step 2 $u_n \not\leq_P u_i$ then neither $u_i u_n^{-1} \in P$ nor $u_i^{-1} u_n \in P$. But by assumption $1 \in P$ and so P is not prime and the output in this case is an evidence of this fact. In Step 3 we may assume $u_n \leq_P u_i$. Note that k with $u_n \not\leq_P u_k$ exists by the property of P. By $1 \in P$ we have $u_n \leq_P u_n$ and hence $k < n$. By the property of i in Step 1, it follows $u_i \leq_P u_k$. Together, $u_k u_i^{-1} \in P$, $u_i u_n^{-1} \in P$ but $u_k u_n^{-1} \notin P$. Therefore, P is not a $\rhd$-ideal and the output also fulfils the property of a Krull functional. □

Algorithm 3.6.23. Let K be a countable field, $E \subseteq K$ be a subring, $a_1, \ldots, a_m$, $b_1, \ldots, b_n \in K \setminus \{0\}$ and $c_k := \sum_{i+j=k} a_i b_j$ for all k. For fixed $0 \leq k \leq m$ and $0 \leq l \leq n$ we define a function $\phi_{[a_1,\ldots,a_m],[b_1,\ldots,b_n]}(P)$ for $P \in 2^K$ as follows:

1. Check if $c_i \notin P$ for some $i \in \{0, \ldots, m+n\}$. If yes, return $(0, c_k)$.

2. Check if $1 \notin P$. If yes, return $(2, [\![\varnothing, ([], X - 1), 1]\!])$.

3. Check if $a_k b_l \in P$. If yes, return $(1, 0)$.

4. Search for the maximal i_0 such that $a_{i_0} \leq_P a_i$ for all i. If such an i_0 does not exists, return $\phi^2_{[a_0,\ldots,a_m]}(P)$.

5. Search for the maximal j_0 such that $b_{j_0} \leq_P b_j$ for all j. If such a j_0 does not exist, return $\phi^2_{[b_0,\ldots,b_n]}(P)$.

6. Check if $a_{i_0} b_{j_0} \in P$. If yes,

   (a) check if $a_{i_0}^{-1} b_{j_0}^{-1} a_k b_l \in P$. If yes, return

   $$(2, [\![\{a_{i_0}^{-1} b_{j_0}^{-1} a_k b_l, a_{i_0} b_{j_0}\}, ([a_{i_0}^{-1} b_{j_0}^{-1} a_k b_l, a_{i_0} b_{j_0}], X - a_k b_l), a_k b_l]\!]).$$

   (b) Otherwise, return

   $$(2, [\![\{a_k a_{i_0}^{-1}, b_l b_{j_0}^{-1}\}, ([a_k a_{i_0}^{-1}, b_l b_{j_0}^{-1}], X - a_{i_0}^{-1} b_{j_0}^{-1} a_k b_l), a_{i_0}^{-1} b_{j_0}^{-1} a_k b_l]\!]).$$

7. Check if $c_{i_0+j_0} \neq 0$ and $a_{i_0} b_{j_0} c_{i_0+j_0}^{-1} \in P$. If yes, return

   $$(2, [\![\{c_{i_0+j_0}, a_{i_0} b_{j_0} c_{i_0+j_0}^{-1}\}, ([c_{i_0+j_0}, a_{i_0} b_{j_0} c_{i_0+j_0}^{-1}], X - a_{i_0} b_{j_0}), a_{i_0} b_{j_0}]\!]).$$

8. Check if $-a_{i_0} b_{j_0} a_i^{-1} b_j^{-1} \in P$ for some i, j with $i + j = i_0 + j_0$ and either $i > i_0$ or $j > j_0$. If yes, make the following case distinction:

(a) If $i > i_0$, take i' with $a_i \not\leq_P a_{i'}$ (which exists by Step 4) and return

$$(2, [\![\{\vec{\alpha}\}, ([\vec{\alpha}], X - a_{i'}a_i^{-1}), a_{i'}a_i^{-1}]\!]),$$

where $\vec{\alpha}$ is a syntactical abbreviation for $-a_{i_0}b_{j_0}a_i^{-1}b_j^{-1}, a_{i'}a_{i_0}^{-1}, b_j b_{j_0}^{-1}$.

(b) If $j > j_0$, take j' with $b_j \not\leq_P b_{j'}$ (which exists by Step 5) and return

$$(2, [\![\{\vec{\beta}\}, ([\vec{\beta}], X - b_{j'}b_j^{-1}), b_{j'}b_j^{-1}]\!]),$$

where $\vec{\beta}$ is a syntactical abbreviation for $-a_{i_0}b_{j_0}a_i^{-1}b_j^{-1}, a_i a_{i_0}^{-1}, b_{j'}b_{j_0}^{-1}$.

9. Return $\phi_s^1(P)$ where s is a list with contains exactly the elements $c_{i_0+j_0}a_{i_0}^{-1}b_{j_0}^{-1}$ and $-a_i b_j a_{i_0}^{-1}b_{j_0}^{-1}$ with either $i > i_0$ or $j > j_0$ in any order.

Lemma 3.6.24. In the situation of Algorithm 3.6.23 the function $\psi_{F,r}(P) := \phi_{[a_1,\ldots,a_m],[b_1,\ldots,b_n]}(P)$ is a Krull functional w.r.t. $F := \{c_0, \ldots, c_{n+m}\}$ and $r := a_k b_l$.

Proof. We check each step in turn. The first three cases are clear. For Step 4 and Step 5 we use Lemma 3.6.22. That P is in the domain of $\phi^2_{[a_0,\ldots,a_m]}$ and $\phi^2_{[b_0,\ldots,b_n]}$, respectively, follows by the failure of Step 2 (i.e. $1 \in P$) and since the non-existence of a maximal i_0 means that there is no i with $a_i \leq_P a_j$ for all j, and similarly for j_0. From Step 6 onwards, we take i_0 and j_0 with the property that $a_{i_0} \leq_P a_i$ for all i and for $i > i_0$ we have $a_i \not\leq_P a_i'$ for some i', and similarly for j_0.

The two cases of Step 6 follow directly by definition, since for (b) we have $\{a_k a_{i_0}^{-1}, b_l b_{j_0}^{-1}\} \subseteq P$ by the property of i_0 and j_0. Step 7 is also straightforward. Hence, from now on we have $a_{i_0}b_{j_0} \notin P$. For Step 8, in both cases, we use the maximality of i_0 and j_0 as well as that $a_{i_0} \leq_P a_{i'}$ and $b_{j_0} \leq_P b_j$.

For Step 9, we have

$$c_{i_0+j_0}a_{i_0}^{-1}b_{j_0}^{-1} - \sum_{\substack{i+j=i_0+j_0\\ i>i_0}} a_i b_j a_{i_0}^{-1}b_{j_0}^{-1} - \sum_{\substack{i+j=i_0+j_0\\ j>j_0}} a_i b_j a_{i_0}^{-1}b_{j_0}^{-1} = 1$$

as in the proof of Proposition 3.6.16. Hence, we use Lemma 3.6.19, where P is in the domain of ϕ_s^1 because of the failure of Step 2, Step 7 and Step 8. □

Corollary 3.6.25 (Algorithmic version of Kronecker's Theorem). Let K be a countable field and $E \subseteq K$ be a subring. We take $a_0, \ldots, a_m, b_0, \ldots, b_n \in K \setminus \{0\}$ and define $c_k := \sum_{i+j=k} a_i b_j$ for all $k \in \{0, \ldots, m+n\}$. We fix some $k \in \{0, \ldots, m\}$ and $l \in \{0, \ldots, n\}$. Suppose that Algorithm 3.4.14 is instantiated on the cover structure (ι, τ, η) from Lemma 3.6.12 and the Krull functional $\psi_{F,r}(P)$ with $F := \{c_0, \ldots, c_{n+m}\}$ and $r := a_k b_l$ defined in Algorithm 3.6.23. Then the algorithm terminates in some state π_K and for $g_{\iota,\tau,\eta,\psi_{F,r}}(\pi_K) = (A, (\vec{a}, p))$ we have $\{\vec{a}\} \subseteq A \subseteq \{c_0, \ldots, c_{n+m}\}$, $p \in E[\vec{a}][X]$ and $p(a_k b_l) = 0$.

Proof. This follows directly by Corollary 3.6.13 and Lemma 3.6.24. □

Remark 3.6.26. As mentioned at the beginning of this section, our approach is quite similar to [56] and indeed this is the main source of this application. Proposition 3.6.16 is similar to Theorem 6 of [56], where the proof of this theorem is made constructive by using entailment relations instead of valuation rings and after the proof of Theorem 6 it is shown how to derive an algorithm from the proof. In [56] the assumption that the underlining K is countable is not used. But if K is an arbitrary field and $a_1, \ldots, a_m, b_1, \ldots, b_n \in K \setminus \{0\}$ are given, we can w.l.o.g. assume that K is the field which is generated by the elements $a_1, \ldots, a_m, b_1, \ldots, b_n$. At least classically this field is countable and therefore we can apply Corollary 3.6.25. However, our algorithm always depends on some coding of the corresponding ring whereas the algorithm in [56] does not.

3.7 Case study: ordered fields

Motivation 3.7.1. The version of Theorem 3.4.3 in this section, is known as Theorem of Artin-Schreier and categorised into constructive real algebra. A version of it was formulated by Artin and Schreier [10], which Artin used to prove Hilbert's 17^{th} problem [9]. Our approach in this chapter is based on [144].

Definition 3.7.2. Let K be a field with $\text{char}(K) \neq 2$. A *preorder* of K is a subset $S \subseteq K$ which contains all squares and is closed under addition and multiplication, i.e.

$$\{x^2 \mid x \in K\} \subseteq S, \qquad S + S \subseteq S, \qquad S \cdot S \subseteq S.$$

In particular, the set of all sums of squares is the smallest preorder of K. We denote this set by $\sum K^2 := \{x_1^2 + \cdots + x_n^2 \mid n \in \mathbb{N},\ x_1, \ldots, x_n \in K,\ x \in \mathbb{N}\}$.

We call a preorder S a *linear order* (or just order) if in addition either $x \in S$ or $-x \in S$ for all $x \in K$.

Notation 3.7.3. Let K be a field and $S \subseteq K$ be a preorder. We define the relation $\leq_S$ by

$$x \leq_S y :\Leftrightarrow y - x \in S.$$

If S is clear from the context we also write $\leq$.

Lemma 3.7.4. A subset $S \subseteq K$ of a field K with $\text{char}(K) \neq 2$ is a preorder if and only if the relation $\leq_S$ is a preorder (i.e. reflexive and transitive) on K and satisfies

$$x \leq_S y \Rightarrow x + z \leq_S y + z, \qquad x \leq_S y \wedge 0 \leq_S x \Rightarrow ax \leq_S bx.$$

A preorder S is an order if and only if $\leq_S$ is total.

Proof. Trivial. □

Definition 3.7.5. For any preorder S on a field K with $\mathrm{char}(K) \neq 2$ and subset $U \subseteq K$, the set $S\langle U\rangle$ is the preorder generated by U and consist of all elements in $x \in K$ for which there is a finite $U_0 \subseteq S$ such that

$$x = \sum_{V \subseteq U_0} \lambda_V \prod V,$$

where $\prod\{v_1, \ldots, v_l\} := \prod_{i=1}^{l} v_i$ and $\lambda_V \in S$.

Notation 3.7.6. Let A, B be sets. We denote $A \triangle B := (A \cup B) \setminus (A \cap B)$ and for any x we define

$$\chi_A(x) := \begin{cases} 1 \text{ if } x \in A \\ 0 \text{ else.} \end{cases}$$

Lemma 3.7.7. In the situation of Definition 3.7.5, $S\langle U\rangle$ is indeed a preorder.

Proof. Since $\prod \varnothing = 1$, we have $x^2 = x^2 \prod \varnothing \in S\langle U\rangle$ for all $x \in K$.

Let $x = \sum_{V \subseteq U_0} \lambda_V \prod V$ and $y = \sum_{V \subseteq U_0} \sigma_V \prod V$. Then

$$x + y = \sum_{V \subseteq U_0 \cup U_1} \left(\lambda_V \chi_{\mathcal{P}(U_0)}(V) + \sigma_V \chi_{\mathcal{P}(U_1)}(V)\right) \prod V \in S\langle U\rangle$$

and

$$\begin{aligned} xy &= \sum_{V \subseteq U_0} \sum_{W \subseteq U_1} \lambda_V \sigma_W \prod V \prod W \\ &= \sum_{V \subseteq U_0} \sum_{W \subseteq U_1} \lambda_V \sigma_W \left(\prod V \cap W\right)^2 \prod (V \triangle W) \\ &= \sum_{X \subseteq U_0 \cup U_1} \left(\sum_{\substack{V \subseteq U_0, W \subseteq U_1 \\ V \triangle W = X}} \lambda_V \sigma_W \left(\prod V \cap W\right)^2 \right) \prod X \in S\langle U\rangle. \end{aligned}$$

□

Theorem 3.7.8 (Artin-Schreier). Let K be a field with $\mathrm{char}(K) \neq 2$, $S \subseteq K$ be a preorder and $F \subseteq K$. Then

$$\bigcap \{U \subseteq K \mid S \subseteq U, F \subseteq U \text{ and } U \text{ is an order of } K\} \subseteq S\langle F\rangle.$$

Proof. We define $\rhd$ and K by $U \rhd x :\Leftrightarrow x \in S\langle U\rangle$ and $\circ$ by the addition of K and use Theorem 3.4.3. Reflexivity, transitivity and encoding follows similar to the proof of Lemma 3.7.12. Therefore, we do not give the details. A proof without the computational consideration can be found in [144, Section 4.6] or [86, Kapitel I]. □

Motivation 3.7.9. We start with an algorithmic formulation of the theorem above by defining the covering and by proving the constructive existence of a cover structure.

Definition 3.7.10. Let K be a countable field with $\mathrm{char}(K) \neq 2$ and $S \subseteq K$ be a preorder on K which is countable. In this section we define the witness type by $S^{(\mathcal{P}_{fin}(K))}$, which is the set of the families $(\lambda_V)_{V\in\mathcal{P}_{fin}(K)}$ with $\lambda_V = 0$ for all but finitely many $V \in \mathcal{P}_{fin}(K)$. For $A \in \mathcal{P}_{fin}(K)$, $(\lambda_V)_V \in S^{(\mathcal{P}_{fin}(K))}$ and $x \in K$ we define the covering $\rhd$ by

$$A \rhd_{(\lambda_V)_V} x :\Leftrightarrow \sum_{U\subseteq A} \lambda_U \prod U = x.$$

Furthermore, we define the operator $\circ$ on K by the addition in K.

Proposition 3.7.11. In the situation of Definition 3.7.10 the $\rhd$-ideals are exactly the preorders of K which contain S and the prime $\rhd$-ideals are exactly the linear orders of K which contains S.

Proof. First, let I be an $\rhd$-ideal with $S \subseteq I$. We show that I is a preorder: If $x \in K$ then $x^2 \in S$, and therefore $\varnothing \rhd_{(\sigma_U)_U} x^2$ for $\sigma_U = x^2\chi_{\{\varnothing\}}(U)$. Since I is an $\rhd$-ideal, it follows $x^2 \in I$. Now, let $x, y \in I$ be given. Then we have $A \rhd_{(\sigma_U)_U} x$ and $B \rhd_{(\lambda_U)_U} y$ for some finite $A, B \subseteq I$ and $(\sigma_U)_U, (\lambda_U)_U \in S^{(\mathcal{P}_{fin}(K))}$. In particular, we have

$$x = \sum_{U\subseteq A} \sigma_U \prod U, \qquad y = \sum_{U\subseteq B} \lambda_U \prod U.$$

Then

$$x + y = \sum_{U\subseteq A\cup B} \left(\sigma_U \chi_{\mathcal{P}(A)}(U) + \lambda_U \chi_{\mathcal{P}(B)}(U)\right) \prod U$$

and

$$xy = \sum_{W\subseteq A\cup B} \left(\sum_{\substack{U\subseteq A, V\subseteq B \\ U\triangle V = W}} \lambda_U \sigma_V \left(\prod U \cap V\right)^2 \right) \prod W.$$

Therefore, $A \cup B \rhd x+y$ and $A \cup B \rhd xy$. Since I is an $\rhd$-ideal, we have $a+b, ab \in I$ and I is a preorder. If furthermore I is a prime ideal, we have $x + (-x) = 0 \in I$ and therefore either $x \in I$ or $-x \in I$ and I is an order.

For the other direction, let I be a preorder with $S \subseteq I$ and assume $A \rhd_{(\sigma_U)_U} x$ for some finite $A \subseteq I$, $(\sigma_U)_U \in S^{(\mathcal{P}_{fin}(K))}$ and $x \in K$, then

$$x = \sum_{U \subseteq A} \sigma_U \prod U.$$

Since $A \subseteq I$, $S \subseteq I$ and I is a preorder, all $\sigma_U \prod U$ for $U \subseteq A$ are in I and so $x \in I$. Hence, I is an ideal. Furthermore, if I is an order and $x + y \in I$ for some $x, y \in K$ then either $x \in I$ or $-x \in I$. In the last case we have $y \in I$ because of $x + y \in I$. Thus, either $x \in I$ or $y \in I$, which means that I is prime. □

Lemma 3.7.12. In the situation of Definition 3.7.10, the relation $\rhd$ is a Σ^0_1-covering and together with $\circ$ it has a computable cover structure (ι, τ, η).

Proof. Since S and K are countable, also $S^{(\mathcal{P}_{fin}(K))}$ is countable and can therefore be coded as base type.

It remains to deal with reflexivity, transitivity and encoding:

– Obviously $\{x\} \rhd_{\left(\chi_{\{\{x\}\}}(V)\right)_V} x$. Therefore, ι is given by $\iota(x) := \left(\chi_{\{\{x\}\}}(V)\right)_V$

– For transitivity, we assume $A \rhd_{(\sigma_V)_V} x$ and $B, x \rhd_{(\lambda_V)_V} y$, in particular

$$x = \sum_{U \subseteq A} \sigma_U \prod U \qquad \text{and} \qquad y = \sum_{V \subseteq B \cup \{x\}} \lambda_V \prod V.$$

We calculate:

$$\begin{aligned}
y &= \sum_{V \subseteq B \setminus \{x\}} \lambda_V \prod V + \sum_{V \subseteq B \setminus \{x\}} \lambda_{V \cup \{x\}} x \prod V \\
&= \sum_{V \subseteq B \setminus \{x\}} \lambda_V \prod V + \sum_{V \subseteq B \setminus \{x\}} \lambda_{V \cup \{x\}} \sum_{U \subseteq A} \sigma_U \prod U \prod V \\
&= \sum_{V \subseteq B \setminus \{x\}} \lambda_V \prod V + \sum_{\substack{V \subseteq B \setminus \{x\} \\ U \subseteq A}} \lambda_{V \cup \{x\}} \sigma_U \left(\prod U \cap V\right)^2 \prod U \vartriangle V \\
&= \sum_{W \subseteq A \cup (B \setminus \{x\})} \left(\lambda_W \chi_{\mathcal{P}(B \setminus \{x\})}(W) + \sum_{\substack{U \subseteq A, V \subseteq B \setminus \{x\} \\ U \vartriangle V = W}} \lambda_{V \cup \{x\}} \sigma_U \left(\prod U \cap V\right)^2 \right) \prod W
\end{aligned}$$

Therefore, τ is given by

$$\tau(A, B, x, y, (\sigma_V)_V, (\lambda_U)_U) := (\kappa_W)_W,$$

where

$$\kappa_W := \lambda_W \chi_{\mathcal{P}(B\setminus\{x\})}(W) + \sum_{\substack{U\subseteq A, V\subseteq B\setminus\{x\} \\ U\triangle V = W}} \lambda_{V\cup\{x\}}\sigma_U(\prod U\cap V)^2$$

for $W \subseteq A \cup (B \setminus \{x\})$ and $\kappa_W := 0$ else.

– It remains to consider encoding: Let $A, x \rhd_{(\sigma_U)_U} z$ and $B, y \rhd_{(\lambda_V)_V} z$ be given. Our goal is to compute $(\kappa_W)_W$ with $A\cup B, x+y \rhd_{(\kappa_W)_W} z$. We define $C := A\cup B$. By modifying $(\sigma_U)_U$ and $(\lambda_V)_V$ we may assume $\sigma_U = 0$ for $U \not\subseteq A \cup \{x\}$ and $\lambda_V = 0$ for $V \not\subseteq B \cup \{y\}$. Furthermore, we assume $x, y \notin C$ because otherwise it is trivial. Then because of our assumptions it follows

$$\begin{aligned} z &= \sum_{U\subseteq C} \sigma_U \prod U + \sum_{U\subseteq C} \sigma_{U\cup\{x\}} x \prod U \\ &= \sum_{U\subseteq C} \lambda_U \prod U + \sum_{U\subseteq C} \lambda_{U\cup\{y\}} y \prod U. \end{aligned}$$

We define

$$\Gamma := \sum_{U\subseteq C} \left(\sigma_{U\cup\{x\}} + \lambda_{U\cup\{y\}}\right) \prod U$$

and first consider the case $\Gamma \neq 0$. Hence, we compute

$$\begin{aligned} \Gamma z = &\sum_{U\subseteq C} \sigma_{U\cup\{x\}} \prod U \left(\sum_{V\subseteq C} \lambda_V \prod V + \sum_{V\subseteq C} \lambda_{V\cup\{y\}} y \prod V\right) + \\ &\sum_{U\subseteq C} \lambda_{U\cup\{y\}} \prod U \left(\sum_{V\subseteq C} \sigma_V \prod V + \sum_{V\subseteq C} \sigma_{V\cup\{x\}} x \prod V\right) \\ = &\sum_{U,V\subseteq C} \left(\sigma_{U\cup\{x\}}\lambda_V + \sigma_U\lambda_{V\cup\{y\}} + (x+y)\sigma_{U\cup\{x\}}\lambda_{V\cup\{y\}}\right) \prod U \prod V, \end{aligned}$$

and therefore

$$\begin{aligned} z = &\sum_{U,V,W\subseteq C} \mu_{U,V,W} \prod U \prod V \prod W \\ &+ \sum_{U,V,W\subseteq C} \nu_{U,V,W}(x+y) \prod U \prod V \prod W, \end{aligned}$$

where

$$\mu_{U,V,W} := \Gamma^{-2}\left(\sigma_{W\cup\{x\}} + \lambda_{W\cup\{y\}}\right)\left(\sigma_{U\cup\{x\}}\lambda_V + \sigma_U\lambda_{V\cup\{y\}}\right),$$
$$\nu_{U,V,W} := \Gamma^{-2}\left(\sigma_{W\cup\{x\}} + \lambda_{W\cup\{y\}}\right)\sigma_{U\cup\{x\}}\lambda_{V\cup\{y\}}.$$

We further obtain

$$z = \sum_{X\subseteq C}\left(\sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X}} \mu_{U,V,W}(g(U,V,W))^2\right)\prod X + \sum_{X\subseteq C}\left(\sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X}} \nu_{U,V,W}(g(U,V,W))^2\right)(x+y)\prod X,$$

where

$$g(U,V,W) := \prod((U\cup V)\setminus W)\prod((U\cup W)\setminus V)\prod((V\cup U)\setminus W)\prod(U\cap V\cap W).$$

In order to cover the case $x+y\in C$, we continue with our calculation

$$\begin{aligned} z = &\sum_{X\subseteq C\setminus\{x+y\}}\sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X}} \mu_{U,V,W}(g(U,V,W))^2\prod X + \\ &\sum_{X\subseteq C\setminus\{x+y\}}\sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X\cup\{x+y\}}} \mu_{U,V,W}(g(U,V,W))^2(x+y)\prod X + \\ &\sum_{X\subseteq C\setminus\{x+y\}}\sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X}} \nu_{U,V,W}(g(U,V,W))^2(x+y)\prod X + \\ &\sum_{X\subseteq C\setminus\{x+y\}}\sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X\cup\{x+y\}}} \nu_{U,V,W}(g(U,V,W))^2(x+y)^2\prod X \end{aligned}$$

$$
\begin{aligned}
= \sum_{X\subseteq C\setminus\{x+y\}} \Bigg(& \sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X}} \mu_{U,V,W}(g(U,V,W))^2 + \\
& \sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X\cup\{x+y\}}} \nu_{U,V,W}(g(U,V,W))^2(x+y)^2 \Bigg) \prod X + \\
\sum_{X\subseteq C\setminus\{x+y\}} \Bigg(& \sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X\cup\{x+y\}}} \mu_{U,V,W}(g(U,V,W))^2 + \\
& \sum_{\substack{U,V,W\subseteq C\\ U\triangle V\triangle W=X}} \nu_{U,V,W}(g(U,V,W))^2 \Bigg) (x+y) \prod X.
\end{aligned}
$$

From this equation we get $(\kappa_X)_X$ with $C, x+y \rhd_{(\kappa_X)_X} z$ in the case $\Gamma \neq 0$. It remains to consider the case $\Gamma = 0$, i.e.

$$\sum_{U\subseteq C} \left(\sigma_{U\cup\{x\}} + \lambda_{U\cup\{y\}}\right) \prod U = 0.$$

We again consider two cases: If

$$\Delta := \sum_{U\subseteq C} \sigma_{U\cup\{x\}} \prod U = 0$$

then

$$z = \sum_{U\subseteq C} \sigma_U \prod U$$

and this case is completed. Hence, we consider the case $\Delta \neq 0$. Here we have

$$0 \neq -\Delta = \sum_{U\subseteq C} \lambda_{U\cup\{y\}} \prod U$$

which implies

$$
\begin{aligned}
-1 = & \Delta^{-2} \sum_{U\subseteq C} \sigma_{U\cup\{x\}} \prod U \sum_{U\subseteq C} \lambda_{U\cup\{y\}} \prod U \\
& \sum_{W\subseteq C} \sum_{\substack{U,V\subseteq C\\ U\triangle V=W}} \Delta^{-2} \sigma_{U\cup\{x\}} \lambda_{V\cup\{y\}} \left(\prod U\cap V\right)^2 \prod W.
\end{aligned}
$$

Together with $\mathrm{char}(K) \neq 2$ we have

$$\begin{aligned} z &= \left(\frac{z+1}{2}\right)^2 + (-1)\left(\frac{z-1}{2}\right)^2 \\ &= \sum_{W \subseteq C} \left(\frac{z+1}{2}\right)^2 \chi_{\{\varnothing\}}(W) \prod W + \\ &\quad \sum_{W \subseteq C} \sum_{\substack{U,V \subseteq C \\ U \triangle V = W}} \Delta^{-2} \sigma_{U \cup \{x\}} \lambda_{V \cup \{y\}} \left(\frac{z-1}{2}\right)^2 \left(\prod U \cap V\right)^2 \prod W. \end{aligned}$$

Therefore, we are done in this case. □

Corollary 3.7.13. In the situation of Definition 3.7.10, suppose that for $F \subseteq K$ and $r \in K$ we have a Krull functional ψ w.r.t. $\rhd$ and $\circ$. Instantiating Algorithm 3.4.14 on the cover structure of Lemma 3.7.12 and the Krull functional ψ, the algorithm terminates in some state π_k. Let $g_{\iota,\tau,\eta,\psi}(\pi_k) = (A, (\lambda_U)_U)$. Then $A \subseteq F$, $(\lambda_U)_U \in S^{(\mathcal{P}_{fin}(K))}$ and

$$\sum_{U \subseteq A_k} \lambda_U \prod U = r.$$

In particular, $(\lambda_U)_U$ is a witness for $r \in S\langle F \rangle$.

Proof. Follows directly from Theorem 3.4.15 and Lemma 3.7.12. □

Remark 3.7.14. The version of the universal Krull-Lindenbaum lemma in this section was used by Artin to prove Hilbert's 17$^{\text{th}}$ problem [9]. Actually Hilbert's 17$^{\text{th}}$ problem could be an example of this case study. Unfortunately, turning Artin's proof into a Krull functinal similar to what we have done in the examples of the last two sections, is highly complex as the proof uses constructions like the real closure of an ordered field. However, it is presumably possible. A possible initial point could be backtracking the proofs in [116] together with Artin's proof of Hilbert's 17$^{\text{th}}$ problem [9]. In Section 4.5 there is an outlook how things can get complex when applying Algorithm 3.3.6, and we assume that a transformation of Artin's proof into a Krull functional is even more complicated. Therefore, we conclude this section without an example. If the reader is interested in a constructive version of Hilbert's 17$^{\text{th}}$ problem, we refer to [61].

3.8 Minor case studies

Motivation 3.8.1. We conclude with a section about three shorter case studies from [144]. In the first two case studies the proofs are short, and therefore also the corresponding cover structures are quite simple. This is due to the fact that the structures in the two case studied are too simple. Hence, we assume that the deduced version of the universal Krull-Lindenbaum lemma is not very useful for further applications as one can just replace the usage of the universal Krull-Lindenbaum lemma by the respective proofs. However, these case studies reveal the many diverse applications of the theory in this chapter. The last case study about filters in commutative rings is probably more useful.

3.8.1 Complete theories

Motivation 3.8.2. The following example is a version of Lindenbaum's lemma[6], known from model theory. It was also considered in [39] by Francesco Ciraulo, Davide Rinaldi and Peter Schuster and revisited in [73] by Giulio Fellin, Peter Schuster and Daniel Wessel. However, it is well-known that Lindenbaum's lemma can be proven constructively if the underlying language is countable. As countability is a necessary assumption in our case, we do not really get a new result in this section.

Definition 3.8.3. We fix some language $\mathcal{L}$ of first order logic and a proof calculus like Genzen's calculus of natural deduction. Let $\mathcal{S}$ be the set of all sentences in the language $\mathcal{L}$. By $\vdash$ we denote the derivability in classical logic. For $A \subseteq \mathcal{S}$ and $\phi \in \mathcal{S}$ we define $A \rhd \phi$ by $A \vdash \phi$. In particular, $A \rhd \phi$ means that there is some proof tree t with assumptions in A and root ϕ. The operator $\circ$ on S is the disjunction $\vee$, i.e. $\phi \circ \psi := \phi \vee \psi$ for all $\phi, \psi \in \mathcal{S}$.

A subset $\Gamma \subseteq \mathcal{S}$ is called *theory* if it is closed under derivability (i.e. if it is an $\rhd$-ideal). The theory Γ is called *complete* if

$$\phi \in \Gamma \text{ or } \neg\phi \in \Gamma$$

for all $\phi \in \mathcal{S}$.

Remark 3.8.4. Note that we have not fixed the proof calculus since any of the popular proof calculi can be used. However, the concrete definition of the cover structure does depend on the proof calculus. Therefore, in this case we use the

[6] As Lindenbaum's lemma in a special case, "Lindenbaum" is part of the name "universal Krull-Lindenbaum lemma".

calculus of *natural deduction* and to shorten the notation, we will use *proof terms* or also called *derivation terms*. The calculus of natural deduction and proof terms are defined in Section 1.1 and 1.2 of [159].

Lemma 3.8.5. In the situation of Definition 3.8.3, the complete theories are exactly the prime $\rhd$-ideals.

Proof. Let Γ be a complete theory and $\phi \vee \psi \in \Gamma$. Because of completeness, either $\phi \in \Gamma$ or $\neg\phi \in \Gamma$. In the first case we are done. In the second case, we have $\phi \vee \psi, \neg\phi \vdash \psi$ and therefore $\psi \in \Gamma$.

For the other direction, assume that Γ is a prime $\rhd$-ideal and let $\phi \in \mathcal{S}$ be given. Γ is a theorie and $\varnothing \vdash \phi \vee \neg\phi$, and so $\phi \vee \neg\phi \in \Gamma$. Since Γ is prime, we have $\phi \in \Gamma$ or $\neg\phi \in \Gamma$. □

Theorem 3.8.6. In the situation of Definition 3.8.3. Let $F \subseteq \mathcal{S}$ be given. With $\overline{F}$ we denote the $\vdash$-closure of F, in particular $\overline{F} := \{\phi \in \mathcal{S} \mid F \vdash \phi\}$. Then

$$\bigcap\{\Gamma \subseteq \mathcal{S} \mid F \subseteq \Gamma \text{ and } \Gamma \text{ is a complete theory}\} \subseteq \overline{F}$$

Proof. This follows from Theorem 3.4.3. The proof of reflexivity, transitivity and encoding is quite similar to the proof of the next lemma. But it is also given in [73, Proposition 2] . □

Lemma 3.8.7. In the situation of Definition 3.8.3 we additionally assume that $\mathcal{L}$ and hence $\mathcal{S}$ is countable. Then $\rhd$ is a Σ^0_1-covering and together with $\circ$ has a computable cover structure (ι, τ, η).

Proof. As proof calculus we use Genzen's calculus of natural deduction but one could also use an equivalent proof calculus and define the cover structure with it.

$A \rhd \phi$ holds if there is a proof tree t with assumptions in A and root ϕ. Therefore, $\rhd$ is a Σ^0_1-covering. We define the cover structure step by step:

- For $\phi \in \mathcal{S}$ the proof which only consists of the assumption ϕ is a witness for $\{\phi\} \rhd \phi$. In the notation of proof terms we have $\iota(\phi) := u^\phi$, where u is a new assumption variable.

- Assume $A \rhd_s \phi$ and $B, \phi \rhd_t \psi$. To get a proof tree which witnesses $A \cup B \rhd \psi$, we take the proof tree t and replace each assumption of the formula ϕ by the proof tree s. In the notation of proof terms: Let $t = t(u_1^\phi, \ldots, u_l^\phi)$ where $u_1, \ldots, u_l$ are exactly the assumption variable in t with type ϕ, and define $\tau(A, B, \phi, \psi, s, t) := t(s, \ldots, s)$. To avoid variable collisions we may have to rename some variables before the substitution.

– Assume $A, \phi \rhd_s \chi$ and $B, \psi \rhd_t \chi$. Using the implication introduction we get proof trees s_1 and t_1 with $A \rhd_{s_1} \phi \to \chi$ and $B \rhd_{t_1} \psi \to \chi$. After assuming $A \vee B$ and using the disjunction elimination rule we get a proof tree of $A \cup B, \phi \vee \psi \rhd \chi$. To make things clear, we use the notation of proof terms: let $s = s(u_1^\phi, \ldots, u_l^\phi)$ where $u_1, \ldots, u_l$ are exactly the assumption variables in s of the formula ϕ. We define $\tilde{s} := \tilde{s}(u) := s(u, \ldots, u)$ where u^ϕ is a new assumption variable of the formula ϕ. We define analogously $\tilde{t} := \tilde{t}(v)$, where v is a new assumption variable of formula ψ. Furthermore, we denote the disjunction elimination rule by $\vee^-$. Then $\eta(A, B, \phi, \psi, \chi, s, t) := \lambda_w(\vee^- w^{\phi \vee \psi}(\lambda_u \tilde{s})(\lambda_v \tilde{t}))$, where w is a new assumption variable of the formula $\phi \vee \psi$.

□

Corollary 3.8.8. In the situation of Definition 3.8.3 with countable $\mathcal{L}$, suppose for $\Gamma \subseteq \mathcal{S}$ and $\phi \in \mathcal{S}$ a Krull functional Ψ w.r.t. $\rhd$ and $\circ$ is given. Instantiating Algorithm 3.4.14 on the cover structure of Lemma 3.7.12 and the Krull functional Ψ, the algorithm terminates in some state π_k. Let $g_{\iota,\tau,\eta,\Psi}(\pi_k) = (A, t)$. Then $A \subseteq \Gamma$ and t is a proof of ϕ with assumptions in A.

Proof. Follows directly from Theorem 3.4.15 and Lemma 3.8.7. □

3.8.2 Distributive lattices

Motivation 3.8.9. Distributive lattices are algebraic objects which have a quite simple structure. They were also considered in [145, Section 4.2.3]. In our case, one could even say that they have too much structure to be interesting, but constructively they are very interesting like in the context of entailment relations [35, 50].

The theorems we consider in this section are a variant of the prime filter theorem or prime ideal theorem the for bounded distributive lattices, respectively.

Definition 3.8.10. A *distributive lattice* is given by a set L and two associative and commutative operators $\wedge$ and $\vee$ on L such that

$$\begin{aligned} a \vee (a \wedge b) = a = a \wedge (a \vee b) \\ a \vee a = a = a \wedge a \\ (a \vee b) \wedge c = a \wedge c \vee b \wedge c \\ a \wedge b \vee c = (a \vee c) \wedge (b \vee c) \end{aligned}$$

for all $a, b, c \in L$ hold, where $\wedge$ binds more strongly then $\vee$. We define $a \leq b$ by $a = a \wedge b$.

Since $\wedge$ and $\vee$ are commutative, we define $\bigwedge A$ and $\bigvee A$ for $A \in \mathcal{P}_{fin}(L) \setminus \{\varnothing\}$ in the canonical way.

Lemma 3.8.11. Assume we are in the situation of Definition 3.8.10, then $a \leq b$ is equivalent to $b = a \vee b$ and $\leq$ is a partial order on L.

Proof. Trivial. □

Notation 3.8.12. If there exists a least element of the partial order in the lemma above, we denote it by 0, and if there exists a greatest element, we denote it by 1. In particular, $0 \leq a \leq 1$ for all $a \in L$.

Definition 3.8.13. Let L be a distributive lattice and $U \subseteq L$ be a subset of L. The ideal (U) of a distributive lattice generated by U consists of all elements $a \in$ for which there are $u_1, \ldots, u_k \in U$ such that $a \leq u_1 \vee \cdots \vee u_k$. If L has a least element 0, the ideal generated by the empty set is $\{0\}$, and we define $\bigvee \varnothing := 0$. If $U = (U)$, we call U an *ideal* of the distributive lattices L. An ideal P is called *prime ideal* if $a \wedge b \in P$ implies $a \in P$ or $b \in P$.

The filter $\langle U \rangle$ generated by U consists of all elements $a \in L$ for which there are $u_1, \ldots, u_k \in U$ such that $u_1 \wedge \cdots \wedge u_k \leq a$. If L has a greatest element 1, the filter generated by the empty set is $\{1\}$, and we define $\bigwedge \varnothing := 1$. If $U = \langle U \rangle$, we call U a *filter* of the distributive lattices L. A filter P is called *prime filter* if $a \vee b \in P$ implies $a \in P$ or $b \in P$.

Theorem 3.8.14. Let L be a distributive latices with 0 and 1 and $F \subseteq L$. Then

$$\bigcap \{P \subseteq L \mid F \subseteq P \text{ and } P \text{ is prime ideal}\} = (F) \text{ and}$$
$$\bigcap \{P \subseteq L \mid F \subseteq P \text{ and } P \text{ is prime filter}\} = \langle F \rangle.$$

Proof. Follows from Theorem 3.4.3. $\rhd$, $\circ$, reflexivity, transitivity and encoding are similar as given in Definition 3.8.16 and Lemma 3.8.19. Therefore, we do not give the details here. □

Motivation 3.8.15. As we see, the notion of filters and ideals in distributive lattices are dual to each other. Therefore, in the following computational consideration, we restrict ourself to filters. Of course, the analogous statements also hold for filters instead of ideals.

Definition 3.8.16. Let L be a countable distributive lattice with minimal element 0. For finite $A \subseteq L$ and $a \in L$ we define the covering $\rhd$ by $A \rhd a :\Leftrightarrow a \in (A)$ and the operator $\circ$ by $\wedge$.

Remark 3.8.17. Assume we are in the situation of Definition 3.8.16. Then for each finite A, we have $A \rhd a$ if and only if $a \leq \bigvee A$. If the operator $\wedge, \vee$ and

$=$ (as a boolean valued function) are computable this statement does not need any evidence. So the witness type is a singleton $\{\varepsilon\}$, where $A \rhd_\varepsilon a$ just says that $a \leq \bigvee A$. Therefore, the functions ι, τ, η are trivial, and we only have to check the properties.

Of course, we could assume that the equality needs some witness, i.e. $x = y \Leftrightarrow \exists_t x =_t y$ for some decidable relation $=_.$. In this case the witness type would not be trivial, but then we also need some evidence for the properties in Definition 3.8.10, which makes things more complicate. Here we assume that everything is decidable to have a case study where the witness type is trivial.

Proposition 3.8.18. In the situation of Definition 3.8.16, the $\rhd$-ideals are exactly the filters of L and the prime $\rhd$-ideals are exactly the prime filters of L.

Proof. Trivial. □

Lemma 3.8.19. In the situation of Definition 3.8.16 let $\wedge, \vee$ and $=$ be completable, then $\rhd$ is a Σ^0_1-covering and $(\underline{\varepsilon}, \underline{\varepsilon}, \underline{\varepsilon})$ is a cover structure for $\rhd$ and $\circ$, where $\underline{\varepsilon}$ denotes the constant function with value ε.

Proof. That $\rhd$ is a Σ^0_1-covering was noted in the remark above. It remains to check the three properties.

- $a \leq a = \bigvee\{a\}$ for all $a \in L$ is trivial.
- Let $A \rhd a$ and $B, a \rhd b$ be given. Then $a = a \wedge \bigvee A$ and $b = b \wedge (a \vee \bigvee B)$. Hence, it follows

$$\begin{aligned} b =& b \wedge \left(\left(a \wedge \bigvee A\right) \vee \bigvee B\right) = b \wedge \left(a \vee \bigvee A\right) \wedge \bigvee (A \cup B) \\ =& b \wedge \bigvee (A \cup B) \end{aligned}$$

- Let $A, a \rhd c$ and $B, b \rhd c$ be given. Then $c = c \wedge (a \vee \bigvee A)$ and $c = c \wedge (b \vee \bigvee B)$. By the first axiom of distributive lattices we have

$$a \vee \bigvee A = \left(a \vee \bigvee A\right) \wedge \left(a \vee \bigvee (A \cup B)\right)$$

and analogous for B and b instead of A and a. Hence, it follows

$$\begin{aligned} c =& c \wedge \left(a \vee \bigvee A\right) = c \wedge \left(a \vee \bigvee A\right) \wedge \left(a \vee \bigvee (A \cup B)\right) \\ =& c \wedge \left(a \vee \bigvee (A \cup B)\right) = c \wedge \left(a \vee \bigvee (A \cup B)\right) \wedge \left(b \vee \bigvee (A \cup B)\right) \\ =& c \wedge \left(a \wedge b \vee \bigvee (A \cup B)\right), \end{aligned}$$

which implies $c \leq a \wedge b \vee \bigvee (A \cup B)$. □

Corollary 3.8.20. Assume we are in the situation of Definition 3.8.16 with computable $\wedge$ and $\vee$, and suppose that for $F \subseteq L$ and $r \in L$ a Krull functional ψ w.r.t. $\rhd$ is given. Instantiating Algorithm 3.4.14 on the cover structure $(\underline{\varepsilon}, \underline{\varepsilon}, \underline{\varepsilon})$ and the Krull functional ψ, the algorithm terminates in some state π_k. Let $g_{\underline{\varepsilon},\underline{\varepsilon},\underline{\varepsilon},\psi}(\pi_k) = (A, t)$, then $A \subseteq F$, $t = \varepsilon$ and $r \leq \bigvee A$.

Proof. Follows directly from Theorem 3.4.15 and Lemma 3.8.19. □

3.8.3 Filters in commutative rings

Motivation 3.8.21. As last case study we consider filters in commutative rings. This version of the universal Krull-Lindenbaum lemma is often called the prime filter theorem for commutative rings.

Definition 3.8.22. Let a commutative ring R be given. A subset $F \subseteq R$ is called a *filter* if $1 \in F$ and $\forall_{a,b \in R}(a, b \in F \Leftrightarrow ab \in F)$. For a subset $U \subseteq R$ we define the filter generated by U as follows:

$$\langle U \rangle = \left\{ x \in R \;\middle|\; \exists_{c \in R} \exists_{u_1,\ldots,u_k \in U} xc = \prod_{i=1}^{k} u_i \right\}.$$

A filter F is called a *prime filter* if $a + b \in F$ implies $a \in F$ or $b \in F$.

Lemma 3.8.23. In the situation of the definition above $\langle U \rangle$ is the smallest filter which contains U.

Proof. Trivial. □

Theorem 3.8.24. Let R be a commutative ring and $U \subseteq R$ be given, then

$$\bigcap \{P \subseteq R \mid U \subseteq P \text{ and } P \text{ is a prime filter of } R\} = \langle U \rangle.$$

Proof. We use Theorem 3.4.3 with $U \rhd x :\Leftrightarrow x \in \langle U \rangle$ and $\circ := +$. The proof of reflexivity, transitivity and encoding is similar to the proof of Lemma 3.8.28 by using Lemma 3.8.23. There is also a proof in [144, Section 4.1.2]. □

Notation 3.8.25. For a given ring R and $[a_1, \ldots, a_k] = \vec{a} \in R^*$ we define $\prod \vec{a} := \prod_{i=1}^{k} a_i$. For $n \in \mathbb{N}$, we define $n * \vec{a}$ recursively by

$$0 * \vec{a} := [], \qquad (n+1) * \vec{a} := \vec{a} :: (n * \vec{a}),$$

where :: is the concatenation of lists, i.e. $\prod(n * \vec{a}) = (\prod \vec{a})^n$.

Definition 3.8.26. Let a countable ring R be given. We define a covering $\rhd$ on R by $A \rhd x :\Leftrightarrow \exists_{c\in R}\exists_{\vec{a}\in R^*}(\vec{a} \in A^* \wedge xc = \prod \vec{a})$ and the operator $\circ$ is given by the addition $+$ on R. The witness type is given by $R \times R^*$ and $A \rhd_{(c,\vec{a})} x :\Leftrightarrow \vec{a} \in A^* \wedge xc = \prod \vec{a}$.

Proposition 3.8.27. In the situation of the definition above, the filters of R are exactly the $\rhd$-ideals, and the prime filters are exactly the prime $\rhd$-ideals.

Proof. Let F be a filter, and assume $A \rhd_{(c,\vec{a})} x$ for some $A \subseteq F$, then $\vec{a} \in F^*$ and $ac = \prod \vec{a} \in F$. By the second filter property it follows $a \in F$.

In the other direction, assume that F is a $\rhd$-ideal. $1 \in F$ follows from $\varnothing \rhd_{(1,[])} 1$. For the second property we first assume $a, b \in F$ then $ab \in F$ follows by $a, b \rhd_{(1,[a,b])} ab$, and if $ab \in F$, we have $a \in F$ and $b \in F$ by $ab \rhd_{(b,[ab])} a$ and $ab \rhd_{(a,[ab])} b$.

The second part of the lemma is now tautological because $\circ$ is exactly $+$. □

Lemma 3.8.28. In the situation of Definition 3.8.26 the covering $\rhd$ is a Σ^0_1-covering, and together with $\circ$ it has a computable cover structure (ι, τ, η).

Proof. Since R is countable, $R \times R^*$ can be considered as a base type, and therefore $\rhd$ is a Σ^0_1-covering. It remains to consider the three properties of a cover structure:

- Obviously $x1 = x$ and therefore $\iota(x) := (1, [x])$ does the trick.
- Let $A \rhd_{(c,\vec{a})} x$ and $B, x \rhd_{(d,\vec{b})} y$ be given. Then $\vec{a} \in A^*$, $xc = \prod \vec{a}$, $\vec{b} \in (B \cup \{x\})^*$ and $yd = \prod \vec{b}$. Let $\vec{b}'$ be the list which comes from $\vec{b}$ by erasing each component with entry x and k be the number of components with entry x in $\vec{b}$. In particular, $\vec{b}' \in B^*$ and $\prod \vec{b} = x^k \prod \vec{b}'$. It follows

$$ydc^k = x^k c^k \prod \vec{b}' = \left(\prod \vec{a}\right)^k \prod \vec{b}' = \prod \left((k * \vec{a}) :: \vec{b}'\right),$$

  and therefore

$$\tau(x, (c, \vec{a}), (d, \vec{b})) := \left(c^k d, (k * \vec{a}) :: \vec{b}'\right)$$

  with k and $\vec{b}'$ as defined above does the trick.
- Let $A, x \rhd_{(c,\vec{a})} z$ and $B, y \rhd_{(d,\vec{b})} z$ be given. So $\vec{a} \in (A \cup \{x\})^*$, $zc = \prod \vec{a}$, $\vec{b} \in (B \cup \{y\})^*$ and $zd = \prod \vec{b}$. We define $\vec{a}'$ as the list which comes from $\vec{a}$ by deleting each component with entry x and k be the number of components

with entry x in $\vec{a}$. Analogously, we define $\vec{b}'$ and l with $\vec{b}$ and y instead of $\vec{a}$ and x. Hence,

$$zc = x^k \prod \vec{a}', \qquad \vec{a}' \in A^*,$$
$$zd = y^l \prod \vec{b}', \qquad \vec{b}' \in B^*.$$

If $k = 0$ or $l = 0$, we are already done. Therefore, we assume $k, l \geq 1$ and we define

$$c' := \left(\prod \vec{a}'\right)^{k-1} \left(\prod \vec{b}'\right)^k \qquad \text{and} \qquad d' := \left(\prod \vec{a}'\right)^l \left(\prod \vec{b}'\right)^{l-1},$$

and this gives

$$zc' = \left(x \prod (\vec{a}' :: \vec{b}')\right)^k \qquad \text{and} \qquad zd' = \left(y \prod (\vec{a}' :: \vec{b}')\right)^l.$$

With the abbreviation $\Omega := \prod(\vec{a}' :: \vec{b}')$ we calculate

$$\begin{aligned}
\prod \Big((k+l) * (\vec{a}' :: \vec{b}' :: x + y)\Big) &= (x\Omega + y\Omega)^{k+l} \\
&= \sum_{i=0}^{k} \binom{k+l}{i} (x\Omega)^i (y\Omega)^{k+l-i} + \sum_{i=1}^{l} \binom{k+l}{i+k} (x\Omega)^{i+k} (y\Omega)^{l-i} \\
&= zd' \sum_{i=0}^{k} \binom{k+l}{i} (x\Omega)^i (y\Omega)^{k-i} + zc' \sum_{i=1}^{l} \binom{k+l}{i+k} (x\Omega)^i (y\Omega)^{l-i}.
\end{aligned}$$

Therefore, we define

$$\begin{aligned}
&\eta(x, y, (c, \vec{a}), (d, \vec{b})) \\
&\quad := \Bigg(d' \sum_{i=0}^{k} \binom{k+l}{i} (x\Omega)^i (y\Omega)^{k-i} + c' \sum_{i=1}^{l} \binom{k+l}{i+k} (x\Omega)^i (y\Omega)^{l-i}, \\
&\qquad\quad (k+l) * (\vec{a}' :: \vec{b}' :: x + y) \Bigg),
\end{aligned}$$

where $\vec{a}', \vec{b}', c', d'$ and Ω are defined as above. □

Corollary 3.8.29. Assume we are in the situation of Definition 3.8.26, and suppose that for $U \subseteq R$ and $r \in R$ a Krull functional ψ w.r.t. $\rhd$ and $\circ$ is given. Instantiating Algorithm 3.4.14 on the cover structure (ι, τ, η) of Lemma 3.8.28 and the Krull functional ψ, the algorithm terminates in some state π_k. Let $g_{\iota,\tau,\eta,\psi}(\pi_k) = (A, (c, \vec{a}))$, then $A \subseteq U$, $\vec{a} \in A^*$ and $rc = \prod \vec{a}$.

Proof. Follows directly from Theorem 3.4.15 and Lemma 3.8.28. □

Chapter 4

An algorithmic version of Zariski's lemma

This chapter is based on the paper [180].

Motivation 4.0.1. In this chapter we provide an elementary and constructive proof of Zariski's lemma. It claims the following:

> Let K be a field and R be an algebra over K which is a field. Suppose that $R = K[x_1, \ldots, x_n]$ for some $x_1, \ldots, x_n \in R$. Then $x_1, \ldots, x_n$ are algebraic over K.

Our proof will only use some basics of integral ring extensions and the consideration that R is discrete (in the sense of Definition 4.1.3).

After giving a constructive proof we take a look at the computational side: First, we formulate a computational interpretation of Zariski's lemma, we call it the *algorithmic version of Zariski's lemma*, and subsequently we use our constructive proof to develop an algorithm which shall realise the computational interpretation. At the end we prove that this algorithm indeed is a realiser of the computational interpretation.

This shows a typical approach in constructive mathematics. Analysing a theorem constructively often goes the following way:

- Formulate a *quite* constructive proof of the theorem.
- Formulate an algorithmic interpretation of the theorem.
- Inspired by the constructive proof formulate an algorithm which shall realise the algorithmic interpretation.

- Prove that the algorithm is indeed a realiser of the algorithmic interpretation.

This chapter serves as an example where all these steps are carried out manually on paper and where the formulation of the quite constructive proof is only necessary to get an inspiration for the other steps. If we were an omniscient magician, we could drop the first step. On the other hand, if we used computer support (as we do in Chapter 5), we would get the other three steps automatically. However, since neither of the two conditions is met, we do all fourth steps.

However, in Chapter 5 we see an example where only the quite constructive proof is formulated manually and the other steps are done by the computer. Note that we have written "quite constructive" because sometimes one can bypass a non-constructive moment or the non-constructive moment can be included as assumption in the algorithmic version. We also see an example of this here: Since our proof uses case distinction on $x = 0$ for every x in an algebraic structure, we assume that this structure is discrete.

At the end, we show an application of the algorithm and give an outlook how one can possibly use it together with the theory in Chapter 3 to get an algorithm for Hilbert's Nullstellensatz.

4.1 Background and basic definitions

Motivation 4.1.1. Presumably the first time Zariski's lemma appeared was in [186], where Oscar Zariskis used it to prove Hilbert's Nullstellensatz. In 1976, John McCabe gave an interesting but non-constructive proof [118], which relies on the existence of maximal ideals. In 2020, Daniel Wessel avoided this maximality argument by using Jacobson radicals [176]. However, the proof still contains a non-constructive moment. To wit, if R is an algebra over a field K and $S \subseteq R$ is a finite subset then there exists $S_0 \subseteq S$ maximal such that all elements in S_0 are algebraically independent over K. To avoid this, one could use the *Noether normalisation* theorem:

> Let K be a field and R be an algebra over K with $R = K[x_1, \ldots, x_n]$ for some $x_1, \ldots, x_n \in R$. Then there are $z_1, \ldots, z_n \in R$ and $m \leq n$ such that $K[z_1, \ldots, z_n] = R$ is integral over $K[z_1, \ldots, z_m]$ and $z_1, \ldots, z_m$ are algebraically independent over K.

A constructive proof of Noether normalisation is given in [83, Theorem 1.18], [121, Theorem 2.4] and [142, Section 3.13] and Zariski's lemma is a corollary of it, see for example [82, Theorem 1.15] and [142, Section 3.15].

Some proofs as in [12,13,162,186] are non-constructive but instead of a maximal algebraically independent subset they use induction on the number of generators of the algebra. This will also be part of our constructive proof. The proof in this chapter is a direct and constructive proof of Zariski's lemma. To get this proof, we have analysed the proofs in the sources above and have combined them together with some new ideas.

To formulate a constructive proof of Zariski's lemma, we assume that the definitions of rings, fields and algebras is already known. Before we formulate the algorithmic version, we give a concrete definition of these objects and even refine them. Therefore, if the reader is not familiar with these notions, they can look at Definition 4.3.2 before continuing to read. In our case all rings are commutative.

Notation 4.1.2. In the following we often say "$K[x_1,\ldots,x_n]$ is a K-algebra" and mean that we have a K-algebra R and $x_1,\ldots,x_n \in R$ with $R = K[x_1,\ldots,x_n]$. Note that we use small letters for elements in an algebra, whereas we use capital letters for the indeterminates of a polynomial ring.[1]

To shorten our formulas we will use the following syntactical abbreviations: $\vec{X} := X_1,\ldots,X_n$; $\vec{x} := x_1,\ldots,x_n$; $\vec{Y} := Y_1,\ldots,Y_m$ and $\vec{y} := y_1,\ldots,y_m$. Furthermore, for $n \in \mathbb{N}$ and any $I \in \mathbb{N}^n$ we define $\vec{x}^I := \prod_{i=1}^n x_i^{I_i}$ and $\vec{X}^I := \prod_{i=1}^n X_i^{I_i}$

Definition 4.1.3. We call a ring R *discrete* if $=$, $+$, $-$ and $\cdot$ are computable. Here $=$ is considered as a boolean valued function. A field K is called discrete if it is discrete as ring and a^{-1} is computable. In this chapter "computable" means that the equality is decidable and we can use the operators freely in our algorithms. In particular, we can distinguish $a = b$ and $a \neq b$ for all a, b in the given ring or field even if a and b are terms which contain $+, -, \cdot$ and $^{-1}$.

Remark 4.1.4. Note that in the textbooks like [115, 121, 184] a field or ring is discrete if the equality is decidable, i.e. $a = b \vee a \neq b$ for all a, b in the discrete structure.

In discrete fields, we have $x = 0 \vee \exists_y xy = 1$ for all x in the field.

Lemma 4.1.5. Let R be a discrete field, then all subrings of R and $R[X_1,\ldots,X_n]$ are discrete. For $f \in R[X_1,\ldots,X_n]$ we can decide if $f \in R$ or $f \notin R$, and if $f \in R[X]$, we can compute the degree of f by $\deg(f) := \max\{n \in \mathbb{N} \mid f_n \neq 0\}$, where we set $\max \varnothing := -1$.

Proof. Trivial. □

[1] At this point want to emphasize that statements like $R = K[x_1,\ldots,x_n]$ are constructively critical. Therefore, in the algorithmic part of this chapter we use the notion $K[x_1,\ldots,x_n]$ only as a way of speaking.

Definition 4.1.6. Let $R \subseteq S$ be a ring extension. We call an element $x \in S$ *integral* over R if there are $n \in \mathbb{N}$ and $a_{n-1}, \ldots, a_0 \in R$ with

$$x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = 0.$$

In other words, an element $x \in R$ is integral if there is a monic polynomial $P \in R[X]$ with $P(x) = 0$.

A ring extension $R \subseteq S$ is integral if each $x \in S$ is integral over R.

4.2 A constructive proof

Motivation 4.2.1. We now provide a constructive proof of Zariski's lemma. The proof does not use any non-constructive principles (except for assuming that the rings are discrete) as the law of excluded middle, Zorn's lemma or any other version of the axiom of choice. Since we do not formulate the algorithm in this section, we will not assume explicitly that each ring is discrete. As explained at the beginning of this chapter, the idea is not to formulate a totally constructive proof (whatever a "totally constructive proof" shall be), but to extract an algorithm out of a "sufficiently" constructive proof.

We will first need two lemmas from the theory of integral ring extensions. Despite the fact that the proofs are given in many textbooks of algebra such as [12, Chapter 5], we also carry out the proofs here since in the next section we use them later to establish an algorithm for the computational interpretation.

Lemma 4.2.2. Let $R \subseteq S$ be a ring extension and $\vec{x} \in S$ be integral over R, then $R \subseteq R[\vec{x}]$ is an integral ring extension.

Proof. Since $x_1, \ldots, x_n$ are integral over R, for each $i \in \{1, \ldots, n\}$ there are $k_i \in \mathbb{N}$ and $a^{(i)}_{k_n-1}, \ldots, a^{(i)}_0 \in R$ with

$$x_i^{k_i} + a^{(i)}_{k_n-1}x_i^{k_n-1} + \cdots + a^{(i)}_1 x_i + a^{(i)}_0 = 0 \tag{4.1}$$

We define $\mathcal{I} := \{I \in \mathbb{N}^n \mid I_1 < k_1, \ldots, I_n < k_n\}$, then $(\vec{x}^I)_{I\in\mathcal{I}}$ is a generator of $R[x_1, \ldots, x_n]$ as R-module because of the equations from (4.1).

Let $y \in R[\vec{x}]$ be given. Then for all $I \in \mathcal{I}$ there is $(b_{IJ})_{J\in\mathcal{I}} \in R^{\mathcal{I}}$ such that

$$y\vec{x}^I = \sum_{J\in\mathcal{I}} b_{IJ}\vec{x}^J.$$

Therefore, the multiplication by y has the transformation matrix $(b_{IJ})_{I,J\in\mathcal{I}}$ w.r.t. the generator $(\vec{x}^I)_{I\in\mathcal{I}}$ of the R-module $R[x_1, \ldots, x_n]$. Let $P \in R[X]$ be the characteristic

polynomial of this matrix. By the theorem of Cayley-Hamilton [69, Theorem 4.3] we have $P(y) = 0$ and P is monic because it is a characteristic polynomial. Hence, y is integral over R. □

Lemma 4.2.3. Let $R \subseteq S$ be an integral ring extension. If S is a field then so is R.

Proof. Let $x \in S$ with $x \neq 0$ be given.[2] Since S is a field there exists $x^{-1} \in S$, and since $R \subseteq S$ is integral, there are $a_{n-1}, \ldots, a_0 \in R$ such that

$$x^{-n} + a_{n-1}x^{-n+1} + \cdots + a_1x^{-1} + a_0 = 0.$$

By multiplying this equation by x^{n-1} and isolating x^{-1} we get

$$x^{-1} = -a_{n-1} - \cdots - a_1x^{n-2} - a_0x^{n-1} \in R.$$

Therefore, R is a field. □

Motivation 4.2.4. Our proof of Zariski's lemma uses induction on the number n of generators and reduction to the case $n = 2$. Therefore, we first consider the cases $n = 1$ and $n = 2$ in the next two lemmas.

Lemma 4.2.5. Let K be a field and R be an algebra over K, which is a field. Suppose that $R = K[x]$ for some $x \in R$. Then x is algebraic over K.

Proof. If $x = 0$, it is obvious that x is algebraic over K. If $x \neq 0$, there is $p \in K[X]$ with $xp(x) = 1$ because $R = K[x]$ and R is a field. We set $q := Xp - 1 \in K[X]$. Then $q \neq 0$ because $p \neq 0$ and $\deg(Xp - 1) > 0$, and so $q(x) = 0$ is an algebraic equation for x. □

Lemma 4.2.6. Let K be a field and R be an algebra over K, which is a field. Suppose that $R = K[x_1, x_2]$ for some $x_1, x_2 \in R$. Then x_1, x_2 are algebraic over K.

Proof. We show that x_1 is algebraic. The argument for x_2 is analogous. If $x_2 = 0$ we are done by Lemma 4.2.5. Otherwise, we have $p \in K[X_1, X_2]$ with $p(x_1, x_2)x_2 = 1$. Therefore, $q := Xp(x_1, X) - 1$ is a polynomial in $K[x_1][X]$ with $q(x_2) = 0$ and $q \neq 0$ as its constant coefficient is -1. Let $y \in K[x_1]$ be the leading coefficient of q, which is non-zero by definition. Then $K[x_1, y^{-1}] \subseteq K[x_1, x_2]$ is an integral ring extension by Lemma 4.2.2 because x_2 is integral over $K[x_1, y^{-1}]$ witnessed by $y^{-1}q \in K[x_1, y^{-1}][X]$. Thus, $K[x_1, y^{-1}]$ is a field by Lemma 4.2.3.

[2]Note that $x \neq 0$ is an assumption. Therefore, we do not need that R is discrete in the computational interpretation of this lemma, i.e. in Lemma 4.3.12.

With this preparation we are now able to construct a non-zero polynomial with root x_1. By $y \in K[x_1]$, there is $f \in K[X]$ such that $f(x_1) = y$. If $f \in K$ then $K[x_1, y^{-1}] = K[x_1]$ and we are done by Lemma 4.2.5. We assume $f \in K[X] \setminus K$. If $1 - f(x_1) = 0$ then x_1 is algebraic over K. Otherwise, $1 - f(x_1)$ is invertible[3] in $K[x_1, y^{-1}]$, and therefore there is $h \in K[X]$ and $N \in \mathbb{N}$ with $(1 - f(x_1))^{-1} = h(x_1)y^{-N} = h(x_1)f(x_1)^{-N}$. Consequently, we have

$$f(x_1)^N - h(x_1)(1 - f(x_1)) = 0.$$

It remains to show that $f^N - h(1 - f) \neq 0$ in $K[X]$. By the binomial theorem there is a $g \in K[X]$ with $f^N = 1 + (1 - f)g$, hence

$$f^N - h(1 - f) = 1 + (1 - f)(g - h).$$

Since f in non-constant, also $1-f$ is non-constant. Now assume that $f^N - h(1-f) = 0$ then $g - h = 0$ as otherwise $\deg((1 - f)(g - h)) > 0$ and $1 + (1 - f)(g - h) \neq 0$. But then $0 = 1$, a contradiction. □

Theorem 4.2.7 (Zariski's lemma). Let K be a field and R be an algebra over K, which is a field. Suppose that $R = K[\vec{x}]$ for some $\vec{x} \in R$. Then $\vec{x}$ are algebraic over K.

Proof. If $n = 0$, there is nothing to show, and if $n = 1$, the statement follows by Lemma 4.2.5.

For $n \geq 2$, we use induction over n. The base case $n = 2$ is done in Lemma 4.2.6. For the induction step, we assume $n \geq 3$. Let $L := K(x_1) := \mathrm{Quot}(K[x_1])$. Since R is a field, we have $L \subseteq R$ and thus $L[x_2, \ldots, x_n] = R$. By induction each x_i for $i \in \{2, \ldots, n\}$ is algebraic over L. Therefore, there are a monic polynomials $f_i \in L[X]$ with $f_i(x_i) = 0$ for all $i \in \{2, \ldots, n\}$. Let v_i be the product of the denominator of all coefficients in f_i and $v := \prod_{i=2}^{n} v_i$. Then, v is invertible as product of non-zero elements and all x_i are integral over $K[x_1, v^{-1}]$. In particular, $K[x_1, v^{-1}] \subseteq K[\vec{x}]$ is an integral ring extension by Lemma 4.2.2, and $K[x_1, v^{-1}]$ is a field by Lemma 4.2.3. Hence, x_1 is algebraic over K by Lemma 4.2.6. □

[3]The idea to take $1-f(x_1)$ is based on a note by Daniel Wessel [176] and a personal conversation with Henri Lombardi. Inspired by [162], the author's first approach was to take $g(x_1)$ for some irreducible $g \in K[X]$ with $g \nmid f$. This is also possible if one can give a construction for such a g.

4.3 Computational interpretation

4.3.1 Preliminary

Motivation 4.3.1. In this section, we develop the algorithmic version of Zariski's lemma. For each lemma in the last section, we state a "sub"-algorithm and after this algorithm we give a lemma or theorem which describes the properties of this algorithm. This can be seen as the computational interpretation of the original statement.

Before formulating the entire algorithm and the computational interpretation, we first have to make clear on which objects this algorithm operates. More specifically: if we state an algorithm about a field, we do not use the field axioms in the algorithm but we will use the field structure like $+$, $\cdot$, 0, 1 and so on. Therefore, we will first define the underlying structures precisely.

Definition 4.3.2. In the setting of this chapter a *ring structure* $(R, +, \cdot, 0, 1, -, =)$ is a set R equipped with an addition operator $+ : R \times R \to R$, a multiplication operator $\cdot : R \times R \to R$, a zero element $0 \in R$, an unit element $1 \in R$, an additive inverse function $- : R \to R$ and an equality $= \subseteq R \times R$. If furthermore $=$ is an equivalence relation and compatible with $+, \cdot, -$ (i.e. $=$ is a congruence relation on $(R, +, \cdot, 0, 1, -)$) and the other ring axioms are fulfilled (w.r.t. the equality $=$), R is a *ring*. In our case a ring is always commutative. We call $(K, +, \cdot, 0, 1, -, ^{-1}, =)$ a *field structure* if $(K, +, \cdot, 0, 1, -, =)$ is a ring structure and $^{-1} : K \to K$ is a map. If K is a ring such that $xx^{-1} = 1 \vee x = 0$ for all $x \in K$ and $0 \neq 1$, we call K a field.

The notions *discrete ring structure* and *discrete field structure* are analogously defined as discrete rings and discrete fields, respectively. In particular, if a structure is discrete, we can freely use their operators in the algorithms.

Since the notation of $+, \cdot, 0, 1, -, ^{-1}$ and $=$ will not change, we do not mention it and just say that R is a ring (structure) or K is a field (structure). A *homomorphism* $\phi : R \to S$ between two ring structures R and S is a map on the underlying sets which preserves the structure in the canonical way.

For a ring structure R we define the *ring structure of polynomials* $R[X]$ with coefficients in R by the well-known construction. Formally, the underlying set of $R[X]$ is the set R^* of all finite sequences in R. For $n \in \mathbb{N}$ we have also the polynomial ring structure in n variables denoted by $R[X_1, \ldots, X_n]$. It consists of all sums $\sum_{I \in \mathcal{I}} a_I \prod \vec{X}^I$ where $\mathcal{I} \in \mathcal{P}_{fin}(\mathbb{N}^n)$ and $a_I \in R$.

An *algebra structure* R over a field structure K, or short *K-algebra structure*, is a ring structure together with a map $K \to R$. If R is a ring, K is a field and the map $K \to R$ is a homomorphism, we call it *K-algebra*. For a K-algebra R and

$x_1, \ldots, x_n \in R$ we get an extension $K[X_1, \ldots, X_n] \to R$ of the homomorphism by $X_i \mapsto x_i$. We denote the image by $K[x_1, \ldots, x_n]$, where an element is in the image of a homomorphism if it is equal (w.r.t. $=$) to a value of the homomorphism.

Remark 4.3.3. It can be easily observed that Lemma 4.1.5 is also valid if R is just a discrete ring structure instead of a discrete ring and if R is discrete, so is $R[\vec{X}]$.

In terms of model theory, a ring structure is a structure in the language of rings, and a ring is a model of the theory of (commutative) rings, and analogously for fields and algebras.

Motivation 4.3.4. In Zariski's lemma a K-algebra $K[\vec{x}]$ is given. In particular, there is a surjective homomorphism from $K[\vec{X}]$ to $K[\vec{x}]$. It is well-known that the existence of a right-inverse of a surjection is only implied by the axiom of choice. Hence, instead of using the right inverse of this homomorphism computationally, we are working on the level of the polynomial rings. In particular, we need a computational interpretation of $K[\vec{y}] \subseteq K[\vec{x}]$ being a ring extension and of $K[\vec{x}]$ being a field on the level of polynomials.

Definition 4.3.5. Let K be a field and R be a K-algebras and $\vec{x}, \vec{y} \in R$. We say that $K[\vec{y}] \subseteq K[\vec{x}]$ is a *ring extension of K-algebras* witnessed by $\vec{h} := h_1, \ldots, h_m \in K[\vec{X}]$ if $h_i(\vec{x}) = y_i$ for all $i \in \{1, \ldots, m\}$. In short notation we write $\vec{h}(\vec{x}) = \vec{y}$.

Definition 4.3.6. Let a field K, a K-algebra R and elements $\vec{x} \in R$ be given. A function $\iota : K[\vec{X}] \to K[\vec{X}]$ is called *algebraic inverse function* on $K[\vec{x}]$ if

$$(\iota(f))(\vec{x})f(\vec{x}) = 1$$

for all $f \in K[\vec{X}]$ with $f(\vec{x}) \neq 0$.

Remark 4.3.7. Note that an algebraic inverse function on $K[\vec{x}]$ does not have to be compatible with the equality relation on the ring structure $K[\vec{X}]$.

From an algebraic inverse function on $K[\vec{x}]$ and a right inverse of the surjection $K[\vec{X}] \to K[\vec{x}]$ we get that $K[\vec{x}]$ is a field. However, in our constructive considerations we use terms like $K[\vec{x}]$ only as mode of speaking like in the two definition above. In particular, we avoid terms like $R = K[\vec{x}]$ or $a \in K[\vec{x}]$ since constructively this in general not decidable.

In the light of this definition: an algorithm which realises Zariski's lemma, takes an algebraic inverse function on $K[\vec{x}]$ as input and returns non-constant polynomials $f_1, \ldots, f_n \in K[X]$ with $f_i(x_i) = 0$ for all $i \in \{1, \ldots, n\}$.

4.3.2 Some algorithms for integral extensions of algebras

Motivation 4.3.8. In this and the next section we transform the statements of Section 4.2 into algorithms and prove their correctness. Here we use the notions of Definition 4.3.2, 4.3.5 and 4.3.6 and the proof of the respective statement. In this section we start with the statements about integral extension, i.e. Lemma 4.2.2 and Lemma 4.2.3.

Algorithm 4.3.9. Given a field structure K, $f \in K[\vec{X}]$ and $k_i \in \mathbb{N}$; $g_{k_i-1}^{(i)}, \ldots, g_0^{(i)} \in K[\vec{Y}]$ for each $i \in \{1, \ldots, n\}$. We compute $k \in \mathbb{N}$ and $g_{k-1}, \ldots, g_0 \in K[\vec{Y}]$ as follows:

1. Define $\mathcal{I} := \{I \in \mathbb{N}^n | I_1 < k_i, \ldots, I_n < k_n\}$ and for each $I \in \mathcal{I}$ compute the finite sum $f\vec{X}^I = \sum_{J \in \mathbb{N}^n} f_{IJ}\vec{X}^J$ with $f_{IJ} \in K$.

2. For each $I \in \mathcal{I}$ and $i \in \{1, \ldots, n\}$ replace each $X_i^{k_i}$ by $-g_{k_i-1}^{(i)}X^{k_i-1} - \cdots - g_0^{(i)}$ in $\sum_{J \in \mathbb{N}^n} f_{IJ}\vec{X}^J$ one by one until we obtain a polynomial of the form $\sum_{J \in \mathcal{I}} g_{IJ}\vec{X}^J$ with $g_{IJ} \in K[\vec{Y}]$.[4]

3. Compute the characteristic polynomial $P \in K[\vec{Y}][X]$ of the matrix $(g_{IJ})_{I,J \in \mathcal{I}}$ as the determinant of the matrix $(\delta_{IJ}X - g_{IJ})_{I,J \in \mathcal{I}}$, where $\delta_{IJ}X := X$ if $I = J$, and $\delta_{IJ}X := 0$ if $I \neq J$.

4. Let $P = \sum_{i=0}^{l} g_i X^i$ for some $l \in \mathbb{N}$ and $g_i \in K[\vec{Y}]$. Return $k := \prod_{i=1}^{n} k_i$ and the first k coefficients $g_{k-1}, \ldots, g_0$ of P, where $g_i := 0$ if $i > l$.

Lemma 4.3.10. In the situation of Algorithm 4.3.9 we assume that K is a field, R is a K-algebra and $\vec{x}, \vec{y} \in R$ with

$$x_i^{k_i} + g_{k_i-1}^{(i)}(\vec{y})x_i^{k_i-1} + \cdots + g_0^{(i)}(\vec{y}) = 0 \tag{4.2}$$

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

$$(f(\vec{x}))^k + g_{k-1}(\vec{y})(f(\vec{x}))^{k-1} + \cdots + g_0(\vec{y}) = 0.$$

Proof. We define the $K[\vec{Y}]$-module $M := K[\vec{Y}][\vec{X}]/\langle G_1, \ldots, G_n\rangle$ where $G_i := X^{k_i} + g_{k_i-1}^{(i)}X^{k_i-1} + \cdots + g_0^{(i)}$ for all i, and go through the steps of Algorithm 4.3.9. Because of the notation in Step 1 and the process in Step 2, we have

$$\sum_{J \in \mathbb{N}^n} f_{IJ}\vec{X}^J = \sum_{J \in \mathcal{I}} g_{IJ}(\vec{Y})\vec{X}^J$$

[4]Note that we do not determine an order in which these replacement has to take place. Therefore, the result is not unique without any further assumptions. However, this is not necessary and Lemma 4.3.10 holds independently of the chosen order.

in M or in other words

$$\sum_{J\in\mathbb{N}^n} f_{IJ}\vec{X}^J - \sum_{J\in\mathcal{I}} g_{IJ}(\vec{Y})\vec{X}^J \in \langle G_1,\dots,G_n\rangle$$

considered in $K[\vec{Y}][\vec{X}]$. Note that $(\vec{X}^I)_{I\in\mathcal{I}}$ is a set of generators of M and multiplication with f corresponds to the matrix $(g_{IJ})_{I,J\in\mathcal{I}}$. Let P be the characteristic polynomial as in Step 3. By the theorem of Cayley-Hamilton, $P(f) = 0$ in M, hence $P(f) \in \langle G_1,\dots,G_n\rangle$ in $K[\vec{Y}][\vec{X}]$. By Formula 4.3.10, we have $G_i(\vec{y},\vec{x}) = 0$ for all i, and hence $0 = P(f)(\vec{y},\vec{x}) = (f(\vec{x}))^k + g_{k-1}(\vec{y})(f(\vec{x}))^{k-1} + \cdots + g_0(\vec{y})$. Here, we have used the definition of the g_i in the last step, and $\deg(P) = k$ because k is the number of elements in $\mathcal{I}$, which is also the cardinality of the generator $(x^I)_{I\in\mathcal{I}}$. □

Algorithm 4.3.11. Let a field structure K, $\vec{h} := h_1,\dots,h_m \in K[\vec{X}]$, $\iota : K[\vec{X}] \to K[\vec{X}]$ and $k_i \in \mathbb{N}$, $g^{(i)}_{k_i-1},\dots,g^{(i)}_0 \in K[\vec{Y}]$ for each $i \in \{1,\dots,n\}$ be given. We define a map $\tilde{\iota} : K[\vec{Y}] \to K[\vec{Y}]$ as follows:

1. Given an input $f \in K[\vec{Y}]$, compute $p := \iota(f(\vec{h})) \in K[\vec{X}]$.
2. Apply Algorithm 4.3.9 to K, p and k_i, $g^{(i)}_{k_i-1},\dots,g^{(i)}_0$ for each $i \in \{1,\dots,n\}$ to get $k \in \mathbb{N}$ and $g_{k-1},\dots,g_0 \in K[\vec{Y}]$.
3. Return $-g_{k-1} - g_{k-2}f - \cdots - g_0 f^{k-1}$.

Lemma 4.3.12. In the situation of Algorithm 4.3.11 we assume that K is a field, R is a K-algebra and $\vec{x},\vec{y} \in R$ such that $K[\vec{y}] \subseteq K[\vec{x}]$ is an extension of K-algebras witnessed by $\vec{h}$, ι is an algebraic inverse function and

$$x_i^{k_i} + g^{(i)}_{k_i-1}(\vec{y})x_i^{k_i-1} + \cdots + g^{(i)}_0(\vec{y}) = 0$$

for all $i \in \{1,\dots,n\}$. Then $\tilde{\iota}$ is an algebraic inverse function on $K[\vec{y}]$.

Proof. Let $f \in K[\vec{Y}]$ with $f(\vec{y}) \neq 0$ be given. Since $\vec{h}$ is a witness that $K[\vec{y}] \subseteq K[\vec{x}]$ is an extension of K-algebras, we have $f(\vec{h}(\vec{x})) = f(\vec{y}) \neq 0$. Let p be given as in Step 1. Then $p(\vec{x})f(\vec{y}) = 1$ since ι is an algebraic inverse function. By Lemma 4.3.10 we have

$$(p(\vec{x}))^k + g_{k-1}(\vec{y})(p(\vec{x}))^{k-1} + \cdots + g_0(\vec{y}) = 0.$$

Multiplying this with $(f(\vec{y}))^{k-1}$ and isolating $p(\vec{x})$ leads to

$$p(\vec{x}) = (-g_{k-1} - g_{k-2}f - \cdots - g_0 f^{k-1})(\vec{y}) = \tilde{\iota}(f)(\vec{y}).$$

□

Remark 4.3.13. We have not defined a computational interpretation of $K[\vec{y}] \subseteq K[\vec{x}]$ being an integral extension. The reason is that we only need an integral equation on the level of polynomials for $\vec{x}$ as in Algorithm 4.3.9 and Lemma 4.3.10. Using this algorithm and the lemma leads to an integral equation of any $f(\vec{x})$ on the level of polynomials, hence we do not need to define an integral extension of K-algebras explicitly.

4.3.3 An algorithm for Zariski's lemma

Algorithm 4.3.14. Given a discrete field structure K, a discrete K-algebra structure R, $x \in R$ and $\iota : K[X] \to K[X]$, we compute an element $f \in K[X]$ as follows:

1. If $x = 0$, return X.
2. If $x \neq 0$, return $X\iota(X) - 1$.

Lemma 4.3.15. In the situation of Algorithm 4.3.14 we assume that K is a field, R is a K-algebra, $x \in R$ and ι is an algebraic inverse function on $K[x]$. Then f is non-constant and $f(x) = 0$, i.e. x is algebraic over K.

Proof. The case distinction is allowed since R is discrete. If $x = 0$, it is trivial. If $x \neq 0$ then $f = X\iota(X) - 1$ is obviously not constant zero and $f(x) = 0$ by the definition of an algebraic inverse function. □

Algorithm 4.3.16. Let a discrete field structure K, a discrete K-algebra structure R, two elements $x_1, x_2 \in R$ and $\iota : K[X_1, X_2] \to K[X_1, X_2]$ be given. We compute $f_1, f_2 \in K[X]$ as follows starting with f_1:

1. If $x_2 = 0$, we use Algorithm 4.3.14 with input K, R, $x_1 \in R$ and $\iota' : K[X] \to K[X]$ defined by $\iota'(p) := \iota(p(X_1))(X, 0)$ and return the output as f_1.
2. Otherwise, compute $\iota(X_2)$ and define g as the polynomial which comes from $X_2\iota(X_2)-1 \in K[X_1, X_2]$ by dropping each coefficient $p \in K[X_1]$ with $p(x_1) = 0$ and let $h \in K[X_1]$ be the leading coefficient of g (and 1 if $g = 0$).
3. Apply Algorithm 4.3.11 to the input K, $\vec{h} := (X_1, \iota(h))$, ι, $g_0^{(1)} = Y_1$ and $g_{k_2-1}^{(2)}, \ldots, g_0^{(2)} \in K[Y_1, Y_2]$, where $g_{k_2-1}^{(2)}, \ldots, g_0^{(2)}$ are the coefficients of $Y_2 g(Y_1, X)$ except the leading coefficient. Let $\tilde{\iota} : K[Y_1, Y_2] \to K[Y_1, Y_2]$ be the output of this algorithm.
4. If $\deg(h) = 0$, i.e. $h = h_0$ for some $h_0 \in K$, apply Algorithm 4.3.14 to K, R, $x_1 \in R$ and $\iota' : K[X] \to K[X]$ given by $\iota'(p) := \tilde{\iota}(p(Y_1))(X, h_0^{-1})$ and return the output of this algorithm as f_1.

5. Otherwise, check if $1 - h(x_1) = 0$. If yes, return $f_1 := 1 - h(X)$.

6. If no, compute $\tilde{\iota}(1 - h(Y_1)) = \sum_{i=0}^{N} a_i Y_2^i$ with $a_i \in K[Y_1]$ and $a_N \neq 0$; define

$$q := \sum_{i=0}^{N} a_i (h(Y_1))^{N-i} \in K[Y_1]$$

and return $f_1 := h(X)^N - (1 - h(X))q(X)$.

Exchange x_1 and x_2 and repeat the steps above to compute $f_2 \in K[X]$.

Lemma 4.3.17. In the situation of Algorithm 4.3.16 we assume that K is a field, R is a K-algebra, $x_1, x_2 \in R$ and ι is an algebraic inverse function on $K[x_1, x_2]$. Then $f_1(x_1) = f_2(x_2) = 0$ and f_1, f_2 are non-constant.

Proof. It suffices to consider f_1 since the statement with f_2 is proven analogously. We follow the algorithm step by step. If $x_2 = 0$ in Step 1, we use Lemma 4.3.14. That ι' is an algebraic inverse function on $K[x_1]$ follows from

$$(\iota(p(X_1)))(x_1, 0)p(x_1) = (\iota(p(X_1))p(X_1))(x_1, x_2) = 1$$

for all $p \in K[X]$ with $p(x_1) \neq 0$.

In Step 2 note that $g(x_1, x_2) = x_2\iota(X_2)(x_1, x_2) - 1 = 0$ and the constant coefficient of g (as polynomial in X_2) is equal to -1.

In Step 3 it is obvious that $X_1, \iota(h)$ is a witness of $K[x_1, \iota(h)(x_1, x_2)] \subseteq K[x_1, x_2]$ being an extension of K-algebras and $x_1 - g_0^{(1)}(x_1, \iota(h)(x_1, x_2)) = x_1 - x_1 = 0$. Furthermore, let $g = \sum_{i=0}^{k_2} g_i X_2^i$ for some $g_i \in K[X_1]$ with $g_{k_2} \neq 0$. Then $h = g_{k_2}$, $g_i^{(2)}(Y_1, Y_2) = g_i(Y_1)Y_2$ for all $i < k_2$, and therefore

$$\begin{aligned} 0 = \iota(h)(x_1, x_2)g(x_1, x_2) &= x_2^{k_2} + \sum_{i=0}^{k_2-1} g_i(x_1)\iota(h)(x_1, x_2)x_2^i \\ &= x_2^{k_2} + \sum_{i=0}^{k_2-1} g_i^{(2)}(x_1, \iota(h)(x_1, x_2))x_2^i. \end{aligned}$$

Hence, $\tilde{\iota}$ is an algebraic inverse function on $K[x_1, \iota(h)(x_1, x_2)]$ by Lemma 4.3.12.

If $\deg(h) = 0$ in Step 4, we have $h = h_0$ for some $h_0 \in K$ and $h_0 \neq 0$ because h is a leading coefficient. Therefore, it follows $\iota(h)(x_1, x_2) = h_0^{-1}$ and we apply Lemma 4.3.15 to $K[x_1] = K[x_1, h_0^{-1}]$. In order to apply this lemma it remains to show that ι' is an algebraic inverse function. If $p \in K[X]$ with $p(x_1) \neq 0$ then

$$(\iota'(p))(x_1)p(x_1) = (\tilde{\iota}(p(Y_1)))(x_1, h_0^{-1})p(x_1) = (\tilde{\iota}(p(Y_1))p(Y_1))(x_1, h_0^{-1}) = 1.$$

Hence, in the following we may assume $\deg(h) \neq 0$.

In the case of Step 5 f_1 is non-constant since $\deg(h) \neq 0$, and $f(x_1) = 0$ by the case assumption.

In Step 6 we have $1 - h(x_1) \neq 0$ and

$$q(x_1)(\iota(h)(x_1, x_2))^N = \tilde{\iota}(1 - h(Y_1))(x_1, \iota(h)(x_1, x_2)).$$

Since ι and $\tilde{\iota}$ are algebraic inverse functions and $h \neq 0$ and $1 - h \neq 0$, it follows

$$q(x_1)(1 - h(x_1)) = (h(x_1))^N.$$

Hence, for $f_1 := h(X)^N - (1 - h(X))q(X)$ we have $f_1(x_1) = 0$ and $f_1 \neq 0$ similar to the end of the proof of Lemma 4.2.6. □

Algorithm 4.3.18. Let a discrete field structure K, a discrete K-algebra structure R, $n > 0$ and $\vec{x} \in R$ be given. We define a field structure as follows:

$$L := \left\{ \frac{f}{g} \;\middle|\; f, g \in K[X], g(x_1) \neq 0 \vee 0 = 1 \right\},$$

$$\frac{f_1}{g_1} = \frac{f_2}{g_2} :\Leftrightarrow f_1(x_1)g_2(x_1) = f_2(x_1)g_1(x_1),$$

$$\frac{f_1}{g_1} + \frac{f_2}{g_2} := \begin{cases} \frac{f_1 g_2 + f_2 g_1}{g_1 g_2} & \text{if } (g_1 g_2)(x_1) \neq 0 \\ \frac{0}{1} & \text{else} \end{cases},$$

$$\frac{f_1}{g_1}\frac{f_2}{g_2} := \begin{cases} \frac{f_1 f_2}{g_1 g_2} & \text{if } (g_1 g_2)(x_1) \neq 0 \\ \frac{0}{1} & \text{else} \end{cases},$$

$$0 := \frac{0}{1}, \qquad 1 := \frac{1}{1},$$

$$-\frac{f}{g} := \frac{-f}{g}, \qquad \left(\frac{f}{g}\right)^{-1} := \begin{cases} \frac{g}{f} & \text{if } f(x_1) \neq 0 \\ \frac{0}{1} & \text{else} \end{cases}$$

For a given map $\iota : K[\vec{X}] \to K[\vec{X}]$ we define a map $\phi : L \to R, \frac{f}{g} \mapsto f(x_1)(\iota(g))(x_1)$, which turns R into an L-algebra structure. Furthermore, we define $\tilde{\iota} : L[X_2, \ldots, X_n] \to L[X_2, \ldots, X_n]$ as follows:

1. Given an input $p \in L[X_2, \ldots, X_n]$. It has the presentation

$$p = \sum_{i_2, \ldots, i_n} \frac{f_{i_2 \ldots i_n}}{g_{i_2 \ldots i_n}} X_2^{i_2} \cdots X_n^{i_n},$$

for finitely many $f_{i_2 \ldots i_n}, g_{i_2 \ldots i_n} \in K[X]$.

2. Let $a \in K[X]$ be the product of all these $g_{i_2\dots i_n}$, and for $j_2, \dots, j_n$ let $h_{j_2\dots j_n}$ be the product of all these $g_{i_2\dots i_n}$ except $g_{j_2\dots j_n}$.

3. Define $\tilde{f}_{i_2\dots i_n} := f_{i_2\dots i_n} h_{i_2\dots i_n}$ and $\tilde{p} := \sum_{i_2,\dots,i_n} \tilde{f}_{i_2\dots i_n}(X_1) X_2^{i_2} \cdots X_n^{i_n}$ and set

$$\tilde{\iota}(p) := (a(X_1)\iota(\tilde{p}))\left(\frac{X}{1}, X_2, \dots, X_n\right),$$

where we consider $b \in K$ also as the element $\frac{b}{1} \in L$.

Remark 4.3.19. In the algorithm above the definition of L and the operators are more complicated than one might expect. This is the case because we do not have any axiom available but we have to define the algorithm in each case.

As already mentioned, we can not define L as $\left\{\frac{a}{b} \mid a, b \in K[x_1], b \neq 0 \vee 0 = 1\right\}$, which is the field of fractions of $K[x_1]$ if it is an integral domain. That is the case as we want to avoid terms like $a \in K[x_1]$, which are constructively delicate. In particular, there is in general no map which takes $a \in K[x_1]$ and returns $f \in K[X]$ with $f(x_1) = a$ without using the axiom of choice. Therefore, we operate on the level of polynomials.

Lemma 4.3.20. In the situation of Algorithm 4.3.18 we assume that K is a field, R is a ring, $\vec{x} \in R$ and ι is an algebraic inverse function on $K[\vec{x}]$. Then L is indeed a discrete field. Given ι, the map $L \to K[\vec{x}], \frac{f}{g} \mapsto f(x_1)(\iota(g))(x_1)$ turns R into an L-algebra and $\tilde{\iota}$ is an algebraic inverse function on $L[x_2, \dots, x_n]$.

Proof. L is a discrete field because in the definition of L and its operators we only use the operators of K and $K[\vec{X}]$.

By using the property of an algebraic inverse function, it is also straightforward to check that the map ϕ is a homomorphism.

It remains to prove that $\tilde{\iota}$ is an algebraic inverse function on $L[x_2, \dots, x_n]$. For this let $p \in L[X_2, \dots, X_n]$ with $p(x_2, \dots, x_n) \neq 0$ be given. We take the representation of p as in Step 1 of the algorithm, a be defined as in Step 2 and $\tilde{f}_{i_2\dots i_n}$ and $\tilde{p}$ as in Step 3. We calculate

$$\begin{aligned} p(x_2 \dots, x_n)a(x_1) &= \sum_{i_2,\dots,i_n} \phi\left(\frac{\tilde{f}_{i_2\dots i_n}}{1}\right) x_2^{i_2} \cdots x_n^{i_n} \\ &= \sum_{i_2,\dots,i_n} \tilde{f}_{i_2\dots i_n}(x_1) x_2^{i_2} \cdots x_n^{i_n} = \tilde{p}(x_1, \dots, x_n). \end{aligned}$$

We have $a(x_1) \neq 0$ as it is a product of non-zero factors. Hence, if $p(x_2, \ldots, x_n) \neq 0$, also $\tilde{p}(x_1, \ldots x_n) \neq 0$. Since additionally ι is an algebraic inverse function,

$$\iota(\tilde{p})(x_1, \ldots, x_n) = (\tilde{p}(x_1, \ldots, x_n))^{-1},$$

and therefore

$$\begin{aligned}(p(x_2, \ldots, x_n))^{-1} &= a(x_1)(\tilde{p}(x_1, \ldots, x_n))^{-1} = a(x_1)(\iota(\tilde{p}))(x_1, \ldots, x_n) \\ &= (a(X_1)\iota(\tilde{p}))\left(\frac{X}{1}, x_2, \ldots, x_n\right).\end{aligned}$$

□

Algorithm 4.3.21. Let K be a discrete field structure, R be a discrete K-algebra structure, $\iota : K[\vec{X}] \to K[\vec{X}]$ be a map and $x_1, \ldots, x_n \in R$. We compute $f_1, \ldots, f_n \in K[X]$ by recursion over n as follows:

1. If $n = 0$, return the empty list. If $n = 1$, use Algorithm 4.3.14 with input K, R, x_1 and ι, and return the output f_1. If $n = 2$, use Algorithm 4.3.16 with input K; R; $x_1, x_2 \in R$ and ι, and return the output f_1, f_2.

2. Apply Algorithm 4.3.18 to K, R, n, $\vec{x}$ and ι, and let the field structure L and the map $\iota' : L[X_2, \ldots, X_n] \to L[X_2, \ldots, X_n]$ be the output.

3. By recursion apply the algorithm to L, the L-algebra structure R, ι' and $x_2, \ldots, x_n \in R$ to get $\tilde{F}_2, \ldots, \tilde{F}_n \in L[X]$.

4. For each i we define F_i as $\tilde{F}_i$ divided by its leading coefficient and replacing the leading coefficient by 1 (or $F_i := 1$ if $\tilde{F}_i = 0$).[5] In particular,

$$F_i = X^{n_i} + \sum_{j=0}^{n_i - 1} \frac{a_{ij}}{b_{ij}} X^j$$

   for some $a_{ij}, b_{ij} \in K[X]$.

5. Let $v := \prod_{(k,l)} b_{kl} \in K[X]$, $\tilde{b}_{ij} := \prod_{(k,l)\neq(i,j)} b_{kl}$, and $\tilde{a}_{ij} := \tilde{b}_{ij} a_{ij}$. Define

$$G_i := \sum_{j=0}^{n_i} \tilde{a}_{ij}(Y_1) Y_2 X^j \in K[Y_1, Y_2, X].$$

[5] This replacement is necessary as the field axioms do not have to be fulfilled.

6. Use Algorithm 4.3.11 with input K, $\vec{h} := (X_1, \iota(v))$, ι, $k_1 := 1$, $g_0^{(1)} := Y_1$ and for $i \in \{2, \ldots, n\}$ take $k_i := n_i$ and $g_{n_i-1}^{(i)}, \ldots, g_0^{(i)}$, which be the non-leading coefficients of G_i. Let $\tilde{\iota}$ be the output.

7. Apply Algorithm 4.3.16 to the input K, R, $x_1, \iota(v)(x_1) \in R$ and $\tilde{\iota}$, and define $f_1 \in K[X]$ as the output.

8. For each $i \in \{2, \ldots, n\}$ exchange x_1 with x_i and repeat the processes starting at Step 2 to get f_i instead of f_1. Then return $f_1, \ldots, f_n$.

Theorem 4.3.22 (Algorithmic version of Zariski's lemma). In the situation of Algorithm 4.3.21 we assume that K is a field, R is a K-algebra, $x_1, \ldots, x_n \in R$ and ι is an algebraic inverse function on $K[x_1, \ldots, x_n]$. Then $f_1(x_1) = \cdots = f_n(x_n) = 0$ and $f_1, \ldots, f_n$ are non-constant.

Proof. We use induction on n and consider the algorithm step by step. If $n = 0$, there is nothing to show. If $n = 1$, the statement follows by Lemma 4.3.15. If $n = 2$, the statement follows by Lemma 4.3.17.

In Step 2 we use Lemma 4.3.20 to obtain that L is a field, R is an L-algebra and ι' is an algebraic inverse function on $L[x_2, \ldots, x_n]$.

In Step 3 and 4 we have $F_2(x_2) = \cdots = F_n(x_n) = 0$ by the induction hypothesis. Note that F_i is indeed $\tilde{F}_i$ divided by its leading coefficient since L is a field.

In Step 5 we have $F_i = G_i(x_1, v^{-1}, X)$ as a polynomial in $R[X]$, and therefore $0 = F_i(x_i) = G_i(x_1, (v(x_1))^{-1}, x_i)$. Thus, the non-leading coefficients of G_i (as polynomials in X) witness that x_i is integral over $K[x_1, \iota(v)(x_1)]$ for each $i \in \{2, \ldots, n\}$.

In Step 6 we use Lemma 4.3.12. The requirements of this lemma are fulfilled by the notes to Step 5. Hence, $\tilde{\iota}$ is an algebraic inverse function on $K[x_1, \iota(v)(x_1)]$.

In Step 7 we get $f_1(x_1) = 0$ and f_1 is non-constant by Lemma 4.3.17.

In Step 8 $f_i(x_i) = 0$ and $f_i \neq 0$ for $i \in \{2, \ldots, n\}$ follows analogous to the case $i = 1$. □

4.4 Application: representation of maximal ideals in polynomial rings over algebraically closed fields

Motivation 4.4.1. In this section we consider the theorem that each maximal ideal in $K[\vec{X}]$ for an algebraically closed field K has the form $\langle X_1 - x_1, \ldots, X_n - x_n \rangle$ for some $x_1, \ldots, x_n \in K$. This theorem is proven by using Zariski's lemma and we

use the same approach as for the proof of Zariski's lemma described in Motivation 4.0.1.

However, being a maximal ideal is constructively a highly strong assumption. Hence, from an constructive point of view this theorem is not very valuable, but in the next section we illustrate how one could use the results in this section together with the results in Chapter 3 to get an algorithm for Hilbert's Nullstellensatz.

As before, we start by giving a constructive proof of the statement:

Proposition 4.4.2. Let K be an algebraically closed field and $M \subseteq K[\vec{X}]$ be a maximal ideal, then there are $y_1, \ldots, y_n \in K$ such that $M = \langle X_1 - y_1, \ldots, X_n - y_n \rangle$.

Proof. We consider the ring $R := K[\vec{X}]/M$. Obviously $R = K[X_i + M, \ldots, X_i + M]$. Since M is maximal, R is a field. By Zariski's lemma there are non-constant polynomials $f_1, \ldots, f_n \in K[X]$ with $f_i(X_i + M) = 0$, i.e. $f_i(X_i) \in M$. Since K is algebraically closed, $f_i = a_i \prod_j (X_i - x_{ij})$ for some $a_i, x_{ij} \in K$ and $a_i \neq 0$. Due to the fact that M is maximal and $a_i \neq 0$, for each i there is j_i with $X_i - x_{ij_i} \in M$. We define $x_i := x_{ij_i}$. Now, let $f \in M$ be given, then there are $g_1, \ldots, g_n \in K[\vec{X}]$ such that $f = f(x_1, \ldots, x_m) + \sum_{i=1}^n g_i(X_i - x_i)$, which are constructed in the remark below. Since $X_1 - x_1, \ldots, X_n - x_n \in M$, also $f(x_1, \ldots, x_m) \in M$. But M is maximal and $f(x_1, \ldots, x_m) \in K$, therefore $f(x_1, \ldots, x_m) = 0$ and $f \in \langle X_1 - x_1, \ldots, X_n - x_n \rangle$. □

Remark 4.4.3. Given elements $x_1, \ldots, x_n \in K$ and $f \in K[\vec{X}]$ one can compute $g_1, \ldots, g_n \in K[\vec{X}]$ with $f = f(y_1, \ldots, y_n) + \sum_{i=1}^n g_i(X_i - y_i)$ by recursion on n as follows: For $n = 0$, there is nothing to show. For $n > 0$, let $f = \sum_{i=0}^m a_i X_n^i$ with $a_i \in K[X_1, \ldots, X_{n-1}]$. Then we have

$$\begin{aligned} f =& \sum_{i=0}^{m} a_i (X_n - x_n + x_n)^i = \sum_{i=0}^{m} a_i \sum_{j=0}^{i} \binom{i}{j} (X_n - x_n)^j x_n^{i-j} \\ =& (X_n - x_n) \sum_{i=0}^{m} a_i \sum_{j=1}^{i} \binom{i}{j} (X_n - x_n)^{j-1} y_n^{i-j} + \sum_{i=0}^{m} a_i x_n^i \\ =& (X_n - x_n) \sum_{i=0}^{m} a_i \sum_{j=1}^{i} \binom{i}{j} (X_n - x_n)^{j-1} x_n^{i-j} + f(X_1, \ldots, X_{n-1}, x_n), \end{aligned}$$

and finalize by using recursion to $f(X_1, \ldots, X_{n-1}, x_n) \in K[X_1, \ldots, X_{n-1}]$.

Therefore, the computational challenge of Proposition 4.4.2 is basically to find the elements $x_1, \ldots, x_n \in K$. We continue by defining the structure on which the algorithm shall operate.

Definition 4.4.4. Let $(R, +, \cdot, 0, 1, -, =)$ be a ring structure and $M \subseteq R$ be a decidable subset of M, i.e. we decide whether $x \in M$ or $x \notin M$ in our algorithms. We define a new ring structure R/M by $R/M := (R, +, \cdot, 0, 1, -, =')$, where the set and all operators stay the same, except the equality, which is defined by $a =' b \Leftrightarrow a - b \in M$.

To avoid any confusion, we often use the notation $x + M$ for $x \in R$ if we want to make clear that we are considering the ring structure R/M. In particular, we write $a + M = b + M$ for $a - b \in M$.

Motivation 4.4.5. The next two definitions are the computational interpretation of a maximal ideal and an algebraically closed field. In both cases, the computational content consists out of the corresponding object and a witness function. In the case of a maximal object, the witness function provides evidence why an element is not in the maximal ideal. This is similar to the notion of an explicit maximal object in Definition 3.2.11. In the case of an algebraically closed field, the witness function allocates its zeros to each non-zero polynomial.

Definition 4.4.6. Let R be a discrete ring. We call a decidable subset $M \subseteq R$ together with a function $\nu : R \to R$ an *explicit maximal ideal*, if M is a proper ideal and $x\nu(x) - 1 \in M$ for all $x \in R \setminus M$.

Definition 4.4.7. We call a discrete field K together with a function $\Gamma : K[X] \to K^*$ an *explicit algebraically closed field* if $f = a\prod_{i=1}^{n}(X - x_n)$ for each $0 \neq f \in K[X]$ and $\Gamma(f) = [x_1, \ldots, x_n]$, where a is the leading coefficient of f.

Algorithm 4.4.8. Let a discrete field structure K, a map $\Gamma : K[X] \to K^*$, a decidable subset $M \subseteq K[\vec{X}]$ and a map $\nu : K[\vec{X}] \to K[\vec{X}]$ be given. We compute $x_1, \ldots, x_n \in K$ as follows:

1. We apply Algorithm 4.3.21 to the field structure K, the K-algebra structure $K[X]/M$, the map $\nu : K[\vec{X}] \to K[\vec{X}]$ and $X_1 + M, \ldots, X_n + M \in K[\vec{X}]/M$. Let $f_1, \ldots, f_n \in K[X]$ be the output.

2. For each i compute $\Gamma(f_i) = [x_{i1}, \ldots, x_{im_i}]$ and check if there is x_{ij} with $X_i - x_{ij} \in M$. If yes, define $x_i := x_{ij}$. If not, define $x_i := 0$.

3. Return the list $x_1, \ldots, x_n$.

Theorem 4.4.9. In the situation of Algorithm 4.4.8 we assume that K, Γ is an explicit algebraically closed field and M, ν is an explicit maximal ideal. Then $M = \langle X_1 - x_1, \ldots, X_n - x_n \rangle$, i.e. for a given $f \in M$ there are $g_1, \ldots, g_n \in K[\vec{X}]$ with $f = \sum_{i=1}^{n} g_i(X_i - x_i)$.

Proof. One can easily check that ν is an algebraic inverse function on $K[X_1 + M, \ldots, X_n + M]$. By Theorem 4.3.22 we have $f_i(X_i + M) = 0$ for all i. Therefore, $X_i + M$ must be equal to one of the zeros in the list $[x_{i1} + M, \ldots, x_{im_i} + M]$ by the property of an explicit algebraically closed field, and thus $X_i - x_{ij} \in M$ some j. Similar to Remark 4.4.3 and the end of the proof of Proposition 4.4.2, it follows $\langle X_1 - x_1, \ldots, X_n - x_n \rangle \subseteq M$. □

4.5 Outlook: an algorithm for Hilbert's Nullstellensatz

Motivation 4.5.1. We conclude this chapter by an outlook on how to use the constructed algorithms in conjunction with Algorithm 3.3.6 of Chapter 3 to get an algorithm for Hilbert's Nullstellensatz. In a nutshell, we use a kind of backtracking for the algorithms which we have developed above in this chapter.

However, this process is highly tedious and can not be carried out on paper in detail since the time and space for this dissertation is limited. Hence, in the following subsection we will just sketch how the approach works. Therefore, this section is marked as an outlook and we do not claim that everything in this section is proven in detail and without any doubt correct.

We want to compare this with the following situation: In a chess game, if black starts without its queen, everyone (who is familiar with chess) would be convinced that white (who still has its queen) must have a winning strategy. However, a proof of this is way to complex since there are more possible positions in chess than elementary particles in the visible universe. In contrast to this, we are convinced that our approach can be formalised without using each elementary particle in the visible universe and even in lifetime, maybe by using computer support.

Our result is similar to [57]. In Section 2 of this paper Coste, Lombardi and Roy developed an algorithm for Hilbert's Nullstellensatz. A constructive proof can also be found in [121, Section 3 of Chapter VIII]. We start our outlook by formulating a version of Hilbert's Nullstellensatz and providing the classical proof which comes from Oskar Zariski [186]. We first consider the weak version of Hilbert's Nullstellensatz because the strong version is a corollary of the weak version, where the standard proof is already constructive as we see in Corollary 4.5.23.

Theorem 4.5.2 (Hilbert's Nullstellensatz). Let K be an algebraically closed field and $f_1, \ldots, f_m \in K[\vec{X}]$ be given. Then either there are $g_1, \ldots, g_m \in K[\vec{X}]$ such that $g_1 f_1 + \cdots + g_m f_m = 1$ or there are $\vec{x} \in K$ with $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$.

Proof. Assume $g_1f_1+\cdots+g_mf_m \neq 1$ for any $g_1,\ldots,g_m \in K[\vec{X}]$, then $\langle f_1,\ldots,f_m\rangle \subseteq K[\vec{X}]$ is a proper ideal, and hence there is a maximal ideal M which contains $f_1,\ldots,f_m$. Then $K[\vec{X}]/M$ is a field and a finitely generated K-algebra. By Zariski's lemma, each $X_i + M \in K[\vec{X}]/M$ is algebraic over K. Hence, there are monic $h_1,\ldots,h_n \in K[X]$ with $h_i(X_i + M) = 0$. Since K is algebraically closed, $h_i = \prod_j (X - y_{ij})$ for all i and finitely many $y_{ij} \in K$. Therefore, for each i there is an j_i with $X_i + M - y_{ij_i} = 0$, i.e. $X_i - y_i \in M$ for $y_i := y_{ij_i}$. Since also $f_1,\ldots,f_m \in M$, it follows $f_i(y_1,\ldots,y_n) = 0$ for all i similar to the proof of Proposition 4.4.2 and Remark 4.4.3. □

Remark 4.5.3. One sees that this proof is an extension of the proof of Proposition 4.4.2. The difference is that in this proof the maximal ideal is not already given and therefore we need transfinite methods. However, we only need maximal ideals in the proof of Hilbert's Nullstellstsatz but not in its statement. This is quite similar to the universal Krull-Lindenbaum lemma given in Section 3.4.

4.5.1 The material interpretation of Zariski's lemma

Motivation 4.5.4. The algorithmic version of Zariski's lemma has the following form: If some axioms about the structures in the algorithm (like the field axioms, algebra axioms and the property of an algebraic inverse functions) are fulfilled then the computed $f_1,\ldots,f_n$ are non-constant and $f_i(x_i) = 0$ for all i.

In order to use Algorithm 3.3.6 to approximate the maximal ideal which we need in the proof of Hilbert's Nullstellensatz, we have to make a so-called *material interpretation* of the algorithmic version. In particular, instead of proving $A \Rightarrow B$ we prove a strong version of $\neg A \vee B$. From a computational point of view, the second statement is stronger, since we at least have to decide which side of the disjunction holds. In our case we take a stronger form of the negation $\neg A$ as we need computational information also in this case. For example, one of the assumption is that a field structure K is indeed a field. The field axioms are universal statements. Hence, the constructive interpretation of not being a field is a marker with the number of the axiom which is not fulfilled and a list of elements which provide the counterexample. For instance, a counterexample to the axiom of multiplicative commutativity in a ring R is a list $[a, b]$ of two elements in R such that $ab \neq ba$. Of course, there is also the field axiom $1 \neq 0$. A counterexample to this axiom is simple the empty list.

The notion "material interpretation" comes from the material implication which is defined as a truth function by using truth tables [119].

Definition 4.5.5. Here and in the rest of this chapter let a numeration of the field axioms, ring axioms and algebra axioms be given.

For a field structure K we say that there is an *evidence that K is not a field* if there is a concrete counterexample that one of the field axioms is not fulfilled. Such a counterexample consists of the number i of the not fulfilled axiom and a list of elements in K with constitute this counterexample. Analogously, we define the notion that there is an *evidence that R is not a ring* for a given ring structure and that there is an *evidence that R is not a K-algebra* for a given field structure K, a given ring structure R and a map from K to R (i.e. a K-algebra structure R).

For a given field structure K, $\vec{x} \in K$ and a map $\iota : K[\vec{X}] \to K[\vec{X}]$, an *evidence that ι is not an algebraic inverse function on $K[\vec{x}]$* is an $f \in K[\vec{X}]$ such that $f(\vec{x}) \neq 0$ and $f(\vec{x})(\iota(f))(\vec{x}) - 1 \neq 0$.

Remark 4.5.6. Note that an evidence that R is not a K-algebra can also be a concrete counterexample that the map from K to R is not a homomorphism although this map is omitted in the notation.

Motivation 4.5.7. Using the notions from Definition 4.5.5, we are now able to formulate the material interpretation of the lemmas in this chapter. After each reformulated lemma, we give an approach for a possible proof which contains also several comments. If the proof of the reformulated lemma or theorem is not complete, we just call them "claim" or "conjecture", respectively. We start with Lemma 4.3.15 since this lemma is the most simple one in this chapter and we can even give a proper proof of it. However, one sees that even for this simple lemma, the proof of the material interpretation becomes astonishing long.

Lemma 4.5.8. In the situation of Algorithm 4.3.14 let $f = 0$ or $f(x) \neq 0$. Then one of the following statements holds:

- There is evidence that K is not a field.
- There is evidence that R is not a K-algebra.
- There is evidence that ι is not an algebraic inverse function on $K[x]$.

Proof. As in Algorithm 4.3.14 we consider the cases $x = 0$ and $x \neq 0$:

In the first case $f = X$, which is an abbreviation for $1X$, and therefore if $f = 0$ it follows $1 = 0$ in K which gives an evidence that K is not a field. If $f(x) \neq 0$ then $1 \cdot 0 \neq 0$ in R. But this provides a counterexample to the axiom that 1 is the neutral element of the multiplication.

In the second case $f := X\iota(X) - 1$. First we assume $f = 0$ and consider the constant coefficients of this polynomial equation and receive $-1 = 0$ in K. It follows that either $-1 + 1 = 0 - 1$ or we have a counterexample that the equality is not compatible with the addition. Hence, either we have a counterexample to the axiom that $-$ is the additive inverse function and we are done, or $-1 + 1 = 0$, and hence either we have a counterexample that to the axiom that 0 is the neutral element of the addition or $0 - 1 = 0$. Together, either we have a counterexample that the equality is not transitive, or $0 = 1$. Finally, either there is a counterexample to the symmetry axiom of the equality or $1 = 0$ and we have a counterexample to the axiom $1 \neq 0$.

In the second case we have $f(x) \neq 0$, and it follows either $f(x) := (X\iota(X) - 1)(x) = x\iota(X)(x) - 1$ or we get a counterexample to one of the ring axioms. (Details are left to the reader.) In the last case we are done. In the first case we have either $x\iota(X)(x) - 1 = 0$ and get an counterexample to the transitivity of the equality or there is evidence that ι is not an algebraic inverse function. □

Claim 4.5.9. In the situation of Algorithm 4.3.9 let a K-algebra structure R and $\vec{x}, \vec{y} \in R$ be given. Assume

$$(f(\vec{x}))^k + g_{k-1}(\vec{y})(f(\vec{x}))^{k-1} + \cdots + g_0(\vec{y}) \neq 0,$$

then one of the following statements holds:

- There is evidence that K is not a field.
- There is evidence that R is not a K-algebra.
- There is an i with $x_i^{k_i} + g_{k_i-1}^{(i)}(\vec{y})x_i^{k_i-1} + \cdots + g_0^{(i)}(\vec{y}) \neq 0$

Proof attempt. This lemma is the material interpretation of Lemma 4.3.10. It can be proven by backtracking the proof of that lemma and Algorithm 4.3.9. Since we deal with modules, one has to define the concepts of a *module structure* and an *evidence that a module structure is not a module* or at least use them implicitly.

An relevant moment here is that the proof of this lemma uses the theorem of Cayley-Hamilton. Hence, one also has to backtrack a proof of this theorem. The proof given in [69, Theorem 4.3] is constructive and can be specialised to our case. □

Claim 4.5.10. In the situation of Algorithm 4.3.11 let R be a ring structure and assume there is evidence that $\tilde{\iota}$ is not an algebraic inverse function on $K[\vec{y}]$, then one of the following statements holds:

- There is evidence that K is not a field.
- There is evidence that ι is not an algebraic inverse function on $K[\vec{x}]$
- There is an i with $h_i(\vec{x}) \neq y_i$.
- There is an i with $x_i^{k_i} + g_{k_i-1}^{(i)}(\vec{y})x_i^{k_i-1} + \cdots + g_0^{(i)}(\vec{y}) \neq 0$.

Proof attempt. This lemma is the material interpretation of Lemma 4.3.12. Therefore, it can be proven by using backtracking of that lemma and Algorithm 4.3.11 and Claim 4.5.9. □

Claim 4.5.11. In the situation of Algorithm 4.3.16 we assume that $f_1 = 0$, $f_1(x_1) \neq 0$, $f_2 = 0$ or $f_2(x_2) \neq 0$. Then one of the following statements holds:

- There is evidence that K is not a field.
- There is evidence that R is not a K-algebra.
- There is evidence that ι is not an algebraic inverse function on $K[x_1, x_2]$.

Proof attempt. This lemma is the material interpretation of Lemma 4.3.17. The proof in done by backtracking the proof of this lemma and Algorithm 4.3.16 and by using Lemma 4.5.8 and Claim 4.5.10. □

Claim 4.5.12. In the situation of Algorithm 4.3.18 we assume that one of the following assumptions holds:

- There is evidence that L is not a field.
- There is evidence that R is not an L-algebra.
- There is evidence that $\tilde{\iota}$ is not an algebraic inverse function on $L[x_2, \ldots, x_n]$.

Then one of the following statements holds:

- There is evidence that K is not a field.
- There is evidence that R is not a K-algebra.
- There is evidence that ι is not an algebraic inverse function on $K[\vec{x}]$.

Proof attempt. This lemma is the material interpretation of Lemma 4.3.20. One needs case distinction on the several cases of the assumption. Each field axiom and each algebra axiom results in an own case. For the proof one uses backtracking of Algorithm 4.3.18 and the proof of Lemma 4.3.20. □

Conjecture 4.5.13 (Material interpretation of Zariski's lemma). In the situation of Algorithm 4.3.21 we assume that there is an i with $f_i = 0$ or $f_i(x_i) \neq 0$. Then one of the following statements holds:

- There is evidence that K is not a field.
- There is evidence that R is not a K-algebra.
- There is evidence that ι is not an algebraic inverse function on $K[\vec{x}]$.

Proof attempt. This lemma is the material interpretation of Theorem 4.3.22. The proof uses backtracking of Algorithm 4.3.21 and the proof of Theorem 4.3.22 as well as Claim 4.5.10, Lemma 4.5.8, Claim 4.5.11 and Claim 4.5.12. □

Conjecture 4.5.14. Let an explicit algebraically closed field K, Γ, a decidable subset $M \subseteq K[\vec{X}]$, a map $\nu : K[\vec{X}] \to K[\vec{X}]$ and polynomials $f_1, \ldots, f_m \in M$ be given. We apply these objects to Algorithm 4.4.8 and $\vec{x}$ be the output. Then either $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$ or we have evidence that M, ν is not an explicit maximal ideal. In particular, either $1 \in M$, or there is an $f \notin M$ with $f\nu(f) - 1 \notin M$, or there are a finite subset $A \subseteq M$, $a_1, \ldots, a_k \in A$ and $\lambda_1, \ldots, \lambda_k \in K[\vec{X}]$ with $\lambda_1 a_1 + \cdots + \lambda_k a_k \notin M$.

Proof attempt. One part of Theorem 4.4.9 is the following statement:

> In the situation of Algorithm 4.4.8 we assume that K, Γ is an explicit algebraically closed field and M, ν is an explicit maximal ideal, then $X_1 - x_1, \ldots, X_n - x_n \in M$.

A semi-material[6] interpretation of this theorem is the following:

> In the situation of Algorithm 4.4.8 we assume that K, Γ is an explicit algebraically closed field. Then if $X_i - x_i \notin M$ for one $i \in \{1, \ldots, n\}$, there is evidence that M, ν is not an explicit maximal object.

This can be proven by backtracking (the first part of) the proof of Theorem 4.4.9 and Algorithm 4.4.8 and using Conjecture 4.5.13. Out of the construction in Remark 4.4.3 we get the following statement:

[6]Here we say "semi-material" since we still assume that K, Γ is an explicit algebraically closed field.

Let K be a field, $\vec{x} \in K$, $M \subseteq K[\vec{X}]$ and $f_1, \ldots, f_m \in M$. Then either $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$, or there is $i \in \{1, \ldots, n\}$ with $X_i - x_i \notin M$, or there is evidence that M is not a proper ideal, i.e. $1 \in M$ or there are a finite $A \subseteq M$, $a_1, \ldots, a_k \in A$, $\lambda_1, \ldots, \lambda_k \in K[\vec{X}]$ with $\lambda_1 a_1 + \cdots + \lambda_k a_k \notin M$.

The theorem follows by combining these two statements. □

4.5.2 Approximate maximal objects in Hilbert's Nullstellensatz

Motivation 4.5.15. In this subsection we show how the result of the last section combined with the state algorithm of Section 3.3 can be used to find an algorithm for Hilbert's Nullstellensatz if the field is discrete and the corresponding polynomial ring is countable. Note that in Chapter 3 we assumed globally that the underlying sets in the algorithms are decidable but here we make such assumptions explicit.

From here on we are able to prove everything in detail. To be absolutely precise, **we consider Conjecture 4.5.14 as a global assumption of this section**. Hence, the reader can ignore everything else in the last section.

We start by applying the theory in Chapter 3 to the case here.

Definition 4.5.16. In the following for a given ring R we define the covering $\rhd \subseteq \mathcal{P}_{fin}(R) \times R$ by

$$A \rhd x :\Leftrightarrow \exists_{\vec{a},\vec{b} \in R^*} \left(\vec{a} \in A^* \wedge \vec{a} \cdot \vec{b} = x \right),$$

where $\vec{a} \cdot \vec{b} := \sum_{i=0}^{|\vec{a}|-1} a_i b_i$ and $b_i = 0$ for $i \geq |\vec{b}|$.

Furthermore, we define the predicate θ on R by $\theta(x) := (x \neq 1)$.

Remark 4.5.17. For a given ring R the $\rhd$-ideals are exactly the (ring theoretic) ideals of R.

The witness type of this covering is $W := R^* \times R^*$ and we write

$$A \rhd_{\vec{a},\vec{b}} x \qquad \text{for} \qquad \vec{a} \in A^* \wedge \vec{a} \cdot \vec{b} = x.$$

Hence, if R is countable and discrete, $\rhd$ is a Σ^0_1-covering.

In Section 3.5 we gave a similar cover structure on a ring R with the difference that instead of $\vec{a} \cdot \vec{b} = x$, we had $\vec{a} \cdot \vec{b} = x^e$, where e is a natural number. Then the ideals in Section 3.5 are the radical ideals of R. In that case we were able to prove encoding, where the operator $\circ$ is the ring multiplication, and by using encoding

we have proved Krull's lemma. In the current case, it can be proven easily that $\rhd$ is reflexive and transitive, but encoding for $\rhd$ and the ring multiplication does not hold. For example, in $R = \mathbb{Z}$ we have $2 \rhd 2$ but not $4 \rhd 2$.

As observed in Remark 3.2.8, in our case the predicate R_A^F given in Notation 3.2.7 is classified by

$$R_S^F(A, (\vec{a}, \vec{\lambda}), g) \Leftrightarrow \left(A \subseteq F \cup S \wedge \vec{a} \in A^* \wedge \vec{\lambda} \cdot \vec{a} = g \Rightarrow g \neq 1\right).$$

In particular, the type $\mathcal{G}$ from Notation 3.2.7 is given by

$$\mathcal{G} = \mathcal{P}_{fin}(R) \times (R^* \times R^*) \times R.$$

In the other direction, we have

$$\neg R_S^F(A, (\vec{a}, \vec{\lambda}), g) \Leftrightarrow A \subseteq F \cup S \wedge \vec{a} \in A^* \wedge \vec{\lambda} \cdot \vec{a} = g \wedge g = 1.$$

Motivation 4.5.18. Our goal is to develop an algorithmic version of Hilbert's Nullstellensatz by using Algorithm 3.3.6. Therefore, we have to define the functions ω and ϕ in a pleasant way out of an explicit algebraically closed field and polynomials $f_1, \ldots, f_m \in K[\vec{X}]$. This construction is given in the next algorithm. Our approach is quite similar to the approach in Section 3.4. In particular, in Definition 3.4.10 and Lemma 3.4.12, we have defined the functions ω and ϕ in the context of the universal Krull-Lindenbaum Lemma.

Algorithm 4.5.19. Let K, Γ be an explicit algebraically closed field, $K[\vec{X}] := \{r_n \mid n \in \mathbb{N}\}$ an enumeration of $K[\vec{X}]$ and $f_1, \ldots, f_m \in K[\vec{X}]$. For a given decidable subset $M \subseteq K[\vec{X}]$ and a map

$$F : \mathrm{dom}(K[\vec{X}]/M) \to \mathcal{P}_{fin}(K[\vec{X}]) \times (K[\vec{X}]^* \times K[\vec{X}]^*) \times K[\vec{X}],$$

we define

$$\omega(M, F) \in \mathbb{Z} \quad \text{and} \quad \phi(M, F) \in \mathcal{P}_{fin}(K[\vec{X}]) \times (K[\vec{X}]^* \times K[\vec{X}]^*) \times K[\vec{X}]$$

as follows:

1. We define $\nu : K[\vec{X}] \to K[\vec{X}]$ for given $h \in K[\vec{X}]$ as follows:
   - If $h \in M$, define $\nu(h) := 0$
   - If $h \notin M$, take n with $h = r_n$ and compute $(A, (\vec{a}, \vec{\lambda}), g) := F(n)$. Let I be the set of exactly those $i < |\vec{a}|$ with $a_i = h$ and define $\nu(h) := \sum_{i \in I} \lambda_i$, where $\lambda_i := 0$ for $i \geq |\vec{\lambda}|$.

2. Apply Algorithm 4.4.8 to K, Γ, M and ν, and let $\vec{x}$ be the output.

3. If $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$, define $\omega(M,F) := -1$ and $\phi(M,F) := (\varnothing, [], [], 1)$.

4. Otherwise, by Conjecture 4.5.14 we have three cases:

   - If there is $1 \in M$, let $k \in \mathbb{N}$ be given such that $r_k = 1$. Then $\phi(M,F) := (\{1\}, (1,1), 1)$ and if $1 \in \{f_1, \ldots, f_m\}$, define $\omega(M,F) := -1$, otherwise $\omega(M,F) := k$.
   - If there is an $f \notin M$ with $f\nu(f) - 1 \notin M$, we take k such that $r_k = f$ and compute $(B_1, (\vec{b}^{(1)}, \vec{\sigma}^{(1)}), g) := F(k)$. Define the sets $B_1' := B_1 \setminus \{f\}$ and $I := \{i \leq |\vec{b}^{(1)}| \mid b_i^{(1)} = f\}$. Erase all $b_i^{(1)}$ from $\vec{b}^{(1)}$ and all $\sigma_i^{(1)}$ from $\vec{\sigma}^{(1)}$ with $i \in I$ to get the lists $\vec{b}^{(3)}$ and $\vec{\sigma}^{(3)}$. Furthermore, take l such that $r_l = f\nu(f) - 1$ and compute $(B_2, (\vec{b}^{(2)}, \vec{\sigma}^{(2)}), h) := F(l)$. Define $B_2' := B_2 \setminus \{f\nu(f) - 1\}$ and $J := \{j \leq |\vec{b}^{(2)}| \mid b_j^{(2)} = f\nu(f) - 1\}$. Erase $b_j^{(2)}$ from $\vec{b}^{(2)}$ for all $j \in J$ to get the new list $\vec{b}^{(4)}$, and expand $\vec{b}^{(4)}$ by the list $\vec{b}^{(3)}$ to get a new list $\vec{b}'$. Then erase $\sigma_j^{(2)}$ from the list $\vec{\sigma}^{(2)}$ for all $j \in J$ to get the new list $\vec{\sigma}^{(4)}$ and expand $\vec{\sigma}^{(4)}$ by the list $\left(-\sum_{j\in J} \sigma_j^{(2)}\right)\vec{\sigma}^{(3)}$, where the multiplication is meant to be componentwise, to get the new list $\vec{\sigma}'$. Then define

   $$\omega(M,F) := \max\{i \in \mathbb{N} \mid r_i \in (B_1' \cup B_2') \setminus \{f_1, \ldots, f_m\}\}$$

   and

   $$\phi(M,F) := (B_1' \cup B_2', (\vec{b}', \vec{\sigma}'), 1).$$

   - If there is $B_1 \subseteq M$ and a linear combination $\sigma_1^{(1)} b_1^{(1)} + \cdots + \sigma_k^{(1)} b_k^{(1)} \notin M$ with $b_i^{(1)} \in B_1$ and $\sigma_i^{(1)} \in K[\vec{X}]$ for all $i \leq k$, let l be given such that $r_l = \sigma_1^{(1)} b_1^{(1)} + \cdots + \sigma_k^{(1)} b_k^{(1)}$. We compute $(B_2, (\vec{b}^{(2)}, \vec{\sigma}^{(2)}), g) := F(l)$. Let $B_2' := B_2 \setminus \{r_l\}$ and $I := \{i \leq k \mid b_i^{(2)} = r_l\}$. Replace all $b_i^{(2)}$ with $i \in I$ in $\vec{b}^{(2)}$ by $b_1^{(1)}, \ldots, b_k^{(1)}$ and all $\sigma_i^{(2)}$ with $i \in I$ in $\vec{\sigma}^{(2)}$ by $\sigma_i^{(2)}\sigma_1^{(1)}, \ldots, \sigma_i^{(2)}\sigma_k^{(1)}$ to get the new lists $\vec{b}'$ and $\vec{\sigma}'$, respectively. Define

   $$\omega(M,F) := \max\{i \in \mathbb{N} \mid r_i \in (B_1 \cup B_2') \setminus \{f_1, \ldots, f_n\}\}$$

   and

   $$\phi(M,F) := (B_1 \cup B_2', (\vec{b}', \vec{\sigma}'), 1).$$

Theorem 4.5.20 (Algorithmic version of Hilbert's Nullstellensatz). Let an explicit algebraically closed field K, Γ be given such that $K[\vec{X}] = \{r_n \mid n \in \mathbb{N}\}$ is countable. Furthermore, let $f_1, \ldots, f_m \in K[\vec{X}]$ be given. Then either there are computable $\vec{x} \in K$ with $f_1(\vec{x}) = f_m(\vec{x}) = 0$ or there are computable $g_1, \ldots, g_m \in K[\vec{X}]$ with $\sum_{i=1}^{m} g_i f_i = 1$.

Proof. We apply Algorithm 3.3.6 to the set $K[\vec{X}]$, the subset $\{f_1, \ldots, f_m\}$ the covering and predicate $\rhd$, θ given in Definition 4.5.16 and the functions ω, ϕ given in Algorithm 4.5.19. By Theorem 3.3.12 this algorithm terminates in an end state π_L and we define $M := M[\pi_L]$ and $F := f[\pi_L]$. Furthermore, let

$$\Omega := \omega(M, F) \quad \text{and} \quad (A, (\vec{a}, \vec{\lambda}), g) := \phi(M, F).$$

By Theorem 3.3.8 (M, F) is an approximate explicit maximal object. Therefore, for each $k \leq \Omega$ we have

$$r_k \in M \Rightarrow \Big(A \subseteq [M](k) \cup \{r_k, f_1, \ldots, f_m\} \wedge \vec{a} \in A^* \wedge \vec{a} \cdot \vec{\lambda} = g \Rightarrow g \neq 1\Big)$$

and

$$r_k \notin M \Rightarrow B \subseteq [M](k) \cup \{r_k, f_1, \ldots, f_m\} \wedge \vec{b} \in B^* \wedge \vec{b} \cdot \vec{\sigma} = h \wedge h = 1,$$

where $(B, (\vec{b}, \vec{\sigma}), h) := F(k)$, and by Lemma 3.3.7 the last formula even holds for all $k \in \mathbb{N}$.

Following Algorithm 4.5.19 one sees that the last component of ϕ is constantly 1. Hence, by the construction of M, F in Algorithm 3.3.6 the last component of F is also constantly 1. Therefore, the two formulas above are equivalent to the following formulas:

$$r_k \in M \Rightarrow \neg \Big(A \subseteq [M](k) \cup \{r_k, f_1, \ldots, f_m\} \wedge \vec{a} \in A^* \wedge \vec{a} \cdot \vec{\lambda} = 1\Big) \tag{4.3}$$

for all $k \leq \Omega$, and

$$r_k \notin M \Rightarrow B \subseteq [M](k) \cup \{r_k, f_1, \ldots, f_m\} \wedge \vec{b} \in B^* \wedge \vec{b} \cdot \vec{\sigma} = 1 \tag{4.4}$$

for all $k \in \mathbb{N}$ and $(B, (\vec{b}, \vec{\sigma}), h) := F(k)$. Furthermore, looking at Algorithm 4.5.19 we see that if the second component of F constantly consists of two lists of the same length then also the second component of ϕ constantly consists of two list of the same length. Hence, by induction on the construction in Algorithm 3.3.6, the second component of each $F(k)$ and $\phi(M, F)$ consists of two lists of the same length. In

the following, we will assume this tacitly. To prove the statement, we consider the cases in Algorithm 4.5.19, i.e. in the definition of $\omega(M,F)$ and $\phi(M,F)$.

We define ν analogously as in the first step of Algorithm 4.5.19 and apply Algorithm 4.4.8 to K, Γ, M and ν and let $\vec{x}$ be the output. If $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$, there is nothing to show. Otherwise, we apply Conjecture 4.5.14 and consider the three cases one by one:

- If $1 \in M$, we have $\phi(M,F) = (\{1\},([1],[1]),1)$. Then either $1 \in \{f_1,\ldots,f_m\}$ and there are obviously $g_1,\ldots,g_m$ with $\sum_{i=1}^m f_i g_i = 1$, or $1 \notin \{f_1,\ldots,f_m\}$ and $\omega(M,F) = k$ for $r_k = 1$. However, the last case cannot occur since it would contradict to (4.3) as one can check easily.

- If there is an $f \in K[\vec{X}]$ with $f \notin M$ and $f\nu(f) - 1 \notin M$. We take $k,l \in \mathbb{N}$ with $r_k = f$ and $r_l = f\nu(f) - 1$ and define $(B_1,(\vec{b}^{(1)},\vec{\sigma}^{(1)}),1) := F(k)$, $(B_2,(\vec{b}^{(2)},\vec{\sigma}^{(2)}),1) := F(l)$. By (4.4) we obtain

$$B_1 \subseteq M \cup \{f,f_1,\ldots,f_m\} \wedge \vec{b}^{(1)} \in B_1^* \wedge \vec{b}^{(1)} \cdot \vec{\sigma}^{(1)} = 1 \qquad \text{and} \tag{4.5}$$

$$B_2 \subseteq M \cup \{f\nu(f) - 1, f_1,\ldots,f_m\} \wedge \vec{b}^{(2)} \in B_2^* \wedge \vec{b}^{(2)} \cdot \vec{\sigma}^{(2)} = 1. \tag{4.6}$$

  We follow the construction of the second case in Step 4 of Algorithm 4.5.19 and use the same notation. By $\vec{b}^{(1)} \cdot \vec{\sigma}^{(1)} = 1$ and the construction of $\vec{b}^{(3)}$, $\vec{\sigma}^{(3)}$ and ν, we have

$$\vec{b}^{(3)} \cdot \vec{\sigma}^{(3)} + f\nu(f) = 1,$$

  and by $\vec{b}^{(2)} \cdot \vec{\sigma}^{(2)} = 1$ and the constructions $\vec{b}^{(4)}$, $\vec{\sigma}^{(4)}$ and J we have

$$\vec{b}^{(4)} \cdot \vec{\sigma}^{(4)} + \left(\sum_{j \in J} \sigma_j^{(2)}\right)(f\nu(f) - 1) = 1.$$

  Together with the formula before it follows

$$\vec{b}^{(4)} \cdot \vec{\sigma}^{(4)} + \left(\sum_{j \in J} \sigma_j^{(2)}\right)(-\vec{b}^{(3)} \cdot \vec{\sigma}^{(3)}) = 1.$$

  By the construction of $\vec{b}'$ and $\vec{\sigma}'$ this is exactly $\vec{b}' \cdot \vec{\sigma}' = 1$. Furthermore, by the construction of ω, ϕ, B_1' and B_2' and by (4.5) and (4.6) it follows

$$A = B_1' \cup B_2' \subseteq M \cup \{f_1,\ldots,f_m\} \wedge \vec{a} = \vec{b}' \in A^*.$$

Hence, it suffices to show $A \subseteq \{f_1, \ldots, f_m\}$: If not, $\Omega = \max\{i \in \mathbb{N} \mid r_i \in (A \setminus \{f_1, \ldots, f_m\})\} \geq 0$. Using (4.3) and noting $\vec{\lambda} = \vec{\sigma}'$, we have $r_\Omega \notin M$. But $r_\Omega \in B_1' \cup B_2' \subseteq M \cup \{f_1, \ldots, f_m\}$ and $r_\Omega \notin \{f_1, \ldots, f_m\}$, and therefore $r_\Omega \in M$, a contradiction.

- If there is $B_1 \subseteq M$ and a linear combination $\sigma_1^{(1)} b_1^{(1)} + \cdots + \sigma_k^{(1)} b_k^{(1)} \notin M$ with $b_i^{(1)} \in B_1$ and $\sigma_i^{(1)} \in K[\vec{X}]$ for all $i \leq k$, we take $l \in \mathbb{N}$ with $r_l = \sigma_1^{(1)} b_1^{(1)} + \cdots + \sigma_k^{(1)} b_k^{(1)}$ and $(B_2, (\vec{b}^{(2)}, \vec{\sigma}^{(2)}), 1) := F(l)$. Then by Formula 4.4 we get

$$B_2 \subseteq [M](l) \cup \{r_l, f_1, \ldots, f_m\} \wedge \vec{b}^{(2)} \in B_2^* \wedge \vec{b}^{(2)} \cdot \vec{\sigma}^{(2)} = 1.$$

We follow the construction in the third case in Step 4 of Algorithm 4.5.19 and use the same notation. By $r_l = \sigma_1^{(1)} b_1^{(1)} + \cdots + \sigma_k^{(1)} b_k^{(1)}, \vec{b}^{(2)} \cdot \vec{\sigma}^{(2)} = 1$ and the constriction of $\phi(M, F), \vec{b}', \vec{\sigma}'$ and B_2' we get $\vec{a} \cdot \vec{\lambda} = \vec{b}' \cdot \vec{\sigma}' = 1$,$A = B_1 \cup B_2' \subseteq M \cup \{f_1, \ldots, f_m\}$ and $\vec{a} \in A^*$. Hence, it suffices to show $A \subseteq \{f_1, \ldots, f_m\}$: If not, $\Omega = \omega(M, F) \geq 0$ and $r_\Omega \notin M$ by (4.3) and the definition of ω. But by definition $r_\Omega \in (B_1 \cup B_2') \setminus \{f_1, \ldots, f_m\} \subseteq M \setminus \{f_1, \ldots, f_m\}$, which is a contradiction.

□

Remark 4.5.21. By using Algorithm 3.3.6 we have managed to avoid the usage of maximals ideals in our proof, although we still need a constructively quite strong assumption, namely the existence of an explicit algebraically closed field. However, in [121, Theorem 3.5] it is shown that to each countable discrete field (e.g. $\mathbb{Q}$ and all finite fields) there exists a discrete algebraic closure. Following the proofs of [121, Theorem 3.5] and the corresponding lemmas, we get a construction of an explicit algebraically closed field. Therefore, the remaining assumption in the algorithmic version of the theorem above is constructively attainable.

Motivation 4.5.22. Similar to what Zariski has done in [186], we even obtain an algorithm for the strong version of Hilbert's Nullstellensatz. Here, the standard proof which uses the Rabinowitsch trick [139] is already constructive and in this way we get an algorithmic version of Hilbert's strong Nullstellensatz:

Corollary 4.5.23 (Algorithmic version of Hilbert's strong Nullstellensatz). Let an explicit algebraically closed field K, Γ be given such that $K[\vec{X}, Y] = \{r_n \mid n \in \mathbb{N}\}$ is countable. Furthermore, let $f_1, \ldots, f_m, f \in K[\vec{X}]$ be given. Then either there are computable $\vec{x} \in K$ with $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$ but $f(\vec{x}) \neq 0$, or there are computable $g_1, \ldots, g_m \in K[\vec{X}]$ and $r \in \mathbb{N}$ such that $\sum_{i=1}^m g_i f_i = f^r$.

Proof. We apply Theorem 4.5.20 to $f_1, \ldots, f_m, 1 - Yf$ and have two cases:

In the first case, there are computable $\vec{x}, y \in K$ with $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 1 - yf(\vec{x}) = 0$. In particular, $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$ and $1 = yf(\vec{x})$, and therefore $f(\vec{x}) \neq 0$.

In the second case there are computable $g'_1, \ldots, g'_m, g' \in K[\vec{X}, Y]$ with

$$\sum_{i=1}^{m} g'_i f_i + g'(1 - Yf) = 1.$$

We consider the homomorphism $K[\vec{X}, Y] \to K(\vec{X})$ given by $X_i \mapsto X_i$ for all i, $Y \mapsto \frac{1}{f}$ and $a \mapsto a$ for all $a \in K$. Applying this homomorphism to the equation above leads to

$$\sum_{i=1}^{m} g'_i \left(\vec{X}, \frac{1}{f}\right) f_i + g' \left(1 - \frac{1}{f} f\right) = 1.$$

Let $r := \max\{\deg_Y g'_1, \ldots, \deg_Y g'_m\}$. Then one can compute $g_1, \ldots, g_m \in K[\vec{X}]$ such that $g'_i \left(\vec{X}, \frac{1}{f}\right) = \frac{g_i}{f^r}$ for all i. Together with the equation above we obtain

$$\sum_{i=1}^{m} g_i f_i = f^r.$$

□

Chapter 5

Limits with signed digit streams

Motivation 5.0.1. We work on the signed digit representation of abstract real numbers, which is roughly the binary representation enriched by the additional digit -1. The main object of this chapter is the formalisation of a constructive proof that the signed digit representation of real numbers between -1 and 1 is closed under limits of converging sequences. The approach is quite similar to the approach in Chapter 4 where we formulate an algorithmic version of Zariski's lemma. Again, we start by formulating a constructive proof. However, in contrast to Chapter 4 we formalise the proof in the proof assistant Minlog. Here the other steps, which were done by hand in Chapter 4, are conducted automatically by Minlog. In particular, Minlog formulates a computational version of our theorem using the *reliability predicate*, and Minlog computes an algorithm out of the constructive proof: the *extracted term*. Finally, Minlog automatically proves that the extracted term fulfils the realisability predicate, this is called the *soundness proof*. The tools of this process are described in Section 2.1.

The constructive proof uses a *coinductive definition* of the signed digit representation. Coinductive definitions are one of Minlog's speciality as Minlog is based on the theory of computational functional **TCF** which is introduced in Section 2.1.

To apply the extracted terms, the programming language Haskell is useful. Therefore, after each proof we show a notation of the extracted term, which can be rewritten as a definition in Haskell.

Minlog can be downloaded at [122]. On this web page it is also demonstrated how to install Minlog. After installing Minlog a tutorial for Minlog is given in `doc/tutor.pdf` of the installed file. Another introduction to Minlog is presented in [178, 179].

The convergence theorem, which is the main theorem of this chapter, is implemented in `sdlim.scm` in the folder `examples/analysis` of the Minlog file. We use

Minlog's automatic program extraction to develop the computational content out of the implemented proofs.

As a first application of the extracted term we use the algorithm together with Heron's method to build up an algorithm which converts the signed digit representation of a non-negative real number into the signed digit representation of its square root. This section is not formalised in Minlog but we use the proofs in this section together with the extracted term of the convergence theorem to develop a Haskell program by hand. This Haskell program is added to the appendix and it is an expansion of the Haskell program in [178]. As a second application we construct an algorithm which takes the signed digit code of two real numbers and returns a signed digit code of their product.

5.1 Preliminaries

5.1.1 Historical notes and sources

Motivation 5.1.1. One of the first paper where signed digits are used to represent real numbers was published by Edwin Wiedmer in 1980 [177]. Based on a paper by Pietro Di Gianantonio [63] from 1999, in 2006 signed digit streams to represent real numbers were revisited by Alberto Ciaffaglione and Pietro Di Gianantonio [38]. This paper already uses coninductive definitions, which were revisited by Ulrich Berger and Tie Hou [23]. The idea to use coinductively defined predicates together with the soundness theorem in this context is due to Ulrich Berger and Monika Seisenberger [25]. The implementation of signed digit streams in Minlog was additionally considered by Kenji Miyamoto, Helmut Schwichtenberg, Nils Köpp and Till Überrück Fries [102, 123, 172]. The notions we use as the definition of ${}^{co}\mathbf{I}$ are mainly taken from [123].

5.1.2 Notations

Motivation 5.1.2. As we want to do program extraction from proofs, we are using the theory of computable functionals as metatheory, which was introduced in Section 2.1. Each variable in **TCF** comes with a type. The types we use are given in Example 2.1.5 plus the type of real numbers $\mathbb{R}$. As we do not want to declare the type of each variable in this chapter one by one, we use the following convention:

Notation 5.1.3. In this chapter we define the following assignment of variables to

types.

$m, n : \mathbb{N}$	$a, b : \mathbb{Q}$	$M, N : \mathbb{Z}^+ \to \mathbb{N}$
$d, e, k : \mathbb{Z}$	$x, y : \mathbb{R}$	$f, g : \mathbb{N} \to \mathbb{R}$
$p, q : \mathbb{Z}^+$	$v, u : \mathbb{S}$	$F : \mathbb{N} \to \mathbb{S}$

These variables can be equipped with indices or other attachments. For example, n_0 and n' have also type $\mathbb{N}$.

There are canonical inclusion maps $\mathbb{Z}^+ \hookrightarrow \mathbb{N} \hookrightarrow \mathbb{Z} \hookrightarrow \mathbb{Q} \hookrightarrow \mathbb{R}$. Using these inclusion maps implicitly we will do calculations between objects of all these types. In particular, an expression like $\frac{d+x}{2}$ is a well-defined term of type $\mathbb{R}$.

Remark 5.1.4. There are several ways to define constructive real numbers. One of the best-known methods is to define them as Cauchy sequences of rational numbers with a Cauchy modulus. In the appendix we give a construction of real numbers. In our case, rather than we are not interested in a specific definition of real numbers, we are interested in their signed digit representation (SD code). Hence, we consider real numbers as abstract objects.

In the following we often use the real equality $=$ between two objects of type $\mathbb{R}$. The real equality is an equivalence relation and compatible with the arithmetic functions on $\mathbb{R}$ and the predicate $\leq$. As usual, in this chapter the real equality and the predicate $\leq$ are non-computational.

5.1.3 Binary code vs. signed digit code

Motivation 5.1.5. The SD code of reals is similar to the binary code of reals but in addition to the digits 0 and 1, the SD code has the digit -1, which we also denote by $\overline{1}$. Since every real x can be represented as $x = k + x'$, where k is an integer and $-1 \leq x' \leq 1$ (this follows from Lemma B.17 in the appendix), we work on the interval $[-1, 1]$. A binary representation of a real number x in $[-1, 1]$ is given by a sequence of the form $sd_1d_2d_3\ldots$ such that

$$x = s\sum_{i=1}^{\infty} d_i 2^{-i},$$

where $s \in \{-1, 1\}$ and $d_i \in \{0, 1\}$ for all i. Reading the digits of the binary representation of a concrete real number one after the other, the interval where the real number is located, is halved in each step. Thus, from the binary code one can determine the real number arbitrarily exactly.

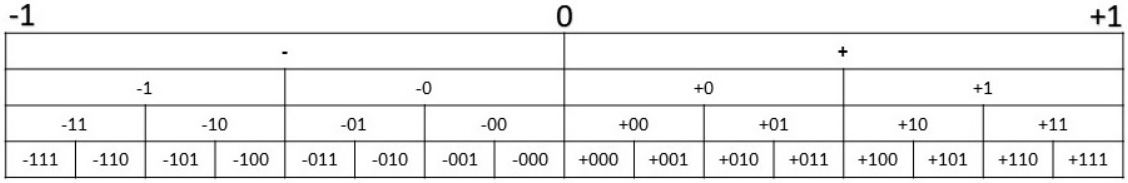


Figure 5.1: Visualization of the binary code

However, it is not always possible to compute the binary representation of a given real number or even to compute the binary representation of $\frac{x+y}{2}$ out of the binary representations of x and y. Here "compute" means to get an algorithm which takes the binary streams of x and y and gives back the digits of the binary stream $\frac{x+y}{2}$ one by one in a finite amount of steps for each digit. In particular, the algorithm can only use finitely many binary digits of x and y to generate finitely many digits of $\frac{x+y}{2}$. This is not possible due to the "gaps" in the binary representation. They are illustrated in Figure 5.1 at 0, $\pm\frac{1}{2}$, $\pm\frac{1}{4}$, $\pm\frac{3}{4}$ and so on. From the first digit (i.e. $+$ or $-$) of a real x one can decide $0 \leq x$ or $x \leq 0$, which in general cannot be done for constructive reals (similar to Remark B.9).

The signed digit code fills these gaps. For a real number $x \in [-1, 1]$ it is given by a sequence $d_1 d_2 d_3 \ldots$ such that

$$x = \sum_{i=0}^{\infty} d_i 2^{-i},$$

where $d_i \in \{\overline{1}, 0, 1\}$ for every i.

As the illustration in Figure 5.2 shows, to compute the first signed digit of a real number $x \in [-1, 1]$ one has to decide which of the cases $x \leq 0$, $-\frac{1}{2} \leq 0 \leq \frac{1}{2}$ or $0 \leq x$ holds. That this is possible, follows by the *comparability theorem* (Lemma B.17 in the appendix), which says that for reals x, y and z with $x < y$ one has $z \leq y \vee x \leq y$.

Figure 5.2 also demonstrates that the SD code of a real number except -1 and 1 is not unique, whereas the binary code is "almost" everywhere unique. This means, only the binary codes $t011\ldots$ and $t100\ldots$ represent the same real number, where t can be any initial list of binary numbers.

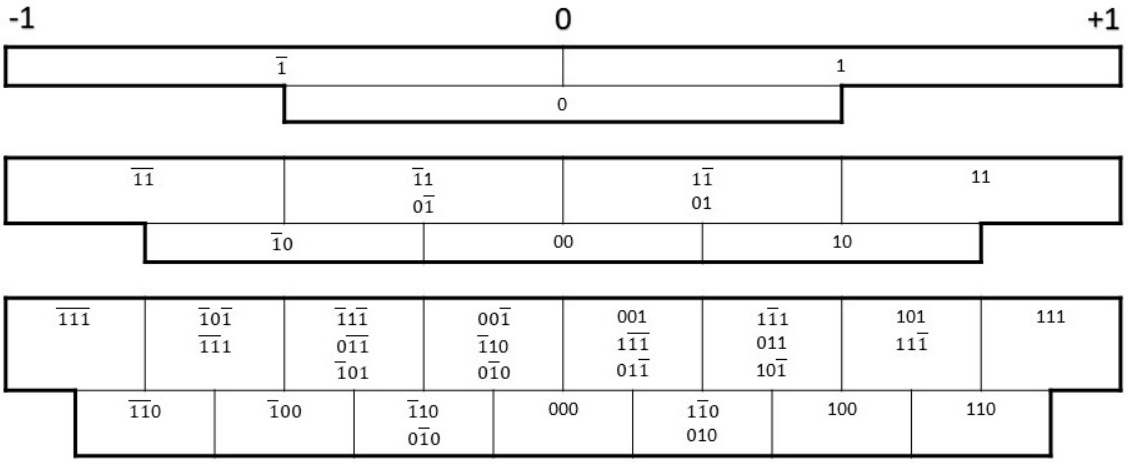


Figure 5.2: Visualization of the signed digit code

5.2 Formalisation

Motivation 5.2.1. In Chapter 4 our starting point was Zariski's lemma. From a constructive proof we managed to develop an algorithmic version of Zariski's lemma. In that case it was not a priori clear how the developed algorithm operates and even on which objects the algorithm operates. Of course, how it operates is a priori also not clear in this case, however it is clear on which objects this algorithm shall operate. Out of a sequence of signed digit codes and some modulus, the algorithm shall generate a new signed digit code. Furthermore, the starting theorem from which we do program extraction is not given. Therefore, our aim is first to formulate a suitable theorem. Since program extraction is an automatized process, this is indeed the crucial point of this chapter. First of all, we need a predicate on real numbers whose computational content is a signed digit code of the corresponding real number. A signed digit code is an infinite list of signed digits, and therefore we have to use coinductively defined predicates, which we have introduced as are part of **TCF** in Section 2.1. In particular, the whole theory in this chapter is based on **TCF**.

5.2.1 The predicate ${}^{co}\mathbf{I}$

Motivation 5.2.2. To formalise the signed digit code, we use *coinductively* defined predicates, which we have already given in Definition 2.1.16. Briefly speaking, whereas an inductively defined predicate can be seen as the least fixpoint of its clauses, a coinductively defined predicate can be seen as the greatest fixpoint of the clauses. We see this in the next definition.

Definition 5.2.3. We define the predicate $\mathbf{I}$ by the single clause

$$\mathbf{I}^+ : \forall_{d,x,x'} \left(\mathbf{Sd}\ d \to \mathbf{I}x' \to |x| \le 1 \to x = \frac{d + x'}{2} \to \mathbf{I}x \right).$$

Here, the predicate $\mathbf{Sd}$ is defined on the integers by the clauses $\mathbf{Sd}\ -1$, $\mathbf{Sd}\ 0$ and $\mathbf{Sd}\ 1$.

The predicate ${}^{co}\mathbf{I}$ is the coinductively defined predicate corresponding to $\mathbf{I}$.

Remark 5.2.4. In the axiom $\mathbf{I}^+$ in Minlog instead of $|x| \le 1$ there is $|x'| \le 1$ required. This is equivalent and does not affect the computational content.

The algebra corresponding to $\mathbf{Sd}$ is $\mathbb{D}$ from Example 2.1.5, and the real equality and inequality are non-computational. Therefore, the type of ${}^{co}\mathbf{I}x$ is

$$\mathbb{S} := \tau({}^{co}\mathbf{I}) = \mu_{\tau(X)}(\mathbb{D} \to \tau(X) \to \tau(X)) = \mu_\xi(\mathbb{D} \to \xi \to \xi).$$

In Minlog this algebra is denoted by `ai`.

Another possibility to define a predicate which says that a real number x has a signed digit code could be

$$\exists_{t \in \mathbf{T}_{\mathbb{N}\to\mathbb{D}}} \forall_n \left| \sum_{i=1}^{n} \frac{t(i)}{2^i} - x \right| \le 2^i,$$

where $\mathbf{T}_{\mathbb{N}\to\mathbb{D}}$ is the totality predicate on $\mathbb{N} \to \mathbb{D}$, which is considered in Example 2.1.15 and add a computational meaning to the predicate. The type of this predicate is $\mathbb{N} \to \mathbb{D}$, i.e. a sequence of signed digit. Presumably, the theory in this chapter would also work with this predicate. However, the approach we use has the advantage that $\mathbb{S}$ is an algebra, and therefore a base type, whereas $\mathbb{N} \to \mathbb{D}$ is a function type.

Notation 5.2.5. We denote the only constructor of $\mathbb{S}$ by C. In another notation $\mathbb{S}$ can be defined by

$$\mathbb{S} ::= \mathrm{C}\ \mathbb{D}\ \mathbb{S}.$$

In the following we identify **Sd** with $\mathbb{D}$, i.e. we write $\overline{1}$ for SdL, 0 for SdM, and 1 for SdR.

In this notation one easily sees that an element Cdv can be interpreted as an SD code with the first digit d and the tail v. We sometimes abbreviate Cdv by dv and use this notation also for reals, i.e. if we write something as dx for a real number x and a signed digit d we mean $\frac{d+x}{2}$.

Remark 5.2.6. The predicate ${}^{co}\mathbf{I}$ is given by its two axioms.

The axiom ${}^{co}\mathbf{I}^-$ is called elimination rule. Expressed in elementary formulas it is

$$ {}^{co}\mathbf{I}^- : \forall_x.{}^{co}\mathbf{I}x \to \exists_{d,x'}\left(\mathbf{Sd}\ d \wedge {}^{co}\mathbf{I}x' \wedge |x| \leq 1 \wedge x = \frac{d+x'}{2}\right). $$

The type of this axiom is $\tau({}^{co}\mathbf{I}^-) = \mathbb{S} \to \mathbb{D} \times \mathbb{S}$ and a realiser is given by the destructor $\mathcal{D} := \mathcal{D}_{\mathbb{S}}$ with its computation rule

$$ \begin{aligned} &\mathcal{D} : \mathbb{S} \to \mathbb{D} \times \mathbb{S} \\ &\mathcal{D}(\mathrm{C}dv) := \langle d, v\rangle. \end{aligned} $$

The destructor takes a stream and returns a pair consisting of its first digit and its tail. By using the projectors π_0 and π_1 one can obtain the first digit and the tail, respectively.

The axiom ${}^{co}\mathbf{I}^+$ is called the introduction axiom of ${}^{co}\mathbf{I}$ and says that ${}^{co}\mathbf{I}$ is the greatest fixpoint of its clause in a strong sense. We use the following version:

$$ \begin{aligned} {}^{co}\mathbf{I}^+ :\ \forall_x.Xx \to\ & \\ & \forall_x\left(Xx \to \exists_{d,x'}\left(\mathbf{Sd}\ d \wedge ({}^{co}\mathbf{I} \cup X)\, x' \wedge |x| \leq 1 \wedge x = \frac{d+x'}{2}\right)\right) \\ & \to {}^{co}\mathbf{I}x \end{aligned} $$

The type of this axiom depends on the type of the predicate variable X:

$$ \tau({}^{co}\mathbf{I}^+) = \tau(X) \to (\tau(X) \to \mathbb{D} \times (\mathbb{S} + \tau(X))) \to \mathbb{S} $$

A realiser of ${}^{co}\mathbf{I}^+$ is the corecursion operator ${}^{co}\mathcal{R} := {}^{co}\mathcal{R}_{\mathbb{S}}^{\tau(X)}$. As in Example 2.1.11 it is given by the computation rule

$$ \begin{aligned} {}^{co}\mathcal{R} &: \tau(X) \to (\tau(X) \to \mathbb{D} \times (\mathbb{S} + \tau(X))) \to \mathbb{S}, \\ {}^{co}\mathcal{R}tf &:= \mathrm{C}(\pi_0(ft))[\mathrm{id}, \lambda_{t'}{}^{co}\mathcal{R}t'f]\pi_1(ft), \end{aligned} $$

where $[F_0, F_1]\mathrm{in}_i T := F_i T$ for $i \in \{0, 1\}$. Here in_0 and in_1 are the two constructors of the type sum $\mathbb{S} + \tau(X)$. If $\pi_1(ft)$ has the form $\mathrm{in}_0 v$, the corecursion stops and we have $\mathrm{C}(\pi_0(ft))v$ as signed digit representation. If it has the form $\mathrm{in}_1 t'$, the corecursion goes further with the new argument t'. In both cases we have obtained at least the first digit of the stream. Iteration of the corecursion, if necessary, generates each digit of the stream one by one.

5.2.2 Basic lemmas

Motivation 5.2.7. We start by formulating two basic lemmas about the predicate ${}^{co}\mathbf{I}$. The first lemma says that ${}^{co}\mathbf{I}$ is compatible with the real equality. On a computational level this states: if $x = y$ then an SD code of x can be converted into an SD code of y. We will observe that the signed digit representation of y is the same as the signed digit representation of x.

The second lemma states: if d is a signed digit and x has a signed digit representation then so has $\frac{d+x}{2}$. Here we will see that the signed digit representation of $\frac{d+x}{2}$ is just the signed digit representation x with d put at the beginning.

Lemma 5.2.8 (`CoICompat`). The predicate ${}^{co}\mathbf{I}$ is compatible with the real equality, i.e.

$$\forall_{x,y}.\, {}^{co}\mathbf{I}x \to x = y \to {}^{co}\mathbf{I}y.$$

Proof. We apply ${}^{co}\mathbf{I}^+$ to the predicate $Px := \exists_y({}^{co}\mathbf{I}y \land x = y)$:

$$\begin{aligned}&\forall_x.\exists_y({}^{co}\mathbf{I}y \land x = y) \to \\ &\quad \forall_x \left(Px \to \exists_{d,x'} \left(\mathbf{Sd}\ d \land ({}^{co}\mathbf{I} \cup P)\, x' \land |x| \le 1 \land x = \frac{d + x'}{2} \right) \right) \\ &\quad \to {}^{co}\mathbf{I}x\end{aligned}$$

It is sufficient to prove the second premise. Therefore, we show

$$\forall_x \left(\exists_y({}^{co}\mathbf{I}y \land x = y) \to \exists_{d,x'} \left(\mathbf{Sd}\ d \land ({}^{co}\mathbf{I} \cup P)\, x' \land |x| \le 1 \land x = \frac{d + x'}{2} \right) \right).$$

Let x, y with ${}^{co}\mathbf{I}y$ and $x = y$ be given. Using ${}^{co}\mathbf{I}y$ with ${}^{co}\mathbf{I}^-$ we get $e \in \mathbf{Sd}$ and $y' \in {}^{co}\mathbf{I}$ with $|y| \le 1$ and $y = \frac{e+y'}{2}$. Hence, $d := e$ and $x' := y'$ have the desired property. □

Remark 5.2.9. In the following proofs this lemma is used tacitly. In Minlog it has the name `CoICompat`. The extracted term is given by

```
[u0]
(CoRec sd yprod ai=>ai)DesYprod u0
([su1]clft su1 pair(InL ai (sd yprod ai))crht su1)
```

Note that in Minlog $\mathbb{S}$ is denoted by `ai`. After using the computation rule of the corecursion operator once, we get

```
[u0]C clft DesYprod u0 crht DesYprod u0
```

which is $\lambda_u \mathrm{C}(\pi_0(\mathcal{D}u))(\pi_1(\mathcal{D}u))$ in standard notation. For streams of the form $\mathrm{C}dv$ this term is the identity function. Since we deal with such streams only, we drop this term hereafter.

Lemma 5.2.10 (`CoIClosureInv`)**.**

$$\forall_{x,d}.\mathbf{Sd}\ d \to {}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}\frac{d+x}{2}$$

Proof. We apply ${}^{co}\mathbf{I}^+$ to the predicate

$$Px := \exists_{d,x'}\left(\mathbf{Sd}\ d \wedge {}^{co}\mathbf{I}x' \wedge x = \frac{d+x'}{2}\right).$$

This leads to the formula

$$\begin{aligned}&\forall_x.\exists_{d,x'}\left(\mathbf{Sd}\ d \wedge {}^{co}\mathbf{I}x' \wedge x = \frac{d+x'}{2}\right) \to \\ &\quad \forall_x\left(Px \to \exists_{d,x'}\left(\mathbf{Sd}\ d \wedge ({}^{co}\mathbf{I} \cup P)\, x' \wedge |x| \le 1 \wedge x = \frac{d+x'}{2}\right)\right) \\ &\quad \to {}^{co}\mathbf{I}x.\end{aligned}$$

To prove the goal formula, it suffices to prove the second premise. Therefore, the new goal formula is

$$\begin{aligned}&\forall_x.\exists_{d,x'}\left(\mathbf{Sd}\ d \wedge {}^{co}\mathbf{I}x' \wedge x = \frac{d+x'}{2}\right) \to \\ &\quad \exists_{d,x'}\left(\mathbf{Sd}\ d \wedge ({}^{co}\mathbf{I} \cup P)\, x' \wedge |x| \le 1 \wedge x = \frac{d+x'}{2}\right).\end{aligned}$$

This is almost a tautology. The only part one has to consider is $|x| \le 1$. From $\mathbf{Sd}\ d$ and ${}^{co}\mathbf{I}x'$ we get $|d|, |x'| \le 1$, and therefore $|x| = \left|\frac{d+x'}{2}\right| \le 1$. □

Remark 5.2.11. In Minlog this lemma has the name `CoIClosureInv` and its extracted term is given by.

```
[s0,u1]
(CoRec sd yprod ai=>ai)(s0 pair u1)
([su2]clft su2 pair(InL ai (sd yprod ai))crht su2)
```

After using the computation rule of ${}^{co}\mathcal{R}$ once, we get

```
[s0,u1]C clft(s0 pair u1)crht(s0 pair u1)
```

This is the Minlog notation for the term $\lambda_d\lambda_u.Cdu$, which can be identified with C itself. Therefore, the constructor C is in fact the computational content of this lemma.

5.3 Convergence theorem

Motivation 5.3.1. The aim of this chapter is to prove the convergence theorem for the SD code. It states that the limit of each convergent sequence in ${}^{co}\mathbf{I}$ is also in ${}^{co}\mathbf{I}$. In Minlog this theorem has the additional assumption that the convergent sequence is a Cauchy sequence. Since each convergent sequence is a Cauchy sequence, this assumption is obsolete if one only considers the truth content of the statement. However, there is a small difference in the computational content. With the additional assumption and the proof in Minlog, the Cauchy modulus appears in the extracted term. Therefore, we will also define the notion of a Cauchy sequence of real and discuss the difference. As extracted term we expect a function which takes a sequence of signed digit streams (i.e. a term of type $\mathbb{N} \to \mathbb{S}$) and its modulus of convergence, and returns a signed digit stream.

We conduct the proof step by step and therefore we first prove a few lemmas. The next two lemmas were already considered in [160, 178].

Lemma 5.3.2 (`CoINegToCoIPlusOne`, `CoIPosToCoIMinusOne`)**.**

$$\forall_x.{}^{co}\mathbf{I}x \to x \leq 0 \to {}^{co}\mathbf{I}(x+1)$$
$$\forall_x.{}^{co}\mathbf{I}x \to 0 \leq x \to {}^{co}\mathbf{I}(x-1)$$

Proof. Since both formulas are shown analogously, we only prove the first one. The

introduction axiom of ${}^{co}\mathbf{I}$ is given by

$$\begin{aligned}&\forall_x.Px \to\\ &\quad \forall_x \left(Px \to \exists_{d,x'} \left(\mathbf{Sd}\ d \wedge ({}^{co}\mathbf{I} \cup P)\, x' \wedge |x| \le 1 \wedge x = \frac{d+x'}{2} \right) \right)\\ &\quad \to {}^{co}\mathbf{I}x.\end{aligned}$$

For Px we insert $\exists_y ({}^{co}\mathbf{I}y \wedge y \le 0 \wedge y+1 = x)$. Hence, it is sufficient to prove the second premise. Let x, y, ${}^{co}\mathbf{I}y$, $y \le 0$ and $y+1 = x$ be given. Our goal is

$$\exists_{d,x'} \left(\mathbf{Sd}\ d \wedge ({}^{co}\mathbf{I} \cup P)\, x' \wedge |x| \le 1 \wedge x = \frac{d+x'}{2} \right).$$

From ${}^{co}\mathbf{I}y$ we get e and y' with $\mathbf{Sd}\ e$, ${}^{co}\mathbf{I}y'$, $|y| \le 1$ and $y = \frac{e+y'}{2}$. Independently from the choice of d and x' we always get $|x| \le 1$ out of $|y| \le 1$, $y \le 0$ and $x = y+1$. In order to prove the remaining part of the formula, we use case distinction on $\mathbf{Sd}\ e$:

If $e = -1$, we define $d := 1$ and $x' := y'$. Here $\mathbf{Sd}\ d$ and ${}^{co}\mathbf{I}x'$ follow directly and we also have

$$x = y+1 = \frac{-1+y'}{2} + 1 = \frac{1+y'}{2} = \frac{d+x'}{2}.$$

If $e = 0$, we define $d := 1$ and $x' := y'+1$. Hence, $\mathbf{Sd}\ d$ is clear and in this case we prove Px'. Hence we show ${}^{co}\mathbf{I}y'$, $y' \le 0$ and $x' = y'+1$. ${}^{co}\mathbf{I}y'$ and $x' = y'+1$ are already given and $y' \le 0$ follows directly from $y \le 0$ and $y = \frac{0+y'}{2}$.

The last case is $e = 1$. Because of $y \le 0$, $y = \frac{-1+y'}{2}$ and $|y'| \le 1$, this is only possible if y is equal to 0, and therefore $x = 1$. Hence we define $d := 1$ and $x' := 1$. Then $\mathbf{Sd}\ d$ and $x = \frac{d+x'}{2}$ are obviously true and ${}^{co}\mathbf{I}\ 1$ is true because 1 has the SD representation $111\ldots$. (Formally, ${}^{co}\mathbf{I}\ 1$ must be proven by coinduction. For details we refer to the Minlog implementation of the theorem `CoIOne` in `examples/analysis/sddiv.scm`.) □

Remark 5.3.3. A realiser of the first formula in this lemma is a function `f` which takes a signed digit stream of a real number x and returns a signed digit stream of $x+1$ if $x \le 0$.
In Minlog the normalized extracted term is displayed by

```
[u]
 (CoRec ai=>ai)u
 ([u0]
   [case (cCoIClosure u0)
     (s pair u1 ->
```

```
[case s
  (SdR -> SdR pair InL cCoIOne)
  (SdM -> SdR pair InR u1)
  (SdL -> SdR pair InL u1)])])
```

In standard notation we have

$$\mathtt{f} := \lambda_u.^{co}\mathcal{R}u\left(\lambda_v.[\langle 1, \mathrm{in}_0(\pi_1(\mathcal{D}v))\rangle, \langle 1, \mathrm{in}_1(\pi_1(\mathcal{D}v))\rangle, \langle 1, \mathrm{in}_0\vec{1}\rangle]\pi_0(\mathcal{D}v)\right),$$

where $[F_{\overline{1}}, F_0, F_1]d := F_d$ for $d \in \mathbf{Sd}$ and $\vec{1}$ is the infinite list with each entry equal to 1.
Another way to characterise this function $\mathtt{f}$ is to give its computation rules:

$$\begin{aligned}\mathtt{f}(\mathrm{C}\overline{1}v) &:= \mathrm{C}1v\\ \mathtt{f}(\mathrm{C}0v) &:= \mathrm{C}1(\mathtt{f}v)\\ \mathtt{f}(\mathrm{C}1v) &:= [1, 1, \dots]\end{aligned}$$

Analogously, the extracted term of the second statement of this lemma is a function $\mathtt{g} : \mathtt{Str} \to \mathtt{Str}$ which is characterised by the rules

$$\begin{aligned}\mathtt{g}(\mathrm{C}\overline{1}v) &:= [\overline{1}, \overline{1}, \dots]\\ \mathtt{g}(\mathrm{C}0v) &:= \mathrm{C}\overline{1}(\mathtt{g}v)\\ \mathtt{g}(\mathrm{C}1v) &:= \mathrm{C}\overline{1}v.\end{aligned}$$

It takes a signed digit stream of a real x and returns a signed digit stream of $x - 1$ if $0 \leq x$.

Using this lemma we are now able to prove the following lemma:

Lemma 5.3.4 (`CoIToCoIDouble`)**.**

$$\forall_x.^{co}\mathbf{I}x \to |x| \leq \frac{1}{2} \to {}^{co}\mathbf{I}(2x)$$

Proof. From ${}^{co}\mathbf{I}x$ we get d, x' with $\mathbf{Sd}\ d$, ${}^{co}\mathbf{I}x'$, $|x| \leq 1$ and $x = \frac{d+x'}{2}$. We distinguish cases on $\mathbf{Sd}\ d$:

$d = 1$: Here $2x = 1 + x'$ and with $|x| \leq \frac{1}{2}$ it follows $x' \leq 0$. By the first statement of Lemma 5.3.2 we have ${}^{co}\mathbf{I}(2x)$.

$d = -1$: The proof in this case is done analogously but with the second statement of Lemma 5.3.2.

$d = 0$: In this case $2x = x'$ and ${}^{co}\mathbf{I}x'$ is already given. □

Remark 5.3.5. If we keep the definition of the functions `f` and `g` from Remark 5.3.3, the computational content $\mathtt{D} : \mathtt{Str} \to \mathtt{Str}$ of this lemma is given by

$$\mathtt{D} = \lambda_u.[\mathtt{g}(\pi_1(\mathcal{D}u)), \pi_1(\mathcal{D}u), \mathtt{f}(\pi_1(\mathcal{D}u))]\pi_0(\mathcal{D}u).$$

Again we give a more readable characterisation of `D` by the computation rules

$$\begin{aligned}\mathtt{D}(\mathrm{C}\overline{1}u) &:= \mathtt{g}u\\ \mathtt{D}(\mathrm{C}0u) &:= u\\ \mathtt{D}(\mathrm{C}1u) &:= \mathtt{f}u.\end{aligned}$$

At this point we want to mention a general observation. Even if the lemmas above only works for real numbers with a certain property, the extracted term is defined for all SD codes. This is similar to the defined algorithms in Chapter 4. There, we had to define the algorithms without assuming any axiom. It is the same case here as the algorithm cannot check whether an assumption is fulfilled or not.

Motivation 5.3.6. The following lemma is not implemented in Minlog because it is a special case of the theorem $\forall_{x,y}.{}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}y \to {}^{co}\mathbf{I}\left(\frac{x+y}{2}\right)$ which is implemented as the theorem `CoIAverage` in `examples/analysis/sdavaux.scm` of Minlog and was considered in [28, 123]. However, here we give a direct proof since it is instructive and elementary.

Lemma 5.3.7 (Spezial case of `CoIAverage`).

$$\forall_x.{}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}\left(\frac{x}{2} \pm \frac{1}{4}\right)$$

Proof. ${}^{co}\mathbf{I}x$ gives $x' \in {}^{co}\mathbf{I}$ and $d \in \mathbf{Sd}$ with $x = \frac{d+x'}{2}$. We show only ${}^{co}\mathbf{I}\left(\frac{x}{2} + \frac{1}{4}\right)$ because ${}^{co}\mathbf{I}\left(\frac{x}{2} - \frac{1}{4}\right)$ is proven analogously. Case distinction on $\mathbf{Sd}$ d leads to the following three cases:
$d = 1$ gives $\frac{x}{2} + \frac{1}{4} = \frac{2+x'}{4} = \frac{1+\frac{x'}{2}}{2}$,
$d = 0$ gives $\frac{x}{2} + \frac{1}{4} = \frac{1+x'}{4} = \frac{\frac{1+x'}{2}}{2}$ and
$d = -1$ gives $\frac{x}{2} + \frac{1}{4} = \frac{x'}{4} = \frac{\frac{x'}{2}}{2}$.
In each case we get ${}^{co}\mathbf{I}\left(\frac{x}{2} + \frac{1}{4}\right)$ by using Lemma 5.2.10 twice. □

Remark 5.3.8. We denote the extracted term of the proven statement by q^+. From the proof and the fact that the extracted term of Lemma 5.2.10 is given by C,

one easily sees that q^+ has the following computation rules:

$$\begin{aligned} q^+(\overline{1}u) &:= 00u \\ q^+(0u) &:= 01u \\ q^+(1u) &:= 10u \end{aligned}$$

Analogously, the extracted term q^- of the statement $\forall_x.{}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}\left(\frac{x}{2} - \frac{1}{4}\right)$ is characterised by

$$\begin{aligned} q^-(\overline{1}u) &:= \overline{1}0u \\ q^-(0u) &:= 0\overline{1}u \\ q^-(1u) &:= 00u. \end{aligned}$$

Definition 5.3.9. Let a real number x, a sequence $f : \mathbb{N} \to \mathbb{R}$ of real numbers and a modulus $M : \mathbb{Z}^+ \to \mathbb{N}$ be given. f *convergences* to x with modulus M if

$$\forall_{p \in \mathbf{T}_{\mathbb{Z}^+}} \forall_{n \geq M(p)} |f(n) - x| \leq 2^{-p},$$

where $\mathbf{T}_{\mathbb{Z}^+}$ is the totality predicate on positive integer from Example 2.1.15. In Motivation 5.3.10 we see why this is necessary. We also assume that the variable n in the formula above is total. This is hidden in the predicate $\leq$.

Furthermore, f is a *Cauchy sequence* with modulus M if

$$\forall_{p \in \mathbf{T}_{\mathbb{Z}^+}} \forall_{n,m \geq M(p)} |f(m) - f(n)| \leq 2^{-p}.$$

Motivation 5.3.10. In the following we give a version of the convergence theorem and a proof of it. A possible formulation could be

$$\forall_x \forall_f \forall_M . \forall_n {}^{co}\mathbf{I}f(n) \wedge \forall_p \forall_{n \geq M(p)} |f(n) - x| \leq 2^{-p} \to {}^{co}\mathbf{I}x.$$

The type of this formula is $\mathbb{S} \to \mathbb{S}$ as the computational moments are only in the formulas ${}^{co}\mathbf{I}f(n)$ and ${}^{co}\mathbf{I}x$. However, as extracted term we expect a function which takes a sequence of signed digit streams and a modulus of convergences and returns a new signed digit stream. Therefore, the premises of the formula appear unsatisfactory and erroneous. To see this, we consider the formula $\forall_n {}^{co}\mathbf{I}f(n)$. To prove this formula, we have to take an arbitrary n of type $\mathbb{N}$ and prove ${}^{co}\mathbf{I}f(n)$. The fact that we do not know anything about n is problematic. Note that in **TCF** we do not require that a natural number n has the form $n = 0$ or $n = Sn'$. In particular, we can not use induction or case distinction. Hence, a proof of ${}^{co}\mathbf{I}f(n)$ is only possible if the signed digit code of $f(n)$ is independent of n and therefore constant.

To solve this problem, we have to weaken the premises. As we have seen, more information about the variables is needed. Therefore, we use the totality predicate. We have introduced the totality predicate in Example 2.1.15. The totality predicate allows us to prove a formula like $\forall_{n\in\mathbf{T}_{\mathbb{N}}}{}^{co}\mathbf{I}f(n)$ by induction on n. In the next section we use this to the Heron sequence.

A similar challenge appears for the premise $\forall_p\forall_{n\geq M(p)}|f(n)-x|\leq 2^{-p}$ and it is even not clear how e.g. 2^{-p} is defined if p is not total. Therefore, we replace this premise by the formula $\forall_{p\in\mathbf{T}_{\mathbb{Z}^+}}\forall_{n\geq M(p)}|f(n)-x|\leq 2^{-p}$. The totality of the variable n shall be hidden in $n\geq M(p)$. In particular, we require that $\leq$ is only defined for total objects.

To prove the theorem with the new premises, we have to assume that the given modulus M is also total. We see in the proof why this assumption is necessary. By assuming that M is total, we obtain that the computational content depends on the modulus of convergence.

In the proof of the convergence theorem we just sketch the totality arguments as the details are simple but blow up the proof. In particular, one just have to write out the definitions. For an implementation of totality in Minlog we refer to Section 5 of [179].

Notation 5.3.11. In the following we write $\mathbf{T}$ for the totality predicate of $\mathbb{Z}^+\to\mathbb{N}$, $\mathbb{N}$ and $\mathbb{Z}^+$ as it is apparent from the context which totality predicate is meant.

Theorem 5.3.12 (`SdLim`). Let $f:\mathbb{N}\to\mathbb{R}$ be a sequence of reals in ${}^{co}\mathbf{I}$ which converges to a real x with a modulus M, then also x is in ${}^{co}\mathbf{I}$.
Expressed in a formula:

$$\forall_x\forall_f\forall_{M\in\mathbf{T}}.\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)\wedge\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x|\leq 2^{-p}\to{}^{co}\mathbf{I}x$$

Proof. We show the equivalent formula

$$\forall_x.\exists_f\exists_{M\in\mathbf{T}}\left(\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)\wedge\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x|\leq 2^{-p}\right)\to{}^{co}\mathbf{I}x$$

by applying the axiom ${}^{co}\mathbf{I}^+$ to the predicate

$$Px:=\exists_f\exists_{M\in\mathbf{T}}\left(\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)\wedge\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x|\leq 2^{-p}\right).$$

Again we need to show the second premise:

$$\forall_x.\exists_f\exists_{M\in\mathbf{T}}\left(\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)\wedge\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x|\leq 2^{-p}\right)\to$$
$$\exists_{d,x'}\left(\mathbf{Sd}\;d\wedge({}^{co}\mathbf{I}\cup P)\,x'\wedge|x|\leq 1\wedge x=\frac{d+x'}{2}\right)$$

Let x, f, $M \in \mathbf{T}$, $\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)$ and $\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x| \leq 2^{-p}$ be given. It suffices to prove

$$\exists_{d,x'}.\mathbf{Sd}\ d \wedge \exists_g \exists_{N\in\mathbf{T}} \left(\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}g(n) \wedge \forall_{p\in\mathbf{T}}\forall_{n\geq N(p)}|g(n)-x'| \leq 2^{-p}\right) \wedge |x| \leq 1 \wedge x = \frac{d+x'}{2}.$$

Regardless of the choice of d and x' we get $\forall_{n\in\mathbf{T}}|f(n)| \leq 1$ from $\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)$. This and $\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x| \leq 2^{-p}$ lead to $|x| \leq 1$. Therefore, in each case we consider $|x| \leq 1$ as proven.

As M and 4 are total, $M(4)$ is total as well. Hence, specializing $\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}f(n)$ to $M(4)$ leads to ${}^{co}\mathbf{I}f(M(4))$. Triple application of ${}^{co}\mathbf{I}^-$ gives $d_1, d_2, d_3 \in \mathbf{Sd}$ and $y' \in {}^{co}\mathbf{I}$ such that $f(M(4)) = \frac{4d_1+2d_2+d_3+y'}{8}$ or in short notation

$$f(M(4)) = d_1 d_2 d_3 y'.$$

Now we distinguish between the various representations of $f(M(4))$:

If $f(M(4))$ has one of the forms $11d_3y'$, $10d_3y'$, $1\bar{1}1y'$, $1\bar{1}0y'$ $011y'$ or $010y'$, it follows that $\frac{1}{8} \leq f(M(4))$. In this case we choose

$$d := 1 \text{ and } x' := 2x - 1,$$

then $\mathbf{Sd}\ d$ and $x = \frac{d+x'}{2}$ follow directly. Furthermore, we define

$$g(n) := 2f(M(4) \sqcup n) - 1$$

for all $n \in \mathbb{N}$, where $m \sqcup l := \max\{m, l\}$, and

$$N(p) := M(p+1)$$

for all $p \in \mathbb{Z}^+$. Since M is total, N is total, too. The formula $\forall_{p\in\mathbf{T}}\forall_{n\geq N(p)}|g(n)-x'| \leq 2^{-p}$ is a direct consequence of $\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n)-x| \leq 2^{-p}$ and it remains to show $\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}g(n)$. We calculate

$$g(n) = 2f(M(4) \sqcup n) - 1 = 4\left(\frac{f(M(4) \sqcup n)}{2} - \frac{1}{4}\right).$$

As n and $M(4)$ are total, also $M(4) \sqcup n$ is total (details in Minlog), and hence $f(M(4) \sqcup n) \in {}^{co}\mathbf{I}$, and by Lemma 5.3.7 we obtain ${}^{co}\mathbf{I}\left(\frac{f(M(4)\sqcup n)}{2} - \frac{1}{4}\right)$. Furthermore, we have $f(M(4)) \geq \frac{1}{8}$ and $\forall_{n\geq M(4)}|f(n)-x| \leq \frac{1}{16}$, and therefore

$$\begin{aligned} f(M(4) \sqcup n) &= (f(M(4) \sqcup n) - x) + (x - f(M(4))) + f(M(4)) \\ &\geq -\frac{1}{16} - \frac{1}{16} + \frac{1}{8} = 0. \end{aligned}$$

Hence, $0 \leq f(M(4) \sqcup n) \leq 1$, which implies $\left|\frac{f(M(4)\sqcup n)}{2} - \frac{1}{4}\right| \leq \frac{1}{4}$. By double application of Lemma 5.3.4 we finally get ${}^{co}\mathbf{I}g(n)$.

If $f(M(4))$ has one of the forms $\overline{1}\overline{1}d_3y'$, $\overline{1}0d_3y'$, $\overline{1}1\overline{1}y'$, $\overline{1}10y'$ $0\overline{1}\overline{1}y'$ or $0\overline{1}0y'$, it follows $f(M(4)) \leq -\frac{1}{8}$. Here, we define

$$d := -1, x' := 2x + 1, g := \lambda_n(2f(M(4) \sqcup n) + 1) \text{ and } N := \lambda_p M(p + 1).$$

The proof in this case is analogous to the proof of the first case.

It remains to consider the case that $f(M(4))$ has one of the forms $00d_3y'$, $\overline{1}11y'$, $1\overline{1}\overline{1}y'$, $01\overline{1}y'$ or $0\overline{1}1y'$. Here we have $-\frac{1}{4} \leq f(M(4)) \leq \frac{1}{4}$ and define

$$d := 0 \text{ and } x' := 2x.$$

The formulas $\mathbf{Sd}\ d$ and $x = \frac{d+x'}{2}$ are obvious. In order to prove

$$\exists_g \exists_{N\in\mathbf{T}} \left(\forall_{n\in\mathbf{T}}\, {}^{co}\mathbf{I}g(n) \wedge \forall_{p\in\mathbf{T}}\forall_{n\geq N(p)}|g(n) - x'| \leq 2^{-p}\right),$$

we define

$$g := \lambda_n 2f(M(4) \sqcup n) \text{ and } N := \lambda_p M(p + 1).$$

Again, N is total because M is. The right-hand side of the formula follows from the assumption $\forall_{p\in\mathbf{T}}\forall_{n\geq M(p)}|f(n) - x| \leq 2^{-p}$. Finally, for $n \in \mathbf{T}$ we have $M(4) \sqcup n \in \mathbf{T}$, and so $f(M(4) \sqcup n) \in {}^{co}\mathbf{I}$. Furthermore,

$$\begin{aligned} |f(M(4) \sqcup n)| &\leq |f(M(4) \sqcup n) - x| + |x - f(M(4))| + |f(M(4))| \\ &\leq \frac{1}{16} + \frac{1}{16} + \frac{1}{4} \leq \frac{1}{2}, \end{aligned}$$

which implies ${}^{co}\mathbf{I}g(n)$ by Lemma 5.3.4. □

Remark 5.3.13. We denote the extracted term by `Lim`. It has the type

$$\texttt{Lim} : (\mathbb{Z}^+ \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{S}) \to \mathbb{S},$$

and takes the modulus of convergence and the sequence of streams and returns the stream of the limit value. In order to give an at most readable characterisation of `Lim`, we define the following sets:

$$\begin{aligned} \mathbf{R} &:= \{11v, 10v, 1\overline{1}1v, 1\overline{1}0v, 011v, 010v \mid v : \mathbb{S}\} \\ \mathbf{M} &:= \{00v, \overline{1}11v, 1\overline{1}\overline{1}v, 01\overline{1}v, 0\overline{1}1v \mid v : \mathbb{S}\} \\ \mathbf{L} &:= \{\overline{1}\overline{1}v, \overline{1}0v, \overline{1}1\overline{1}v, \overline{1}10v, 0\overline{1}\overline{1}v, 0\overline{1}0v \mid v : \mathbb{S}\} \end{aligned}$$

These three sets correspond to the three cases in the proof, and therefore we have the following rule for `Lim`:

$$\texttt{Lim}\ M\ F := \begin{cases} \mathrm{C}\ 1\ (\texttt{Lim}\ \lambda_p M(p+1)\ \lambda_n(\texttt{DD}q^-F(M(4)\sqcup n))) & \text{if } F(M(4)) \in \mathbf{R} \\ \mathrm{C}\ 0\ (\texttt{Lim}\ \lambda_p M(p+1)\ \lambda_n(\texttt{D}F(M(4)\sqcup n))) & \text{if } F(M(4)) \in \mathbf{M} \\ \mathrm{C}\ \overline{1}\ (\texttt{Lim}\ \lambda_p M(p+1)\ \lambda_n(\texttt{DD}q^+F(M(4)\sqcup n))) & \text{if } F(M(4)) \in \mathbf{L} \end{cases}$$

The functions `D`, q^+ and q^- are the computational content of the lemmas above.

Note that the definition of the new sequence is not unique. For reasons of efficiency one should be flexible with the choice of the new sequence, which is called g in the proof above. For example one can replace $M(4) \sqcup n$ by $M(4) + n$. The efficiency depends on the concrete sequence. In the self-provided Haskell file we have chosen $M(4) \sqcup n$ since in the next section we define the Heron sequence and apply it to `Lim`. In this case the definition with $M(4) \sqcup n$ is most efficient. In the Minlog file we have chosen $M(3) + n$. Here we can take $M(3)$ instead of $M(4)$ as in the Minlog file M is also required to be a Cauchy sequence. That is the only difference if one assumes that M is a Cauchy sequence. Of course, one can also choose $M(3) \sqcup n$ instead of $M(3) + n$.

We give an example for an implementation as a Haskell program. First, the sets $\mathbf{R}$ and $\mathbf{L}$ are realised as Boolean functions:

```
rAux :: Str -> Bool
rAux (SdR :~: SdR :~: v) = True
rAux (SdR :~: SdM :~: v) = True
rAux (SdR :~: SdL :~: SdR :~: v) = True
rAux (SdR :~: SdL :~: SdM :~: v) = True
rAux (SdM :~: SdR :~: SdR :~: v) = True
rAux (SdM :~: SdR :~: SdM :~: v) = True
rAux v = False

lAux :: Str -> Bool
lAux (SdL :~: SdL :~: v) = True
lAux (SdL :~: SdM :~: v) = True
lAux (SdL :~: SdR :~: SdL :~: v) = True
lAux (SdL :~: SdR :~: SdM :~: v) = True
lAux (SdM :~: SdL :~: SdL :~: v) = True
lAux (SdM :~: SdL :~: SdM :~: v) = True
lAux v = False
```

Then one can define `cCoILim` by case distinction:

```
cCoILim :: (Int -> Int) -> (Int -> Str) -> Str
cCoILim m f
   | rAux (f (m 4)) = SdR :~: (cCoILim n (funcR m f))
   | lAux (f (m 4)) = SdL :~: (cCoILim n (funcL m f))
   | otherwise      = SdM :~: (cCoILim n (funcM m f))
   where n = \p -> (m (p+1))
```

In this implementation the constructor `C` is denoted by `:~:` and written as an infix. The tree elements in **Sd** are given by `SdR`, `SdM` and `SdL` and shall be interpreted by 1, 0 and $\overline{1}$, respectively. The functions `funcR`, `funcL` and `funcM` are the corresponds to the functions in the definition of `Lim` above, i.e. `funcR` corresponds to the $\lambda_n(\mathtt{DD}q^-F(M(4)\sqcup n)))$, `funcL` corresponds to $\lambda_n(\mathtt{DD}q^+F(M(4)\sqcup n)))$ and `funcM` corresponds to $\lambda_n(\mathtt{D}F(M(4)\sqcup n)))$.
The proof in Minlog of this theorem is quite straightforward and direct, however it reveals to be a lengthy procedure, as we have to consider each of the cases one by one. In the handwritten part above they are divided into three classes, which shortens the proof.

5.4 Application: Heron's method

Motivation 5.4.1. In this chapter we apply the convergent theorem to Heron's method and get an algorithm which computes the SD code of the square root of a non-negative real number. This chapter is currently not implemented in Minlog, but we present the extracted terms in Haskell notation.

We do not specify, how the square root is defined. This depends on the model of the real numbers. We just need that $\sqrt{x}$ exists for all $x \geq 0$ and has the properties $\sqrt{x} \geq 0$, $\sqrt{x}^2 = x$ and $x \leq y$ implies $\sqrt{x} \leq \sqrt{y}$.

The following statements about the Heron sequences are well-known and similar statements can be found in many textbooks.

Definition 5.4.2. We define $\mathrm{H} : \mathbb{R} \to \mathbb{N} \to \mathbb{R}$ by the computation rules

$$\begin{aligned} \mathrm{H}(x, 0) &:= 1, \\ \mathrm{H}(x, n+1) &:= \frac{1}{2}\left(\mathrm{H}(x, n) + \frac{x}{\mathrm{H}(x, n)}\right). \end{aligned}$$

For all non-negative x the sequence $\lambda_n H(x, n) =: H(x) : \mathbb{N} \to \mathbb{R}$ is the sequence, we get from Heron's method with start value 1.

Remark 5.4.3. Note that H is well-defined for non-negative x since one can easily prove $H(x,n) \geq 2^{-n}$ by induction. This is especially important for a possible implementation in Minlog as division in Minlog needs a witness of positivity.

Lemma 5.4.4. For all $x \in [0,1]$ the sequence $H(x)$ converges to $\sqrt{x}$ with modulus $\iota : \mathbb{Z}^+ \to \mathbb{N}$ which is the inclusion from $\mathbb{Z}^+$ to $\mathbb{N}$. Furthermore, we have $\forall_{n \in \mathbf{T}} \sqrt{x} \leq H(x,n)$.[1]

Proof. Let $x \in [0,1]$ be given. For each total n we define $\Delta(x,n) := H(x,n) - \sqrt{x}$. A short calculation gives

$$\begin{aligned}\Delta(x,n+1) &= \frac{1}{2}\left(\mathrm{H}(x,n) + \frac{x}{\mathrm{H}(x,n)}\right) - \sqrt{x} = \frac{(H(n,x))^2 - 2H(x,n)\sqrt{x} + x}{2H(x,n)} \\ &= \frac{(\Delta(x,n))^2}{2H(x,n)}.\end{aligned}$$

By induction on n we immediately get $0 < H(x,n)$ and therefore $0 \geq \Delta(x,n+1)$. Since in addition $\Delta(x,0) = 1 - \sqrt{x} \geq 0$, we have $\forall_{n \in \mathbf{T}} \sqrt{x} \leq H(x,n)$.
Furthermore, we calculate

$$\begin{aligned}\Delta(x,n+1) &= \frac{(\Delta(x,n))^2}{2H(x,n)} = \frac{1}{2}\Delta(x,n)\frac{\Delta(x,n)}{H(x,n)} = \frac{1}{2}\Delta(x,n)\left(1 - \frac{\sqrt{x}}{H(x,n)}\right) \\ &\leq \frac{1}{2}\Delta(x,n).\end{aligned}$$

Therefore, by induction we have $|H(x,n) - \sqrt{x}| = \Delta(x,n) \leq 2^{-n}$ and this implies $\forall_{p \in \mathbf{T}} \forall_{n \geq p} |H(x,n) - \sqrt{x}| \leq 2^{-p}$, i.e. $H(x)$ converges to $\sqrt{x}$ with modulus ι. □

Remark 5.4.5. This lemma does not have any computational content, but it states that ι is a modulus of convergence of Hx to $\sqrt{x}$ from above. In some special cases we can even improve the modulus:

Proposition 5.4.6. If $x \in [\frac{1}{4}, 1]$ then $\mathtt{poslog} : \mathbb{Z}^+ \to \mathbb{N}$ is a modulus of the converges from $H(x)$ to $\sqrt{x}$. For a positive integer p we define $\mathtt{poslog}(p)$ as the least natural number n with $p \leq 2^n$.

Proof. Let $x \in [\frac{1}{4}, 1]$ be given. From Lemma 5.4.4 we know $\forall_{n \in \mathbf{T}} \sqrt{x} \leq H(x,n)$ and therefore $\forall_{n \in \mathbf{T}} \frac{1}{2} \leq H(x,n)$. In the proof of Lemma 5.4.4 the formula

$$\Delta(x,n+1) = \frac{(\Delta(x,n))^2}{2H(x,n)}$$

[1] Similar to the discussion in Motivation 5.3.10, we quantify over total objects as the proof uses induction over n.

is proven. This implies $\Delta(x, n+1) \leq (\Delta(x, n))^2$. By using induction an $\Delta(x, 0) = 1 - \sqrt{x} \leq \frac{1}{2}$ we get

$$\Delta(x, n) \leq 2^{-2^n}$$

for each total n. Hence, for given p and $n \geq \texttt{poslog}(p)$ we get $p \leq 2^n$ and therefore

$$\left|H(x, n) - \sqrt{x}\right| = \Delta(x, n) \leq 2^{-2^n} \leq 2^{-p}$$

□

Remark 5.4.7. One possibility to implement the function `poslog` is to define an auxiliary function $\texttt{auxlog} : \mathbb{Z}^+ \to \mathbb{N} \to \mathbb{N}$ with the computation rules

$$\texttt{auxlog}\ p\ n := \begin{cases} n & \text{if } p \leq 2^n \\ \texttt{auxlog}\ p\ (n+1) & \text{otherwise,} \end{cases}$$

and then set $\texttt{poslog}(p) := \texttt{auxlog}\ p\ 0$.
We do not use this proposition and the function `poslog` in the main theorem, but for concrete calculations we use it to reduce their runtime.

Lemma 5.4.8. For all $x \in {}^{co}\mathbf{I}$ with $\frac{1}{16} \leq x$ we have $\forall_{n \in \mathbf{T}}{}^{co}\mathbf{I}(H(x, n))$. Expressed as a formula:

$$\forall_x.{}^{co}\mathbf{I}x \to \frac{1}{16} \leq x \to \forall_{n \in \mathbf{T}}{}^{co}\mathbf{I}(H(x, n))$$

Proof. We use the results of [123], [160] and of Section 3.3 from [178]. In the first script the statement

$$\forall_{x,y}.{}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}y \to {}^{co}\mathbf{I}\frac{x+y}{2} \tag{5.1}$$

is proven. In Minlog this theorem is implemented in `sdavaux.scm` in the folder `examples/analysis` and has the name `CoIAverage`. In the last two scripts, the theorem

$$\forall_{x,y}.{}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}y \to |x| \leq y \to \frac{1}{4} \leq y \to {}^{co}\mathbf{I}\frac{x}{y} \tag{5.2}$$

is proven. In Minlog this theorem is implemented in `sddiv.scm` in the folder `examples/analysis` and has the name `CoIDiv`. Using these formulas the proof of this lemma is done by induction on n:

For $n = 0$ it is easy since $H(x, 0) = 1$ and ${}^{co}\mathbf{I}\ 1$.
For any total n we have

$$H(x, n+1) = \frac{1}{2}\left(H(x, n) + \frac{x}{H(x, n)}\right).$$

By Lemma 5.4.4 we get $\sqrt{x} \le H(n, x)$, and therefore

$$\sqrt{\frac{1}{16}} = \frac{1}{4} \le H(x, n) \quad \text{and} \quad x \le \sqrt{x} \le H(x, n).$$

Additionally, by the induction hypothesis ${}^{co}\mathbf{I}(H(x, n))$. By (5.2) we have ${}^{co}\mathbf{I}_{\frac{x}{H(x,n)}}$, so with (5.1) we get ${}^{co}\mathbf{I}(H(x, n+1))$. □

Remark 5.4.9. We denote the computational content of (5.1) and (5.2) by `cCoIAverage` and `cCoIDiv`, respectively. Each of these terms takes two streams of reals and returns a stream of their average and their quotient, respectively.

As we have used induction over n to prove the lemma, the extracted term `Heron` is defined by recursion:

$$\begin{aligned} \texttt{Heron}\ v\ 0 &:= [1, 1, \dots] \\ \texttt{Heron}\ v\ (n+1) &:= \texttt{cCoIAverage}(\texttt{Heron}\ v\ n)(\texttt{cCoIDiv}\ v\ (\texttt{Heron}\ v\ n)) \end{aligned}$$

This is Definition 5.4.2 in the notation of streams.

Theorem 5.4.10.

$$\forall_x.{}^{co}\mathbf{I}x \to 0 \le x \to {}^{co}\mathbf{I}\sqrt{x}$$

Proof. Recall the introduction axiom of ${}^{co}\mathbf{I}^{+}$:

$$\begin{aligned} &\forall_x.\exists_y\left({}^{co}\mathbf{I}y \land 0 \le y \land \sqrt{y} = x\right) \to \\ &\quad \forall_x\left(Px \to \exists_{d,x'}\left(\mathbf{Sd}\ d \land ({}^{co}\mathbf{I} \cup P)\, x' \land |x| \le 1 \land x = \frac{d + x'}{2}\right)\right) \\ &\quad \to {}^{co}\mathbf{I}x \end{aligned}$$

We apply it to the predicate

$$Px := \exists_y\left({}^{co}\mathbf{I}y \land 0 \le y \land \sqrt{y} = x\right).$$

To show the goal formula, we show the second premise:

$$\forall_x.\exists_y\left({}^{co}\mathbf{I}y \wedge 0 \leq y \wedge \sqrt{y} = x\right) \to \\ \exists_{d,x'}\left(\mathbf{Sd}\ d \wedge \left({}^{co}\mathbf{I} \cup P\right) x' \wedge |x| \leq 1 \wedge x = \frac{d+x'}{2}\right)$$

Let x, y with ${}^{co}\mathbf{I}y$, $0 \leq x$ and $\sqrt{y} = x$ be given. $|x| \leq 1$ follows directly from $0 \leq y \leq 1$ and $y = \sqrt{x}$. From ${}^{co}\mathbf{I}y$ and triple application of ${}^{co}\mathbf{I}^-$ we get $d_1, d_2, d_3 \in \mathbf{Sd}$ and $y' \in {}^{co}\mathbf{I}$ with $y = d_1 d_2 d_3 y'$. We distinguish three different cases:

If y has one of the forms $\overline{1}d_2d_3y'$, $0\overline{1}d_3y'$ or $00\overline{1}y'$, it follows $y \leq 0$, and therefore $x = \sqrt{y} = 0$. Hence, we define $d := 0$ and $x' := 0$, then $\mathbf{Sd}\ d$, ${}^{co}\mathbf{I}x'$ and $x = \frac{d+x'}{2}$ are obvious.

If y has one of the forms $000y'$, $001y'$, $01\overline{1}y'$ or $1\overline{11}y'$, we can rewrite $y = 00ey'$ for an $e \in \{0, 1\}$. Here we define $d := 0$ and $x' := \sqrt{\frac{e+y'}{2}}$, then $\mathbf{Sd}\ d$ and

$$x = \sqrt{y} = \sqrt{\frac{e+y'}{8}} = \frac{\sqrt{\frac{e+y}{2}}}{2} = \frac{d+x'}{2}.$$

Furthermore, ${}^{co}\mathbf{I}\left(\frac{e+y'}{2}\right)$ by Lemma 5.2.10, and $0 \leq \frac{e+y'}{2}$ since $0 \leq y = \frac{e+y'}{8}$. Altogether we get Px'.

The remaining case is that y has one of the forms $010y'$, $011y'$, $1\overline{1}1y'$, $1\overline{1}0y'$, $10d_3y'$ or $11d_3y'$. Here it follows $\frac{1}{8} \leq y$. Hence, $\forall_{n \in \mathbf{T}}{}^{co}\mathbf{I}(H(y, n))$ by Lemma 5.4.8. Furthermore, by Lemma 5.4.4 we know that $H(y)$ converges to $\sqrt{y}$ with modulus $\iota : \mathbb{Z}^+ \to \mathbb{N}$. Thus, we use Theorem 5.3.12 to get ${}^{co}\mathbf{I}x$, and applying ${}^{co}\mathbf{I}^-$ leads to

$$\exists_{d,x'}\left(\mathbf{Sd}\ d \wedge {}^{co}\mathbf{I}x' \wedge |x| \leq 1 \wedge x = \frac{d+x'}{2}\right),$$

which proves the goal formula. □

Remark 5.4.11. By the definitions of `Lim` as the extracted term of Theorem 5.3.12 and `Heron` as the extracted term of Lemma 5.4.8 we have the following rules for the computational content `sqrt` : `Str` → `Str` of this theorem:

$$\begin{aligned} \texttt{sqrt}(\overline{1}u) &:= [0, 0, \dots] \\ \texttt{sqrt}(0\overline{1}u) &:= [0, 0, \dots] \\ \texttt{sqrt}(00u) &:= 0\ \texttt{sqrt}\ u \\ \texttt{sqrt}(01\overline{1}u) &:= 0\ \texttt{sqrt}\ 1u \\ \texttt{sqrt}(1\overline{11}u) &:= 0\ \texttt{sqrt}\ 1u \\ \texttt{sqrt}\ u &:= \texttt{Lim}\ \iota\ (\texttt{heron}\ u) \end{aligned}$$

The last rule shall only be applied if the other rules do not fit.

This algorithm has an inefficient runtime, which comes from the recursive definition of `Heron`. In each step the function `cCoIDiv` is used twice and it has already a quadratic runtime. Therefore, `Heron` has an at least exponential runtime and Haskell already takes quite a few minutes to compute even the first digit of `Heron` $[1,0,0,\dots]$ 10. By using `poslog` from Lemma 5.4.6 instead of ι we get a bit more digits of $\sqrt{\frac{1}{2}}$. If we enter

```
cCoILim poslog (heron phalf)
```

in Haskell and wait about one minute, we get

```
+1 +1 0 -1 +1 -1 +1 -1 0 0 0 0 +1 -1 +1 -1 0 0 -1 +1 +1 -1 -1 +1 +1
-1 -1 +1 +1
```

as output. These are the first 29 digits of $\sqrt{\frac{1}{2}} = \frac{\sqrt{2}}{2}$. One can check that this result is indeed valid. Since each element of the sequence $(H(\frac{1}{2},n))_n$ is a rational number, the real number $\sqrt{2}$ has the representation $\left(\left(H(\frac{1}{2},n)\right)_n, \texttt{poslog}\right)$ as a pair of a Cauchy sequence of rational numbers and a Cauchy modulus. However, by this example we can check that our algorithm provides indeed valid results. But we also want to compute the square root of irrational numbers. A possible approach to generate some irrational numbers is the general form of the Liouville number to the base 2. It is given by

$$L = \sum_{i=1}^{\infty} 2^{-f(i)},$$

where $f : \mathbb{N} \to \mathbb{N}$ is strictly monotonically increasing. If $f(n) = n!$ for all n, we have the proper Liouville number to the base 2 and this number is even transcendent as Liouville showed in [108]. Because of its representation one can easily implement its signed digit code in Haskell:

```
liouvilleAux :: (Int -> Int) -> Int -> Int -> Str
liouvilleAux f m n = if (f(m) == n)
   then (SdR :~: liouvilleAux f (m+1) (n+1))
   else  (SdM :~: liouvilleAux f m (n+1))

liouville f = liouvilleAux f 1 1
```

Again we can use `poslog` as modulus, and therefore we enter

```
cCoILim poslog (heron (liouville facul))
```

where `facul` is the factorial function. When we wait about one minute, the output is

```
+1 +1 +1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 +1 -1 0 +1 -1 0 +1
-1 0 +1 -1
```

5.5 Application: multiplication

Motivation 5.5.1. The application we show in this section is the result of a collaboration with Nils Köpp. It is motivated by Helmut Schwichtenberg and the Minlog file `exaples/analysis/sdmult.scm`. In this file it is proven that ${}^{co}\mathbf{I}x$ and ${}^{co}\mathbf{I}y$ implies ${}^{co}\mathbf{I}(xy)$. In the following we use Theorem 5.3.12 to develop another proof of this statement. Our proof uses that ${}^{co}\mathbf{I}x$ implies that x is the limit of a sequence of the form

$$\left(\sum_{i=1}^{n} \frac{d_i^{(n)}}{2^i}\right)_n,$$

where each $d_i^{(n)}$ is a signed digit. Multiplication by each number in this sequence can be divided into repeated multiplication by a signed digit and application of `CoIAverage` (which was already mentioned in proof of Lemma 5.4.8).

This section is rather informal as we formulate the extracted term by hand.

Notation 5.5.2. In this section the variable l has type $\mathbb{L}(\mathbb{Z})$ and the variable L has type $\mathbb{L}(\mathbb{D})$, where $\mathbb{L}$ is the list type from Example 2.1.5. We denote the constructor Cons also by :: as infix notation and we also use :: to denote the attachment from right, i.e. for a list $l = [d_1, \ldots, d_n]$, we have

$$d_0 :: [d_1, \ldots, d_n] = [d_0, \ldots, d_n] \qquad \text{and} \qquad [d_1, \ldots, d_n] :: d_{n+1} = [d_1, \ldots, d_{n+1}].$$

The totality $\mathbf{T}$ for objects of type $\mathbb{L}(\mathbb{Z})$ is inductively defined by the rules $[] \in \mathbf{T}$ and $\forall_{d,l}(l \in \mathbf{T} \to d :: l \in \mathbf{T})$.

For a list l, $\mathrm{lh}(l)$ denotes the length of l and for $i \leq \mathrm{lh}(l)$ we denote the i-th entry by $l(i)$. In this section we start the enumeration by 1, i.e. $l = [l(1), \ldots, l(\mathrm{lh}(l))]$ and $(d :: l)(1) = d$ for all $l \in \mathbf{T}$ and d.

Remark 5.5.3. Note that :: from right is formally defined by the rules $[] :: d := d :: []$ and $(e :: l) :: d := e :: (l :: d)$.

The totality of lists we use in this chapter is also called *structural* totality [159, Section 7.1.6]. The elimination axiom of $l \in \mathbf{T}$ can be seen as induction over the length of l. E.g. using the induction axiom one can easily prove $\forall_{d,l}(l \in \mathbf{T} \to l :: d \in \mathbf{T})$.

Lemma 5.5.4.

$$\forall_x.{}^{co}\mathbf{I}x \to \forall_{l\in\mathbf{T}}\left(\forall_{i\le \mathrm{lh}(l)}\mathbf{Sd}\ l(i) \to {}^{co}\mathbf{I}\left(x\sum_{i=1}^{\mathrm{lh}(l)}\frac{l(i)}{2^i}\right)\right)$$

Proof. In this proof we use the theorems `CoIAverage` (i.e. $\forall_{x,y}.{}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}y \to {}^{co}\mathbf{I}\frac{x+y}{2}$) and `CoISdTimes` (i.e. $\forall_{x,d}.{}^{co}\mathbf{I}x \to \mathbf{Sd}\ d \to {}^{co}\mathbf{I}(dx)$), which are implemented in `sdavaux.scm` and `sdmult.scm`, respectively.

Let x with ${}^{co}\mathbf{I}x$ be given. We use induction over $l \in \mathbf{T}$. If $l = []$ then $x\sum_{i=1}^{\mathrm{lh}(l)}\frac{d_i}{2^i} = 0$ and we are done as 0 has the infinite list of 0 as signed digit code.

For the induction step we assume

$$\forall_{i\le \mathrm{lh}(l)}\mathbf{Sd}\ l(i) \to {}^{co}\mathbf{I}\left(x\sum_{i=1}^{\mathrm{lh}(l)}\frac{l(i)}{2^i}\right)$$

and our goal is

$$\forall_{i\le \mathrm{lh}(d::l)}\mathbf{Sd}\ (d::l)(i) \to {}^{co}\mathbf{I}\left(x\sum_{i=1}^{\mathrm{lh}(d::l)}\frac{(d::l)(i)}{2^i}\right)$$

for some given d. From $\forall_{i\le \mathrm{lh}(d::l)}\mathbf{Sd}\ (d::l)(i)$ is follows $\mathbf{Sd}\ d$ and $\forall_{i\le \mathrm{lh}(l)}\mathbf{Sd}\ l(i)$, and therefore by the induction hypothesis

$${}^{co}\mathbf{I}\left(x\sum_{i=1}^{\mathrm{lh}(l)}\frac{l(i)}{2^i}\right).$$

Furthermore, we calculate

$$x\sum_{i=1}^{\mathrm{lh}(d::l)}\frac{d_i}{2^i} = \frac{1}{2}\left(xd + x\sum_{i=1}^{\mathrm{lh}(l)}\frac{l(i)}{2^i}\right)$$

and apply `CoISdTimes` to get ${}^{co}\mathbf{I}(xd)$. Hence, `CoIAverage` finishes the proof. □

Remark 5.5.5. From the proof we obtain that the computational content of this lemma is a function $\mathtt{f} : \mathbb{S} \to \mathbb{L}(\mathbb{D}) \to \mathbb{S}$ given by the following rules:

$$\begin{aligned}\mathtt{f}\ u\ [] &:= [0,0,\dots]\\ \mathtt{f}\ u\ (d::L) &:= \mathtt{cCoIAverage}(\mathtt{f}\ u\ L)(\mathtt{cCoISdTimes}\ d\ u)\end{aligned}$$

The functions $\texttt{cCoIAverage} : \mathbb{S} \to \mathbb{S} \to \mathbb{S}$ and $\texttt{cCoISdTimes} : \mathbb{D} \to \mathbb{S} \to \mathbb{S}$ are the computational content of `CoIAverage` and `CoISdTimes`, respectively. `f` takes a signed digit code of a real number x and a list l of signed digits and returns a signed digit code of $x \sum_{i=1}^{\mathrm{lh}(l)} \frac{l(i)}{2^i}$.

Lemma 5.5.6.

$$\forall_{n\in\mathbf{T}}\forall_x.{}^{co}\mathbf{I}x \to \exists_{l\in\mathbf{T},y}\left(\forall_{i\le\mathrm{lh}(l)}\mathbf{Sd}\ l(i) \land \mathrm{lh}(l) = n \land {}^{co}\mathbf{I}y \land x = \sum_{i=1}^{n}\frac{l(i)}{2^i} + \frac{y}{2^n}\right)$$

Proof. We use induction on $n \in \mathbf{T}$. If $n = 0$, we choose $y = x$ and $l = []$. For the induction step we assume

$$\forall_x.{}^{co}\mathbf{I}x \to \exists_{l\in\mathbf{T},y}\left(\forall_{i\le\mathrm{lh}(l)}\mathbf{Sd}\ l(i) \land \mathrm{lh}(l) = n \land {}^{co}\mathbf{I}y \land x = \sum_{i=1}^{n}\frac{l(i)}{2^i} + \frac{y}{2^n}\right),$$

and for given x with ${}^{co}\mathbf{I}x$ we have to show

$$\exists_{l'\in\mathbf{T},y'}\left(\forall_{i\le\mathrm{lh}(l')}\mathbf{Sd}\ l'(i) \land \mathrm{lh}(l') = n+1 \land {}^{co}\mathbf{I}y' \land x = \sum_{i=0}^{n+1}\frac{l'(i)}{2^i} + \frac{y'}{2^{n+1}}\right).$$

From ${}^{co}\mathbf{I}x$ and ${}^{co}\mathbf{I}^-$ we get d and x' with $\mathbf{Sd}\ d$, ${}^{co}\mathbf{I}x'$ and $x = \frac{x'+d}{2}$. We apply the induction hypothesis to x' and get $l \in \mathbf{T}$ and y with

$$\forall_{i\le\mathrm{lh}(l)}\mathbf{Sd}\ l(i) \land \mathrm{lh}(l) = n \land {}^{co}\mathbf{I}y \land x' = \sum_{i=1}^{n}\frac{l(i)}{2^i} + \frac{y}{2^n}.$$

To prove our goal, we define $l' := d :: l$ and $y' := y$. Then $\forall_{i\le\mathrm{lh}(l')}\mathbf{Sd}\ l'(i)$, $\mathrm{lh}(l') = n+1$ and ${}^{co}\mathbf{I}y'$ are obvious and

$$x = \frac{1}{2}(d + x') = \frac{1}{2}\left(d + \sum_{i=1}^{n}\frac{l(i)}{2^i} + \frac{y}{2^n}\right) = \sum_{i=1}^{n+1}\frac{(d :: l)(i)}{2^i} + \frac{y}{2^{n+1}}.$$

□

Remark 5.5.7. The computational content is a function $\mathsf{g} : \mathbb{N} \to \mathbb{S} \to \mathbb{L}(\mathbb{D}) \times \mathbb{S}$ given by the rules

$$\mathsf{g}\ 0\ u := \langle [], u\rangle$$
$$\mathsf{g}\ (n+1)\ (\mathrm{C}du) := \langle d :: (\pi_0(\mathsf{g}\ n\ u)), \pi_1(\mathsf{g}\ n\ u)\rangle.$$

For input n and u the function returns a list of the first n elements of u and the remainder of u.

By using the axiom of choice[2], i.e.

$$\forall_{n\in\mathbf{T}}\exists_{t^\tau}A(n,t)\to\exists_{h^{\mathbb{N}\to\tau}}\forall_{n\in\mathbf{T}}A(n,h(n)),\tag{5.3}$$

we can reformulate the lemma above to

$$\forall_x.{}^{co}\mathbf{I}x\to\exists_{l_{(\cdot)},y_{(\cdot)}}\forall_{n\in\mathbf{T}}\left(l_n\in\mathbf{T}\wedge\forall_{i\le\mathrm{lh}(l_n)}\mathbf{Sd}\ l_n(i)\wedge\mathrm{lh}(l_n)=n\wedge{}^{co}\mathbf{I}y_n\wedge x=\sum_{i=1}^{n}\frac{l_n(i)}{2^i}+\frac{y_n}{2^n}\right).\tag{5.4}$$

A realiser of Formula (5.3) is given by the identity on $\mathbb{N}\to\tau(A)$. This can easily be checked by using the definition of the reliability predicate in [157]. Therefore a realiser $\mathsf{g}':\mathbb{S}\to\mathbb{N}\to\mathbb{L}(\mathbb{D})\times\mathbb{S}$ of (5.4) can be characterised by

$$\mathsf{g}'\ u\ n=\mathsf{g}\ n\ u.$$

Probably (5.4) is also probable without the axiom of choice by using the recursion operator and the computational content of `ApproxSplit` (Lemma B.17), but we are mainly interested in the computational content and the axiom of choice makes it considerably simpler.

Lemma 5.5.8. Assuming (5.3), we have

$$\forall_{x,y}.{}^{co}\mathbf{I}x\to{}^{co}\mathbf{I}y\to\exists_{l_{(\cdot)}}\left(\forall_{n\in\mathbf{T}}{}^{co}\mathbf{I}\left(x\sum_{i=1}^{n}\frac{l_n(i)}{2^i}\right)\wedge\forall_{p\in\mathbf{T}}\forall_{n\ge p}\left|x\sum_{i=1}^{n}\frac{l_n(i)}{2^i}-xy\right|\le 2^{-p}\right).$$

Proof. Let x,y with ${}^{co}\mathbf{I}x$ and ${}^{co}\mathbf{I}y$ be given. We apply Lemma 5.5.6 and Remark 5.5.7 to y and ${}^{co}\mathbf{I}y$. Then we get sequences $l_{(\cdot)}$ and $y_{(\cdot)}$ such that

$$\forall_{n\in\mathbf{T}}\left(l_n\in\mathbf{T}\wedge\forall_{i\le\mathrm{lh}(l_n)}\mathbf{Sd}\ l_n(i)\wedge\mathrm{lh}(l_n)=n\wedge{}^{co}\mathbf{I}y_n\wedge y=\sum_{i=1}^{n}\frac{l_n(i)}{2^i}+\frac{y_n}{2^n}\right).$$

[2]Despite the name "axiom of choice", (5.3) is a constructive principle due to the strong existence quantifiers.

Hence, for given $n \in \mathbf{T}$ we have ${}^{co}\mathbf{I}\left(x \sum_{i=1}^{n} \frac{l_n(i)}{2^i}\right)$ by Lemma 5.5.4. Furthermore, for $p \in \mathbf{T}$ and $n \geq p$ we calculate

$$\left| x \sum_{i=1}^{n} \frac{l_n(i)}{2^i} - xy \right| = |x| \left| \sum_{i=1}^{n} \frac{l_n(i)}{2^i} - \left(\sum_{i=1}^{n} \frac{l_n(i)}{2^i} + \frac{y_n}{2^n} \right) \right| = |x| \frac{|y_n|}{2^n} \leq 2^{-p},$$

where we have used $|x|, |y_n| \leq 1$, which follows from ${}^{co}\mathbf{I}x$ and ${}^{co}\mathbf{I}y_n$. □

Remark 5.5.9. The second part of the lemma above states that $\left(x \sum_{i=1}^{n} \frac{l_n(i)}{2^i}\right)_n$ converges to xy with modulus ι. Therefore, the computational content of this lemma is only contain in the first part. It is a function $\mathtt{h} : \mathbb{S} \to \mathbb{S} \to \mathbb{N} \to \mathbb{S}$ which takes the signed digit streams of x and y and a natural number n and returns a signed digit stream of $x \sum_{i=1}^{n} \frac{l_n(i)}{2^i}$. It can be characterised by

$$\mathtt{h}\ u\ v\ n := \mathtt{f}\ u\ (\pi_0(\mathtt{g} n v)),$$

where $\mathtt{f}$ and $\mathtt{g}$ are defined in Remark 5.5.5 and 5.5.7, respectively.

Theorem 5.5.10. If (5.3) then

$$\forall_{x,y}.\, {}^{co}\mathbf{I}x \to {}^{co}\mathbf{I}y \to {}^{co}\mathbf{I}(xy).$$

Proof. Let x, y with ${}^{co}\mathbf{I}x$ and ${}^{co}\mathbf{I}y$ be given. Applying this to Lemma 5.5.8 we receive a sequence $f : \mathbb{N} \to \mathbb{R}$ of real numbers such that $\forall_{n \in \mathbf{T}}\, {}^{co}\mathbf{I}f(n)$ and f converges to xy with modulus ι. Therefore, ${}^{co}\mathbf{I}(xy)$ follows by Theorem 5.3.12. □

Remark 5.5.11. Using the computational content $\mathtt{h}$ from Remark 5.5.9, the extracted term of this Theorem is a function $\mathtt{mult} : \mathbb{S} \to \mathbb{S} \to \mathbb{S}$ given by

$$\mathtt{mult}\ u\ v = \mathtt{Lim}\ \iota\ (\mathtt{h}\ u\ v)$$

5.6 Outlook and future work

Remark 5.6.1. It is possible to generalise the Heron sequence for roots of higher order. For a positive integer n and $x \in [0, 1]$ we define

$$\begin{aligned} G\ n\ x\ 0 &:= 1 \\ G\ n\ x\ (k+1) &:= \frac{1}{n}\left((n-1)(G\ n\ x\ k) + \frac{x}{(G\ n\ x\ k)^{n-1}}\right). \end{aligned}$$

This sequence originates from Newton's method applied to the function $y \mapsto y^n - x$. Obviously, $G\ 2 = H$. By this formula and the modulus from Newton's method one could prove a general version of Theorem 5.4.10:

$$\forall_x.{}^{co}\mathbf{I}x \to 0 \leq x \to {}^{co}\mathbf{I}\sqrt[n]{x}$$

A difference of this generalisation is that one has to take a look at the first $n+1$ digits of the radicand to compute the first digit of the root. Another problem which increases the duration for higher roots is that in the definition of G one divides by $(G\ n\ x\ k)^{n-1}$. If $(G\ n\ x\ k)$ is small, $(G\ n\ x\ k)^{n-1}$ is even smaller for large n and the smaller the divisor the longer the duration of the division.

A generalisation of this is given in the context of functions with can be defined as power series. The main examples here are the trigonometric functions sin and cos. To prove that all real numbers in these sequences are in ${}^{co}\mathbf{I}$, one can use the formulas (5.1) and (5.2) and Lemma 5.3.4.

As an even larger generalisation one could pose the question: How can we use Theorem 5.3.12 to get an algorithm which takes a continuous function f from $[-1,1]$ to $[-1,1]$ and returns a computable function which takes a signed digit stream of a real x and returns a signed digit stream of $f(x)$? Constructive continuous functions are for example defined in [30, Section 2.4]. Uniformly continuous functions on signed digit stream were already considered by Ulrich Beger in [20]. Connected with this is the intermediate value theorem and the fundamental theorem of algebra. Both theorems are constructively considered in [152] by Peter Schuster and Helmut Schwichtenberg. Of course, the fundamental theorem of algebra needs complex numbers but they can be seen as pairs of real numbers.

Another direction in which one could extend this work is to replace the signed digit code by the Gray code. Similar to the signed digit code, the Gray code is suitable to represent real numbers [170]. In contrast to the signed digit code, the Gray code has the property that a small change in the value of a real number effects only a small change in the Gray code of this real number. To implement the Gray code, one needs simultaneously defined types and simultaneously coinductively defined predicates. In [160] there are proofs of the analogous statements to Lemma 5.3.2 and Lemma 5.3.4 and the analogous statements to (5.1) and (5.2).

Chapter 6

Rates of convergence for asymptotically weakly contrative maps in normed spaces

The results we present in this chapter are based on a collaboration with Thomas Powell [138].

Motivation 6.0.1. This chapter is about a typical usage of proof mining in analysis. We apply ideas and techniques from logic to classical proofs with the aim to get some quantitative information out of the proofs and generalise the proven theorems. In Chapter 3 we have seen an example of proof mining in algebra, and there are also applications of proof mining in number theory [117] and combinatorics [18]. However, analysis is the main field where proof mining is applied. [90] presents this in great detail and [92] shows the recent results of proof mining in non-linear analysis. However, we do not require any knowledge about proof mining and only require some basics of analysis.

In Chapter 5 we have already discussed the computational content of proofs in analysis. In contrast to that chapter we now use classical logic (e.g. $\mathbf{PA}^\omega$) and at some points also a slightly different notation. For instants, in this chapter we use the notion "rate of convergence" instead of "modulus of convergence" as the first one is more typical in the field of proof mining.

6.1 Introduction

Motivation 6.1.1. Let (X, d) be a complete metric space. A map $T : X \to X$ is a *contraction* if there is some $k \in [0, 1)$ such that

$$\forall_{x,y \in X} d(Tx, Ty) \leq kd(x, y).$$

Banach's fixed point theorem tells us that there exists a unique fixed point $q \in X$ of T and the Picard iteration $x_{n+1} := Tx_n$ for any arbitrary starting point $x_0 \in X$ converges to q such that

$$d(x_n, q) \leq \frac{k^n}{1-k} d(x_0, q)$$

for all $n \in \mathbb{N}$.

The aim of this chapter is to develop more theorems in this form by applying typical techniques of proof mining. Our approach is inspired by [100,103]. Instead of the Picard iteration, we consider the Krasnoselski-Mann iteration (Definition 6.2.7), and instead of contractions in metric spaces we consider variants of contractive maps in normed space, which were studied by Albert, Chidume, Guerre-Delabriere, Reich, Yao and Zegeye [2, 3, 6, 37] just to mention a few. We present these variants in Section 6.3. All these fixed point theorems use a certain convergence lemma. In Section 6.4 we take a classical proof of this lemma as starting point and extract a rate of convergence from this proof and reformulate this into a quantitative version of the convergence lemma. After this we present some case studies. In each case study we use the convergence lemma to prove a fixed point theorem which contains a rate of convergence, and we present some corollaries of this fixed point theorem, which are quantitative versions of some theorems in [2,3,37]. In particular, the main theorem in each case study is a generalisation of the theorem in our sources.

6.2 Basic definition

6.2.1 Rate of convergence and divergence

Motivation 6.2.1. The following notions are just quantitative versions of convergence and divergence. Note that they are quite different to the modulus of convergence from Chapter 5.

Definition 6.2.2. Let X be a metric space (with distance function d) and let $(\mu_n)_n$ be a sequence in X and $q \in X$. A *rate of convergence* for $(\mu_n)_n$ to q is a function

$f : \mathbb{R}^+ \to \mathbb{N}$ such that

$$\forall_{\varepsilon>0}\forall_{n\geq f(\varepsilon)} d(\mu_n, q) \leq \varepsilon.$$

Definition 6.2.3. Let $(\alpha_n)_n$ be a sequence of non-negative real numbers. We say that $r : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ is a *rate of divergence* for $(\sum_{i=0}^{n} \alpha_i)_n$ if

$$\forall_{n\in\mathbb{N},x\in\mathbb{R}^+} \sum_{i=n}^{r(n,x)} \alpha_i > x.$$

Example 6.2.4. For the constant sequence $(1)_n$ a rate of divergence of $\sum_{i=0}^{\infty} 1$ is given by

$$r(n, x) := \lceil x \rceil + n.$$

6.2.2 Iterative sequences

Motivation 6.2.5. The class of iterative sequences which we consider in our case studies are the so-called *Krasnoselski-Mann sequences* [2]. Krasnoselski-Mann sequences are general versions of the Picarde sequences from Motivation 6.1.1. By using non-expansive retractions we even get a general form of Krasnoselski-Mann sequences which we use in the second case study.

Definition 6.2.6. Let X be a normed space and $E \subseteq X$ be a subset. A map $P : X \to E$ is called *retraction* if $P|_E = \mathrm{id}_E$. It is called *non-expansive retraction* if additionally

$$\forall_{x,y\in X} \|Px - Py\| \leq \|x - y\|.$$

Definition 6.2.7. Let a normed space X, a subset $E \subseteq X$, a sequence $(A_n : E \to E)_n$ of maps and a sequence $(\alpha_n)_n$ of non-negative real numbers be given. We say that a sequence $(x_n)_n$ in E is a *Krasnoselski-Mann sequence* w.r.t. $(A_n)_n$ and $(\alpha_n)_n$ if

$$x_{n+1} = (1 - \alpha_n)x_n + \alpha_n A_n x_n$$

for all $n \in \mathbb{N}$.

For a non-expansive retraction P we say that $(x_n)_n$ is a *P-Krasnoselski-Mann sequence* w.r.t. $(A_n)_n$ and $(\alpha_n)_n$ if

$$x_{n+1} = P((1 - \alpha_n)x_n + \alpha_n A_n x_n)$$

for all $n \in \mathbb{N}$.

6.3 Notions of contractivity

Motivation 6.3.1. Most of the maps explored in this chapter can be viewed as generalisation of the concept of contractivity in the sense of Edelstein [67]. In this section we present several version of contractivity. We also discuss the connection between them.

6.3.1 Contractive and weakly contractive maps

Motivation 6.3.2. We start by defining contractivity and a modulus of uniformly contractivity. Such a modulus is discussed in [98], where it is later used to formulate a general rate of convergence for a variant of Edelstein's fixed point theorem for contractive maps.

Definition 6.3.3. Let X be a normed space, $E \subseteq X$ and suppose that $T : E \to E$ is a map. We say that T is *contractive* if for all $x, y \in E$,

$$x \neq y \Rightarrow \|Tx - Ty\| < \|x - y\|.$$

A function $\tau : \mathbb{R}^+ \to \mathbb{R}^+$ which satisfies

$$\forall_{x,y\in E}\forall_{\varepsilon>0}.\|x - y\| \geq \varepsilon \Rightarrow \|Tx - Ty\| + \tau(\varepsilon) \leq \|x - y\|$$

is called a *modulus of (uniform) contractivity.*

Motivation 6.3.4. A more restrictive notion of contractivity was introduced and studied by Alber and Guerre-Delabriere [3]. It is the so-called weak contractivity, which we present in the following definition. Here the modulus of contractivity is given by a function ψ with certain properties.

Definition 6.3.5. Let X be a normed space, $E \subseteq X$ and suppose that $T : E \to E$ is a map. T is called *ψ-weakly contractive* for some continuous and non-decreasing function $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ which is positive on $\mathbb{R}^+$ if for all $x, y \in E$,

$$\|Tx - Ty\| \leq \|x - y\| - \psi(\|x - y\|).$$

Remark 6.3.6. From the definitions above we directly see that if T is ψ-weakly contractive, it is also contractive with modulus ψ, and if T is contractive with any modulus, it is also contractive. The other directions do not hold without further assumptions:

We can reformulate contractivity in Definition 6.3.3 as follows:

$$\forall_{x,y\in E}\forall_{\varepsilon>0}\exists_{\delta>0}.\|x-y\|\geq\varepsilon\Rightarrow\|Tx-Ty\|+\delta\leq\|x-y\|.$$

For $\tau(\epsilon)$ instead of δ this is similar to the definition of contractivity with modulus τ. However, δ can also depend on x and y whereas $\tau(\varepsilon)$ cannot. Thus a contractive map with a modulus is contractive in a uniform way. In [98, Section 4.3] it is even shown that if E is compact, a modulus of contractivity can be characterised as the so-called monotone functional interpretation of the statement that T is contractive and in [98, Section 5.1] such a modulus is used to formulate a rate of convergence for Edelstein's fixed point theorem given in [67]. A special case of contractive maps with modulus are the almost uniformly contractive maps which are introduced in [33, Section 4].

In the case that T is contractive with modulus τ, we define

$$\psi(\varepsilon):=\inf\{\tau(\mu)\mid\varepsilon\leq\mu\}$$

for all $\varepsilon>0$ and $\psi(0):=0$. Obviously, ψ is non-decreasing. If furthermore ψ is positive on $\mathbb{R}^+$, T is ψ-weakly contractive. This assumption is not very strong, and in the most cases τ is already non-decreasing. Therefore, uniform contractivity with modulus and weak contractivity are closely connected.

In the next section we discuss a more general version of these notions.

6.3.2 Asymptotically weakly contractive maps

Motivation 6.3.7. Classes of asymptotically weakly contractive maps have been studied in several places, particularly by Albert and Guerre-Delabriere in [3, 4] and then subsequently by Chidume, Zegeye and Aneke in [37]. For a simple example, an asymptotic variant of Definition 6.3.5, i.e. of being ψ-weakly contractive, is given by the condition

$$\|T^nx-T^ny\|\leq(1+a_n)\|x-y\|-\psi(\|x-y\|)\tag{6.1}$$

where now $(a_n)_n$ is some sequence of non-negative reals satisfying $a_n\to 0$ for $n\to\infty$. Another example is given by

$$\|T^nx-T^ny\|\leq\|x-y\|-\psi(\|x-y\|)+l_n,$$

where $(l_n)_n$ is a sequence which converges to 0 for $n\to\infty$. In both cases we consider a sequence of maps which becomes weakly contractive in the limit. This leads to our central definition of asymptotically weakly contractive maps.

Definition 6.3.8. Let X be a normed space, $E \subseteq X$, $(\kappa_n)_n \in (\mathbb{R}_0^+)^{\mathbb{N}}$, $(A_n : E \to E)_n$ be a sequence of maps and $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$. We say that $(A_n)_n$ is *asymptotically ψ-weakly contractive* w.r.t. $(\kappa_n)_n$ and modulus $\sigma : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ if for all $x, y \in E$ and $\delta, K > 0$ we have

$$\forall_{n \geq \sigma(\delta,K)}. \|x - y\| \leq K \Rightarrow \|A_n x - A_n y\| \leq (1 + \kappa_n)\|x - y\| - \psi(\|x - y\|) + \delta.$$

For a given point $q \in X$ we say that $(A_n)_n$ is *quasi asymptotically ψ-weakly contractive* w.r.t. $(\kappa_n)_n$ and q and modulus σ if for all $x \in E$ and $\delta, K > 0$ we have

$$\forall_{n \geq \sigma(\delta,K)}. \|x - q\| \leq K \Rightarrow \|A_n x - q\| \leq (1 + \kappa_n)\|x - q\| - \psi(\|x - q\|) + \delta.$$

If we do not mention $(\kappa_n)_n$ explicitly, it is by default the zero sequence.

Remark 6.3.9. The definition above is divided into two parts as in many of the results we prove later it is sufficient that the map is quasi asymptotically ψ-weakly contractive instead of asymptotically ψ-weakly contractive. Note that in our cases q will be a fixed point of the A_n's and therefore asymptotic ψ-weak contractivity implies quasi asymptotic ψ-weak contractivity.

In the same way that the weakly contractive maps are generalised by the class of (uniformly) contractive maps with modulus, for any asymptotically weakly contractive map T, as in Motivation 6.3.7, the sequence $(T^n)_n$ is asymptotically weakly contractive in our new sense of Defintion 6.3.8. To see this, note that if $f : \mathbb{R}^+ \to \mathbb{R}^+$ is a rate of convergence for $a_n \to 0$ then it is clear that for any T satisfying (6.1), $\sigma(\delta, K) := f(\delta/K)$ forms a modulus of asymptotic contractivity for $(T^n)_n$.

However, as we shall see later, a number of more general notions of asympototic weak contractiveness are captured by Definition 6.3.8. For instant the following example comes from [37] and we will discuss some theorems containing it in Section 6.5.

Definition 6.3.10. Let X be a normed space, $E \subseteq X$ be a subset, $\phi, \psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be two non-decreasing functions both positive on $\mathbb{R}^+$, and $(k_n)_n, (l_n)_n$ be two sequences which both converge to 0. A sequence of maps $(A_n : E \to X)_n$ is called *totally asymptotically weakly contractive* if

$$\|A_n x - A_n y\| \leq \|x - y\| + k_n \phi(\|x - y\|) - \psi(\|x - y\|) + l_n$$

for all $x, y \in E$ and $n \in \mathbb{N}$.

We say that a map $T : E \to E$ is *totally asymptotically weakly contractive* if the sequence $(T^n)_n$ is.

Lemma 6.3.11. In the situation of Definition 6.3.10 let f, g be the rates of convergence of $(k_n)_n$ and $(l_n)_n$, respectively. Then the sequence $(A_n)_n$ is ψ-asymptotically contractive with modulus

$$\sigma(\delta, K) := \max\left\{ f\left(\frac{\delta}{2\phi(K)}\right), g\left(\frac{\delta}{2}\right)\right\}.$$

Proof. Since $(A_n)_n$ is totally asymptotically weakly contractive, we have

$$\|A_n x - A_n y\| \leq \|x - y\| + k_n\phi(\|x - y\|) - \psi(\|x - y\|) + l_n$$

for all $x, y \in E$ and $n \in \mathbb{N}$. As ϕ is non-decreasing, it follows

$$\|x - y\| \leq K \Rightarrow \|A_n x - A_n y\| \leq \|x - y\| - \psi(\|x - y\|) + k_n\phi(K) + l_n$$

for all $x, y \in E$, $n \in \mathbb{N}$ and $K > 0$. Hence, $(A_n)_n$ is ψ-asymptotically contractive with the modulus σ as above. □

Remark 6.3.12. A related notion of being asymptotically contractive in the setting of arbitrary complete metric spaces (X, d) is given by Kirk [85], where a convergence result for Picard iterates is proven. This has been analysed from a proof theoretic standpoint first by Gerhardy [75] and then Briseid [34], both of whom develop quantitative notions of being asymptotically contractive (see [75, Definition 2] and [34, Definition 3.10]) similar in spirit to Definition 6.3.8 in this chapter. However, we are interested in the rates of convergence of the Krasnoselski-Mann iteration in normed spaces, and the proofs that we analyse have a very different character.

6.3.3 d-weakly contractive maps

Motivation 6.3.13. Another version of weakly contractive maps can be formulated by using the normalized duality map. These are particularly interesting for us, as the associated convergence results often rely on geometric properties of the underlying space, such as uniform smoothness, allowing us to produce rates of convergence in terms of the corresponding moduli as we see in Section 6.6.

Definition 6.3.14. Let X be a normed space. We denote its dual space by X^*.[1] The *normalized duality map* $J : X \to 2^{X^*}$ is defined by

$$Jx := \{j \in X^* \mid \langle x, j\rangle = \|x\|^2 = \|j\|^2\},$$

where $\langle x, j\rangle := j(x)$ denotes the duality pairing.

[1] In this chapter there is no list type. Therefore, X^* does always denote the dual space.

Motivation 6.3.15. Following [4], a map $T : E \to E$ is called *d-weakly contractive* w.r.t. some non-decreasing $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ positive on $\mathbb{R}^+$ if for all $x, y \in E$ there exists some $j \in J(x-y)$ such that

$$|\langle Tx - Ty, j\rangle| \leq \|x-y\|^2 - \psi(\|x-y\|^2).$$

The following definition introduces a class of asymptotically d-weakly contractive maps analogously to Definition 6.3.8.

Definition 6.3.16. Let X be a normed space and J be the normalized duality map and let $E \subseteq X$ be a subset. A sequence $(A_n : E \to X)$ of maps is called *asymptotically d-weakly contractive* with modulus $\sigma : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ if there exists some non-decreasing $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ which is positive on $\mathbb{R}^+$ such that for all $x, y \in E$ there exists $j \in J(x-y)$ with

$$\forall_{b,\delta>0}\forall_{n\geq\sigma(\delta,b)}.\|x-y\| \leq b \Rightarrow |\langle A_n x - A_n y, j\rangle| \leq \|x-y\|^2 - \psi(\|x-y\|) + \delta.$$

We say that $(A_n)_n$ is *quasi asymptotically d-weakly contractive* w.r.t. q if for all $x \in E$ there exists $j \in J(x-y)$ with

$$\forall_{b,\delta>0}\forall_{n\geq\sigma(\delta,b)}.\|x-q\| \leq b \Rightarrow |\langle A_n x - q, j\rangle| \leq \|x-q\|^2 - \psi(\|x-q\|) + \delta.$$

Remark 6.3.17. Similar to Definition 6.3.8 one could replace $\|x-y\|^2$ by $(1+\kappa_n)|\|x-y\|^2$ for some non negative sequence $(\kappa_n)_n$. However, we will not need this in our case studies. But if one adds $(1+\kappa_n)$ in Definition 6.3.16, it is a generalisation of Definition 6.3.8:

Given any asymptotically ψ-weakly contractive sequence of maps $(A_n)_n$ w.r.t. $(\kappa_n)_n$ and modulus σ, for $x, y \in E$ with $\|x-y\| < K$ there exists $j \in J(x-y)$ by the Hahn–Banach theorem. For $\delta, K > 0$, $n \geq \sigma(\delta/K, K)$ and $x, y \in E$ with $\|x-y\| \leq K$ we have

$$\begin{aligned}
|\langle A_n x - A_n y, j\rangle| + \|x-y\|\psi(\|x-y\|) &\leq \|A_n x - A_n y\|\|x-y\| + \|x-y\|\psi(\|x-y\|) \\
&\leq \|x-y\|(\|A_n x - A_n y\| + \psi(\|x-y\|)) \\
&\leq \|x-y\|\left((1+\kappa_n)\|x-y\| + \frac{\delta}{K}\right) \\
&\leq (1+\kappa_n)\|x-y\|^2 + \delta.
\end{aligned}$$

Thus $(A_n)_n$ is asymptotically d-contractive w.r.t. $\chi(\varepsilon) := \varepsilon\psi(\varepsilon)$ and modulus $\rho(\delta, K) := \sigma(\delta/K, K)$.

6.4 Quantitative recursive inequalities

Motivation 6.4.1. We want to extract rates of convergence from generalisations of a number of convergence proofs, which all involve variants of contractive maps as discussed on Section 6.3. These proofs utilise an abstract theory of recursive inequalities, a quantitative analysis of which is not only crucial in obtaining our rates of convergence, but forms a unifying scheme which, together with the abstract notions of contractivity discussed above, allows us to bring together several distinct convergence results from the literature. For illustrative purposes and to motivate the results in this section, we start with an example:

Example 6.4.2. Let X be a normed space, $E \subseteq X$ a subset and $T : E \to E$ a map satisfying

$$\forall_{x,y\in E}\|T^n x - T^n y\| \leq \|x - y\| - \psi(\|x - y\|) + l_n$$

for some $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ which is positive on $\mathbb{R}^+$ and some non-negative sequence $(l_n)_n$ which converges to 0. Suppose that T has some fixed point $q \in E$ and let a sequence $(x_n)_n$ in E be given such that

$$x_{n+1} = (1 - \alpha_n)x_n + \alpha_n T^n x_n$$

for some sequence $(\alpha_n)_n$ in $(0, 1]$ satisfying $\sum_{n=0}^{\infty} \alpha_n = \infty$ (i.e. $(x_n)_n$ is a Krasnoselski-Mann sequence). We observe

$$\begin{aligned}\|x_{n+1} - q\| &\leq (1 - \alpha_n)\|x_n - q\| + \alpha_n\|T^n x_n - T^n q\| \\ &\leq (1 - \alpha_n)\|x_n - q\| + \alpha_n(\|x_n - q\| - \psi(\|x_n - q\|) + l_n) \\ &\leq \|x_n - q\| - \alpha_n\psi(\|x_n - q\|) + \alpha_n l_n\end{aligned}$$

and therefore the sequence $(\mu_n)_n := (\|x_n - q\|)_n$ satisfies the following recursive inequality:

$$\mu_{n+1} \leq \mu_n - \alpha_n\psi(\mu_n) + \alpha_n l_n$$

From a general theory of recursive inequalities of this kind given in [1, 5] it follows that $\mu_n \to 0$ for $n \to \infty$. A more specific treatment of this recursive inequality is given in [7, Lemma 2.5].

Many convergence theorems about variants of weakly contractive maps use this inequality in their proofs. Therefore, we start with an quantitative analysis of this recursive inequality.

6.4.1 Main quantitative lemmas

Motivation 6.4.3. The following lemma about the recursive inequality from Example 6.4.2 is the basis of this chapter. Here we see the strength of proof mining. We first present a standard proof which is inspired by [7]. In particular, it is a special case of Lemma 2.5 and mentioned in Remark 2.6 of [7]. After formulating this proof, we show how one can extract some quantitative information out of it, and even how to generalise the lemma.

In [103, Lemma 1] and [100, Lemma 3.4] the similar (but not identical) recursive inequality

$$\mu_{n+1} \leq \mu_n - \alpha_n \psi(\mu_{n+1}) + \alpha_n l_n$$

is considered. However, the respective proofs that $\mu_n \to 0$ are different, and therefore the results which follow are new.

Lemma 6.4.4. Let $(\mu_n)_n$, $(\alpha_n)_n$ and $(\gamma_n)_n$ be sequences of non-negative real numbers and $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be a non-decreasing function which is positive on $\mathbb{R}^+$ and $\psi(0) = 0$ such that $(\alpha_n)_n$ is positive and bounded, $\sum_{n=0}^{\infty} \alpha_n = \infty$, $\frac{\gamma_n}{\alpha_n} \to 0$ for $n \to \infty$ and

$$\mu_{n+1} \leq \mu_n - \alpha_n \psi(\mu_n) + \gamma_n \tag{6.2}$$

for all $n \in \mathbb{N}$. Then $\mu_n \to 0$ for $n \to \infty$.

Proof. First, we show that there is a subsequence $(\mu_{k_n})_n$ which converges to 0. We assume that this is not the case. Then there are $\varepsilon > 0$ and $M \in \mathbb{N}$ such that $\mu_n \geq \varepsilon$ for all $n \geq M$, and therefore $\psi(\mu_n) \geq \psi(\varepsilon)$ for all $n \geq M$. Since $\left(\frac{\gamma_n}{\alpha_n}\right)_n$ converges to 0, we take $N \geq M$ such that $\frac{\gamma_n}{\alpha_n} < \frac{\psi(\varepsilon)}{2}$ for all $n \geq N$. By iterating (6.2) we have

$$\begin{aligned}\mu_{n+1} &\leq \mu_N - \sum_{i=N}^{n} \alpha_i \left(\psi(\mu_i) - \frac{\gamma_i}{\alpha_i}\right) \leq \mu_N - \sum_{i=N}^{n} \alpha_i \left(\psi(\varepsilon) - \frac{\psi(\varepsilon)}{2}\right) \\ &\leq \mu_N - \frac{\psi(\varepsilon)}{2} \sum_{i=N}^{n} \alpha_i\end{aligned}$$

for all $n \geq N$. But $(\sum_{i=N}^{n} \alpha_i)_n$ converges to ∞ and therefore $\mu_N - \frac{\psi(\varepsilon)}{2} \sum_{i=N}^{n} \alpha_i$ becomes finally negative, which is a contradiction as $(\mu_n)_n$ is a sequence of non-negative real numbers. Therefore, there must be a subsequence $(\mu_{k_n})_n$ which converges to 0.

Now, we show that $(\mu_n)_n$ convergence to 0. Hence, let $\varepsilon > 0$ be given. Since $(\mu_{k_n})_n$ and $\left(\frac{\gamma_n}{\alpha_n}\right)_n$ both converge to 0 and $(\alpha_n)_n$ bounded, we take k_n sufficiently

large such that $\mu_{k_n} \leq \frac{\varepsilon}{2}$, $\gamma_m \leq \frac{\varepsilon}{2}$ and $\frac{\gamma_m}{\alpha_m} \leq \frac{\psi(\varepsilon/2)}{2}$ for all $m \geq k_n$. We show $\mu_m < \varepsilon$ for all $m \geq k_n$ by induction. For k_n this is obvious true. For the induction step assume $\mu_m < \varepsilon$ for some $m \geq k_n$. We first consider the case $\mu_m \geq \frac{\varepsilon}{2}$, then

$$\mu_{m+1} \leq \mu_m - \alpha_m \left(\psi(\mu_m) - \frac{\gamma_m}{\alpha_m}\right) \leq \mu_m - \alpha_m \left(\psi\left(\frac{\varepsilon}{2}\right) - \frac{\psi(\varepsilon/2)}{2}\right) < \mu_m < \varepsilon.$$

Now we consider the case $\mu_m < \frac{\varepsilon}{2}$. Here we have

$$\mu_{m+1} \leq \mu_m - \alpha_m \psi\left(\frac{\varepsilon}{2}\right) + \frac{\varepsilon}{2} < \varepsilon.$$

Hence, in both cases we have $\mu_{m+1} \leq \varepsilon$. □

Remark 6.4.5. First, we see that the conclusion of the lemma above is that the sequence $(\mu_n)_n$ converges to 0. Therefore, in a quantitative version we expect a rate of convergence of the sequence $(\mu_n)_n$ to 0.

As input we expect a rate of convergence for $\left(\frac{\gamma_n}{\alpha_n}\right)_n$, a bound $\alpha > 0$ for $(\alpha_n)_n$ and a rate of divergence of $\sum_{i=0}^{\infty} \alpha_i$.

Considering the proof, we see that the first part of the proof is not constructive as it is proven by contradiction: We assume that there is no adequate subsequence and the contradiction comes from

$$\mu_{n+1} \leq \mu_N - \frac{\psi(\varepsilon)}{2} \sum_{i=N}^{n} \alpha_i,$$

$\sum_{i=0}^{\infty} \alpha_i = \infty$ and $(\mu_n)_n \in (\mathbb{R}_0^+)^{\mathbb{N}}$. But to get a contradiction, we do not need all members of the sequences $(\mu_n)_n$ and $(\alpha_n)_n$. We only need them up to an l such that $\mu_N - \frac{\psi(\varepsilon)}{2} \sum_{i=N}^{l} \alpha_i$ is negative. Then the assumption $\mu_i \geq \varepsilon$ can not be true for all $i \in [N, l]$ and we have the upper bound l for the next member of the subsequence. In particular, provided that $\leq$ is decidable, we have a strong existence proof of the formula

$$\forall_{\epsilon>0, M \in \mathbb{N}} \exists_{n \geq M} \mu_n < \varepsilon.$$

This corresponds to a function $F : \mathbb{R}^+ \to \mathbb{N} \to \mathbb{N}$ with $\mu_{F(\varepsilon,M)} < \varepsilon$ and $F(\varepsilon, M) \geq M$ for all $M \in \mathbb{N}$ and $\varepsilon > 0$.

In contrast to the first part, the second part of the proof is already constructive and we just need the given rates of convergences and divergences, respectively, and the function F from above.

Furthermore, looking at the proof one sees that it is sufficient if Formula (6.2) holds for sufficiently large n and the sequence $\left(\frac{\gamma_n}{\alpha_n}\right)_n$ can be seen as an error which becomes arbitrary small for large n. These two considerations can be combined as we will see in the following lemma.

Lemma 6.4.6. Let $(\mu_n)_n$ and $(\alpha_n)_n$ be two sequences of non-negative real numbers, and $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be a non-decreasing function which is positive on $\mathbb{R}^+$. Suppose that there are $\alpha > 0$, $r : \mathbb{R}^+ \to \mathbb{R}^+ \to \mathbb{N}$ and $N : \mathbb{R}^+ \to \mathbb{N}$ such that

- $\alpha_n \in (0, \alpha]$ for all $n \in \mathbb{N}$,
- r is a rate of divergence for $\sum_{n=0}^{\infty} \alpha_n$,

and

$$\mu_{n+1} \leq \mu_n - \alpha_n(\psi(\mu_n) - \delta) \tag{6.3}$$

for all $\delta > 0$ and $n \geq N(\delta)$. Then $\mu_n \to 0$ for $n \to \infty$ with the rate of convergence

$$\Phi_{\psi,(\mu_n)_n,\alpha,N}(\varepsilon) := r\left(M_{\psi,\alpha}(\varepsilon), 2\int_{\frac{\varepsilon}{2}}^{\mu_{M_{\psi,\alpha}(\varepsilon)}} \frac{dt}{\psi(t)}\right) + 1,$$

where

$$M_{\psi,\alpha}(\varepsilon) := N\left(\frac{1}{2}\min\left\{\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}\right).$$

Proof. We first show that for any given $\zeta > 0$,

$$N_0(\zeta) := N\left(\min\left\{\frac{\psi(\zeta)}{2}, \frac{\zeta}{\alpha}\right\}\right) \qquad \text{and}$$
$$l := r\left(N_0(\zeta), 2\int_{\zeta}^{\mu_{N_0(\zeta)}} \frac{dt}{\psi(t)}\right)$$

there is a natural number $m \in [N_0(\zeta), l+1]$ with $\mu_m < \zeta$: By (6.3) we have both

$$\mu_{n+1} \leq \mu_n - \alpha_n\left(\psi(\mu_n) - \frac{\psi(\zeta)}{2}\right) \tag{6.4}$$

and (using that $\alpha_n \leq \alpha$)

$$\mu_{n+1} \leq \mu_n - \alpha_n\psi(\mu_n) + \zeta \tag{6.5}$$

for all $n \geq N_0(\zeta)$. Now suppose that $\zeta \leq \mu_n$ for all $N_0(\zeta) \leq n \leq l+1$. Then by monotonicity of ψ we have $0 < \psi(\zeta) \leq \psi(\mu_n)$, and thus by (6.4) it follows

$$\mu_{n+1} \leq \mu_n - \alpha_n \left(\psi(\mu_n) - \frac{\psi(\mu_n)}{2} \right) = \mu_n - \alpha_n \frac{\psi(\mu_n)}{2}.$$

Therefore we obtain

$$\frac{1}{2} \sum_{n=N_0}^{l} \alpha_n \leq \sum_{n=N_0(\zeta)}^{l} \frac{\mu_n - \mu_{n+1}}{\psi(\mu_n)} \leq \sum_{n=N_0(\zeta)}^{l} \int_{\mu_{n+1}}^{\mu_n} \frac{dt}{\psi(t)} = \int_{\mu_{l+1}}^{\mu_{N_0(\zeta)}} \frac{dt}{\psi(t)},$$

where for the second inequality we observe that for $N_0(\zeta) \leq n \leq l$ we have $0 < \mu_{n+1} < \mu_n$ and thus the function $t \mapsto \frac{1}{\psi(t)}$ is well-defined, positive valued and monotonically decreasing on $[\mu_{n+1}, \mu_n]$, and hence integrable with

$$\frac{\mu_n - \mu_{n+1}}{\psi(\mu_n)} \leq \int_{\mu_{n+1}}^{\mu_n} \frac{dt}{\psi(t)}.$$

Finally, since $\zeta \leq \mu_{l+1} < \mu_{N_0(\zeta)}$ we have

$$\sum_{n=N_0(\zeta)}^{l} \alpha_n \leq 2 \int_{\zeta}^{\mu_{N_0(\zeta)}} \frac{dt}{\psi(t)}$$

which is a contraction to the definition of l and the assumption that r is the rate of divergence of $\sum_{n=0}^{\infty} \alpha_n$. Therefore, our claim is proven.

Now, we show that $\Phi_{\psi,(\mu_n)_n,\alpha,N}$ is a rate of convergence. Let $\varepsilon > 0$ be given. By the claim above, there is $m \leq r\left(N_0\left(\frac{\varepsilon}{2}\right) + 1, 2\int_{\frac{\varepsilon}{2}}^{\mu_{N_0(\frac{\varepsilon}{2})}} \frac{dt}{\psi(t)}\right)$ with $\mu_m < \frac{\varepsilon}{2}$ and $m \geq N_0(\varepsilon/2)$. We now claim, that $\mu_k < \varepsilon$ for all $k \geq m$. This is shown by induction, where the base case is trivial. For the induction step we deal with two cases: Firstly, if $\frac{\varepsilon}{2} \leq \mu_k < \varepsilon$ then since $k \geq m \geq N_0(\varepsilon/2)$ and $\psi(\varepsilon/2) \leq \psi(\mu_k)$, it follows from (6.4) that

$$\mu_{k+1} \leq \mu_k - \alpha_k \left(\psi(\mu_k) - \frac{\psi(\varepsilon/2)}{2} \right) \leq \mu_k - \alpha_k \frac{\psi(\varepsilon/2)}{2} < \mu_k < \varepsilon.$$

On the other hand, if $\mu_k < \frac{\varepsilon}{2}$ then from (6.5) we have

$$\mu_{k+1} \leq \mu_k - \alpha_k \psi(\mu_k) + \frac{\varepsilon}{2} \leq \mu_k + \frac{\varepsilon}{2} < \varepsilon.$$

This proves the claim, and thus it follows that $\mu_k \leq \varepsilon$ for all

$$k \geq r\left(N_0\left(\frac{\varepsilon}{2}\right), 2\int_{\frac{\varepsilon}{2}}^{\mu_{N_0(\frac{\varepsilon}{2})}} \frac{dt}{\psi(t)}\right) + 1.$$

Writing out $N_0\left(\frac{\varepsilon}{2}\right)$ in full gives us the rate of convergence as above. □

Remark 6.4.7. If the sequence $(\mu_n)_n$ from the lemma above is uniformly bounded by some $c > 0$, we can replace $\mu_{M_{\phi,\alpha}(\varepsilon)}$ in the rate of convergence by c. In this way, the rate of convergence can be formulated independently of the sequence $(\mu_n)_n$ as follows:

$$\Phi_{\psi,c,\alpha,N}(\varepsilon) := r\left(N\left(\frac{1}{2}\min\left\{\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}\right), 2\int_{\frac{\varepsilon}{2}}^{c} \frac{dt}{\psi(t)}\right) + 1,$$

Lemma 6.4.8. Let $(\mu_n)_n$, $(\alpha_n)_n$ and $(\beta_n)_n$ be sequences of non-negative real numbers and $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be a non-decreasing function which is positive on $\mathbb{R}^+$. Suppose that there are $\alpha > 0$, $d \geq 1$, $r : \mathbb{N} \to \mathbb{R}^+ \to \mathbb{N}$ and $N : \mathbb{R}^+ \to \mathbb{N}$ such that

- $\alpha_n \in (0, \alpha]$ for all $n \in \mathbb{N}$,
- $\prod_{i=0}^{n}(1 + \beta_i) \leq d$ for all $n \in \mathbb{N}$,
- r is a rate of divergence for $\sum_{n=0}^{\infty} \alpha_n$,

and

$$\mu_{n+1} \leq (1 + \beta_n)\mu_n - \alpha_n(\psi(\mu_n) - \delta)$$

for all $\delta > 0$ and $n \geq N(\delta)$. Then $\mu_n \to 0$ for $n \to \infty$ with the rate of convergence

$$\Phi_{\psi,(\mu_n)_n,\alpha,d,r,N}(\varepsilon) := r\left(M_{\psi,\alpha,d}(\varepsilon), 2d\int_{\frac{\varepsilon}{2}}^{\mu_{M_{\psi,\alpha,d}(\varepsilon)}} \frac{dt}{\psi(t)}\right) + 1,$$

where

$$M_{\psi,\alpha,d}(\varepsilon) := N\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon/2)}{d}, \frac{\varepsilon}{\alpha}\right\}\right).$$

Proof. We define $\lambda_n := \frac{\mu_n}{\prod_{i=0}^{n-1}(1+\beta_i)}$ for all $n \in \mathbb{N}$. Then for any $\delta > 0$ and all $n \geq N(\delta)$ is follows

$$\lambda_{n+1} \leq \lambda_n - \frac{\alpha_n\psi(\mu_n)}{\prod_{i=0}^{n}(1+\beta_i)} + \frac{\alpha_n\delta}{\prod_{i=0}^{n}(1+\beta_i)} \leq \lambda_n - \alpha_n d^{-1}\psi(\mu_n) + \alpha_n\delta,$$

where we have used $1 \leq \prod_{i=0}^{n}(1+\beta_i) \leq d$. Moreover, since $\mu_n = \lambda_n \prod_{i=0}^{n-1}(1+\beta_i) \geq \lambda_n$ for all $n \in \mathbb{N}$, by monotonicity of ψ we have $\psi(\mu_n) \geq \psi(\lambda_n)$ and thus

$$\lambda_{n+1} \leq \lambda_n - \alpha_n(\phi(\lambda_n) - \delta) \tag{6.6}$$

for $\phi(t) := d^{-1}\psi(t)$, which is clearly non-decreasing and positive on $\mathbb{R}^+$. Observing finally that $\lambda_{M_{\psi,\alpha,d}(\varepsilon)} \leq \mu_{M_{\psi,\alpha,d}(\varepsilon)}$ and applying Lemma 6.4.6, we obtain the stated rate of convergence. □

Remark 6.4.9. Similar to Lemma 6.4.6, if there is a uniform bound $c > 0$ of the sequence $(\mu_n)_n$, it is also a uniform bound of $(\lambda_n)_n$. Hence, in this case we can formulate the rate of convergence in the lemma above independently of the sequence $(\mu_n)_n$ by

$$\Phi_{\psi,c,\alpha,d,r,N}(\varepsilon) := r\left(N\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon/2)}{d}, \frac{\varepsilon}{\alpha}\right\}\right), 2d\int_{\frac{\varepsilon}{2}}^{c}\frac{dt}{\psi(t)}\right) + 1.$$

6.4.2 Reformulation in terms of traditional rates of convergence

Motivation 6.4.10. We now give a rough translation of our main quantitative result phrased in terms of direct rates of convergence, where we seek some explicit function $f : \mathbb{N} \to \mathbb{R}^+$ with $\lim_{n\to\infty} f(n) = 0$ such that $\mu_n \leq f(n)$ for all $n \in \mathbb{N}$. Being able to provide an elegant version of this kind typically requires additional assumptions, such as the existence of inverse functions, and thus we prefer our formulation above. However, the translation we provide in the following proposition is comparable with known convergence rates in the literature. In particular, [3] provides several theorems in this form.

Proposition 6.4.11. In the situation of Lemma 6.4.8 we additionally assume that there is no n such that $\mu_i = 0$ for all $i \geq n$, there is an upper bound $c > 0$ for $(\mu_n)_n$, $\Psi : \mathbb{R}^+ \to \mathbb{R}$ is an antiderivative of $\frac{1}{\psi}$, and $N : \mathbb{R}^+ \to \mathbb{N}$ is non-increasing. Let $\tilde{N} : \mathbb{R}^+ \to \mathbb{R}^+$ be a non-increasing and continuous map such that $\forall_{t>0} N(t) \leq \tilde{N}(t)$. We define $F : \mathbb{R}^+ \to \mathbb{R}$ by

$$F(t) := 2d\Psi(t/2) - \alpha\tilde{N}\left(\frac{1}{2}\min\left\{\frac{\psi(t/2)}{d}, \frac{t}{\alpha}\right\}\right).$$

Then F is strictly decreasing with range $(-\infty, L)$, where $L := \lim_{t\to\infty} F(t) \in \mathbb{R} \cup \{\infty\}$.

Furthermore, let $F^{-1} : (-\infty, L) \to \mathbb{R}^+$ be the inverse function of F, then for all $n \geq r(0, 2d\Psi(c) - L) + 2$, we have

$$\mu_n \leq F^{-1}\left(2d\Psi(c) - \sum_{i=0}^{n-2} \alpha_i\right).$$

Proof. As $\frac{1}{\psi}$ is positive on $\mathbb{R}^+$, its antiderivative Ψ is strictly increasing. Furthermore, $-\tilde{N}$ is non-decreasing, and thus F is strictly increasing. Therefore $L = \lim_{t\to\infty} F(t)$ exists in $\mathbb{R} \cup \{\infty\}$.

Next we show $\lim_{t\to 0} F(t) = -\infty$. As Ψ is strictly increasing, it suffices to show $\lim_{t\to 0} \tilde{N}(t) = \infty$. Thus, we consider the case that this in not true. As $\tilde{N}$ is non-increasing, there must be some $k \in \mathbb{N}$ such that $\tilde{N}(t) \leq k$ and thus $N(t) \leq k$ for all $t \in \mathbb{R}^+$. Defining λ_n as in the proof of Lemma 6.4.8, then by the property of N and (6.6) it follows

$$\lambda_{n+1} \leq \lambda_n - \alpha_n d^{-1} \psi(\lambda_n)$$

for all $n \geq k$. Hence, it there is an $n \geq k$ with $\lambda_n = 0$, we have $\lambda_i = 0$ for all $i \geq n$, but this is a contraction to the assumption. Therefore $\lambda_n \neq 0$ and thereby $\psi(\lambda_n) \neq 0$ for all $n \geq k$, and we obtain

$$\alpha_n d^{-1} \leq \frac{\lambda_n - \lambda_{n+1}}{\psi(\lambda_n)} \leq \int_{\lambda_{n+1}}^{\lambda_n} \frac{dt}{\psi(t)}.$$

Using this inequality, for arbitrary $m > k$ we have

$$\Psi(\lambda_k) - \Psi(\lambda_m) = \sum_{n=k}^{m-1} \int_{\lambda_{n+1}}^{\lambda_n} \frac{dt}{\psi(t)} \geq d^{-1} \sum_{n=k}^{m-1} \alpha_n \to \infty$$

for $m \to \infty$ and thus $\Psi(\lambda_m) \to -\infty$. Therefore, also in this case $F(t) \to -\infty$ for $t \to 0$.

As F is strictly increasing and continuous, $\lim_{t\to\infty} F(t) = L$ and $\lim_{t\to-\infty} F(t) = -\infty$, the range of F is $(-\infty, L)$ and $F^{-1} : (-\infty, L) \to \mathbb{R}^+$ exists.

Now let $n \geq r(0, 2d\Psi(c) - L) + 2$ be given. By the definition of r, we have $2d\Psi(c) - \sum_{i=0}^{n-2} \alpha_i < L$, and therefore $F^{-1}\left(2d\Psi(c) - \sum_{i=0}^{n-2} \alpha_i\right)$ is well-defined. We define $\varepsilon_n := F^{-1}\left(2d\Psi(c) - \sum_{i=0}^{n-2} \alpha_i\right)$ and we have to show that $\mu_n \leq \varepsilon_n$. By Lemma 6.4.8 and Remark 6.4.9, we have $\mu_m \leq \varepsilon_n$ for all $m \geq \Phi_{\psi,c,\alpha,d,r',N}(\varepsilon_n)$, where r' is any rate of divergence for $\sum_{i=0}^{\infty} \alpha_i$. Hence, it suffices to show that $n \geq \Phi_{\psi,c,\alpha,d,r',N}(\varepsilon_n)$, or in other words

$$n \geq r'\left(N\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon_n/2)}{d}, \frac{\varepsilon_n}{\alpha}\right\}\right), 2d\int_{\frac{\varepsilon_n}{2}}^{c} \frac{dt}{\psi(t)}\right) + 1 \tag{6.7}$$

for any rate of divergence r'. Suppose therefore that r' is given by

$$r'(m,x) := \min\left\{ k \in \mathbb{N} \;\middle|\; \sum_{i=m}^{k} \alpha_i > x \right\}.$$

This is well-defined as $r(m,x) \in \left\{ k \in \mathbb{N} \;\middle|\; \sum_{i=m}^{k} \alpha_i > x \right\}$, and $n \geq r'(m,x)$ is equivalent to $\sum_{i=m}^{n} \alpha_i > x$. Therefore (6.7) is equivalent to

$$\sum_{i=N_0}^{n-1} \alpha_i > 2d \int_{\frac{\varepsilon_n}{2}}^{c} \frac{dt}{\psi(t)} = 2d(\Psi(c) - \Psi(\varepsilon_n/2))$$

for $N_0 := N\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon_n/2)}{d}, \frac{\varepsilon_n}{\alpha}\right\}\right)$. This can be reformulated as

$$2d\Psi(\varepsilon_n/2) - \sum_{i=0}^{N_0-1} \alpha_i > 2d\Psi(c) - \sum_{i=0}^{n-1} \alpha_i.$$

We establish this inequality as follows:

$$\begin{aligned}
2d\Psi(c) - \sum_{i=0}^{n-1} \alpha_i &< 2d\Psi(c) - \sum_{i=0}^{n-2} \alpha_i \\
&= F(\varepsilon_n) \\
&= 2d\Psi(\varepsilon_n/2) - \alpha\tilde{N}\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon_n/2)}{d}, \frac{\varepsilon_n}{\alpha}\right\}\right) \\
&\leq 2d\Psi(\varepsilon_n/2) - \alpha N_0 \\
&\leq 2d\Psi(\varepsilon_n/2) - \sum_{i=0}^{N_0-1} \alpha_i
\end{aligned}$$

□

6.4.3 Weakly contractive maps: a simple case study

Motivation 6.4.12. We conclude this section by demonstrating that a quantitative convergence result for weakly contractive maps already established in [3] falls out as a simple case of our framework. In particular, we formulate a proposition which is inspired by [3, Theorem 3.1] and afterwards we compare it with the original source. In the next sections we consider more complex case studies.

Proposition 6.4.13. Let X be a normed space, $E \subseteq X$ a subset and $T : E \to X$ be a map which satisfies

$$\forall_{x,y \in E}. \|Tx - Ty\| \leq \|x - y\| - \psi(\|x - y\|) \tag{6.8}$$

for some non-decreasing function $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ which is positive on $\mathbb{R}^+$. Let $q \in E$ be a fixed point of T, and suppose that $(x_n)_n$ is a sequence in E such that $x_{n+1} = Tx_n$ for all $n \in \mathbb{N}$. Then $\|x_n - q\| \to 0$ for $n \to \infty$ with rate

$$\Phi_{x_0,q,\psi}(\varepsilon) := \left\lceil 2 \int_{\frac{\varepsilon}{2}}^{\|x_0 - q\|} \frac{dt}{\psi(t)} \right\rceil + 1.$$

Furthermore, if $\Psi : \mathbb{R}^+ \to \mathbb{R}$ is an antiderivation of $\frac{1}{\psi}$, then either $\|x_n - q\| = 0$ for all but finitely many $n \in \mathbb{N}$, or

$$\|x_n - q\| \leq 2\Psi^{-1}\left(\Psi(\|x_0 - q\|) - \frac{n-1}{2}\right)$$

for all $n \geq 2$.

Proof. For all $n \in \mathbb{N}$ we have

$$\|x_{n+1} - q\| = \|Tx_n - q\| \leq \|x_n - q\| - \psi(\|x_n - q\|). \tag{6.9}$$

Therefore, we can apply Lemma 6.4.6 together with Remark 6.4.7, where $\alpha_n = 1$, $N(\delta) = 0$, $r(n, x) = n + \lceil x \rceil$, and $\|x_0 - q\|$ is an upper bound for $(\|x_n - q\|)_n$ by induction and (6.9).

For the second part of the statement we assume that there is no $n \in \mathbb{N}$ such that $\|x_i - q\| = 0$ for all $i \geq 0$. Then the inequity follows from Proposition 6.4.11: We have

$$\|x_n - q\| \leq F^{-1}\left(2\Psi(\|x_0 - q\|) - \sum_{i=0}^{n-2} \alpha_i\right)$$

for all $n \geq \lceil 2\Psi(c) - \lim_{t\to\infty} 2\Psi(t) \rceil + 2$, where $F(t) := 2\Psi\left(\frac{t}{2}\right)$ and $\sum_{i=0}^{n-2} \alpha_i = n - 1$. As Ψ is strictly increasing, we have that $n \geq \lceil 2\Psi(c) - \lim_{t\to\infty} 2\Psi(t) \rceil + 2$ hold for all $n \geq 2$, and by $F(t) = 2\Psi\left(\frac{t}{2}\right)$ it follows $F^{-1}(t) = 2\Psi^{-1}(\frac{t}{2})$. Putting things together yields the inequality from above. □

Remark 6.4.14. In contrast to Theorem 3.1 of [3] we have assumed that the fixed point q already exists as we are mainly interested in the rate of convergence. Because

of this we are able to omit certain assumptions of T which are only necessary to prove the existence of a fixed point. For example we do not need that ψ is continuous or $\lim_{t\to\infty} \psi(t) = \infty$.

Similar, we have assumed that the sequence $(x_n)_n$ already exists and has the property $x_{n+1} = Tx_n$ for all $n \in \mathbb{N}$, rather then demanding any additional properties of T and E which would ensure that the Picard iteration can be generated from any initial point x_0.

Other properties can be inferred from our assumptions. For instance, $\psi(0) = 0$ follows directly from (6.8), $\psi(0) \geq 0$ and the existence of $q \in E$:

$$0 = \|Tq - Tq\| \leq \|q - q\| - \psi(\|q - q\|) = -\psi(0) \leq 0.$$

Furthermore, we have given a rate of convergence in the sense of Definition 6.2.6. Such a rate of convergence is typically simpler as we do not have to prove $\mu_n \leq f(n)$ and $\lim_{t\to\infty} f(t) = 0$ for some function $f : \mathbb{R}^+ \to \mathbb{R}^+$. However, Proposition 6.4.11 provides a translation between our rate of convergence into the traditional one. For example, the inequality given in our case matches up well with that in [3, Theorem 3.1]:

$$\|x_n - q\| \leq \Psi^{-1}(\Psi(\|x_0 - q\|) - n + 1).$$

This similarity for this simple case suggests that our abstract quantitative result provides good rates of convergence.

6.5 Case study: quasi asymptotically ψ-weakly contractive maps

Motivation 6.5.1. In this case study we consider four theorems of [2, Section 4] by Alber, Chidume and Zegeye and two theorems of [3, Section 3] by Alber and Guerre-Delabriere. In all these theorems a special case of an asymptotically ψ-weakly contractive map is given, and it is proven that under certain conditions a variant of the Krasnoselski-Mann iteration converges to a fixed point of this map. In the proof of these theorems Lemma 6.4.6 or Lemma 6.4.8 were used.

The following theorem is a quantitative and generalised version of all these theorems, and after proving this theorem, we formulate and prove a quantitative version for each of these theorems.

Theorem 6.5.2. Let X be a normed space, $E \subseteq X$ be a subset, $(A_n : E \to X)_n$ be a sequence of maps, $q \in X$ and $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be some non-decreasing function which

is positive on $\mathbb{R}^+$. Suppose that $(k_n)_n$ is some sequence in $\mathbb{R}_0^+$ and $\sigma : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{N}$ is a modulus such that $(A_n)_n$ is quasi asymptotically weakly contractive in the following sense:

$$\forall_{x\in E}\forall_{b,\delta>0}\forall_{n\geq\sigma(\delta,b)}.\|x-q\| \leq b \Rightarrow \|A_n x - q\| \leq (1+k_n)\|x-q\| - \psi(\|x-q\|) + \delta$$

Furthermore, let $\alpha > 0$ and $(\alpha_n)_n \in (0,\alpha]^{\mathbb{N}}$ be given such that $\sum_{i=0}^{\infty} \alpha_i = \infty$ with rate of divergence r, and let $(x_n)_n \in E^{\mathbb{N}}$ be a Krasnoselski-Mann sequence w.r.t. $(\alpha_n)_n$ and $(A_n)_n$. Finally, let $d > 0$ be given such that $\prod_{i=0}^{n}(1+\alpha_n k_n) \leq d$ for all $n \in \mathbb{N}$. Then:

1. If σ is independent of the second argument (and we consider σ as unary map), $(\|x_n - q\|)_n$ converges to 0 with rate

$$\Phi_{\psi,(\|x_n-q\|)_n,\alpha,d,r,\sigma}(\varepsilon) := r\left(M_{\psi,\alpha,d}(\varepsilon), 2d\int_{\frac{\varepsilon}{2}}^{\|x_{M_{\psi,\alpha,d}(\varepsilon)}-q\|}\frac{dt}{\psi(t)}\right)+1,$$

   where

$$M_{\psi,\alpha,d}(\varepsilon) := \sigma\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon/2)}{d}, \frac{\varepsilon}{\alpha}\right\}\right).$$

2. If there is $c > 0$ with $\forall_{n\in\mathbb{N}}\|x_n - q\| \leq c$, $(\|x_n - q\|)_n$ converges to 0 with rate

$$\Phi_{\psi,c,\alpha,d,r,\sigma}(\varepsilon) := r\left(\sigma\left(\frac{1}{2}\min\left\{\frac{\psi(\varepsilon/2)}{d}, \frac{\varepsilon}{\alpha}\right\}, c\right), 2d\int_{\frac{\varepsilon}{2}}^{c}\frac{dt}{\psi(t)}\right)+1.$$

Proof. For $\delta > 0$ we have

$$\begin{aligned}\|x_{n+1} - q\| &\leq (1-\alpha_n)\|x_n - q\| + \alpha_n\|A_n x_n - q\| \\ &\leq (1-\alpha_n k_n)\|x_n - q\| - \alpha_n(\psi(\|x_n - q\|) - \delta)\end{aligned}$$

in the first case for all $n \geq \sigma(\delta)$ and in the second case for all $n \geq \sigma(\delta, c)$. In the first case we apply Lemma 6.4.8 where $\mu_n := (\|x_n - q\|)_n$, $\beta_n := \alpha_n k_n$ and $N := \sigma$. In the second case we apply Lemma 6.4.8, where $\mu_n := (\|x_n - q\|)_n$, $\beta_n := \alpha_n k_n$ and $N := \sigma(., c)$, together with Remark 6.4.9. □

6.5.1 Totally asymptotically weakly contractive maps

Motivation 6.5.3. As a first application of Theorem 6.5.2 we give a quantitative version of four theorems in [2, Section 4], which consider totally asymptotically weakly contractive maps. The first two corollaries uses the second part of Theorem 6.5.2. In these cases we have provided some bound of the sequence $(\|x_n - q\|)_n$. The third and fourth corollary use the first part of Theorem 6.5.2.

Corollary 6.5.4 (Quantitative version of Theorem 4.1 in [2]). Let a normed space X, a subset $E \subseteq X$ and a totally asymptotically weakly contractive map $T : E \to E$ be given. In particular, there are non-decreasing functions $\psi, \phi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ both positive on $\mathbb{R}^+$ together with two sequences $(k_n)_n, (l_n)_n$ of non-negative reals such that $k_n, l_n \to 0$ for $n \to \infty$ with rate f and g, respectively, and for all $x, y \in E$ and $n \in \mathbb{N}$ we have

$$\|T^n x - T^n y\| \leq \|x - y\| + k_n \phi(\|x - y\|) - \psi(\|x - y\|) + l_n.$$

Suppose that there is a fixed point $q \in E$ of T and that $(x_n)_n$ is a Krasnoselski-Mann sequence in E w.r.t. $(T^n)_n$ and $(\alpha_n)_n$, where $(\alpha_n)_n$ is some sequence in $(0, \alpha]$ such that $\sum_{n=0}^{\infty} \alpha_n = \infty$ with rate r.

Finally, suppose that there exist bounds $d_1, d_2 > 0$ for $(k_n)_n$ and $(l_n)_n$, respectively, and the equation $\phi(\mu) = d_1\psi(\mu) + d_2$ has a unique root μ^*.

Then for any $c > 0$ with $\max\{\|x_0 - q\|, \mu^* + d_1\phi(\mu^*) + d_2\} \leq c$ we have $\|x_n - q\| \to 0$ with rate

$$\Phi_{\psi,\phi,f,g,\alpha,r,c}(\varepsilon) := r\left(\sigma_{\phi,f,g}\left(\frac{1}{2}\min\left\{\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}, c\right), 2\int_{\frac{\varepsilon}{2}}^{c} \frac{dt}{\psi(t)}\right) + 1,$$

where

$$\sigma_{\phi,f,g}(\delta, b) := \max\left\{f\left(\frac{\delta}{2\phi(b)}\right), g\left(\frac{\delta}{2}\right)\right\}.$$

Proof. By Lemma 6.3.11, the sequence $(T^n)_n$ is asymptotically ψ-weakly contractive with modulus $\sigma_{\phi,f,g}$. It is proven in [2, Lemma 3.4 and Theorem 4.1] that the existence of a root μ^* implies that $\|x_n - q\| \leq c$ for any $c \geq \max\{\|x_0 - q\|, \mu^* + d_1\phi(\mu^*) + d_2\}$ and therefore the second part of Theorem 6.5.2 applies directly. □

Lemma 6.5.5. Let $(\lambda_n)_n$, $(\kappa_n)_n$ and $(\gamma_n)_n$ be sequences of non-negative real numbers such that $\sum_{n=0}^{\infty} \kappa_n$ is bounded by $K_1 > 0$ and $\sum_{n=0}^{\infty} \gamma_n$ is bounded by $K_2 > 0$ and for all $n \in \mathbb{N}$

$$\lambda_{n+1} \leq (1 + \kappa_n)\lambda_n + \gamma_n.$$

Then $(\lambda_n)_n$ is bounded by $e^{K_1}(\lambda_0 + K_2)$.

Proof. By induction, we have

$$\lambda_{n+1} \leq \prod_{i=0}^{n}(1+\kappa_i)\left(\lambda_0 + \sum_{i=0}^{n}\gamma_i\right)$$

and

$$\prod_{i=0}^{n}(1+\kappa_i) \leq e^{\sum_{i=0}^{n}\kappa_i}.$$

Putting things together gives the desired bound. □

Remark 6.5.6. The lemma above is inspired by [2, Lemma 3.1] as that lemma is used in the proof of [2, Theorem 4.4]. However, [2, Lemma 3.1] says that $(\lambda_n)_n$ even converges. But it is not possible to compute a rate of convergence, as we have already discussed in Section 2.2.3. However, only boundedness is really used in that proof.

Corollary 6.5.7 (Quantitative version of Theorem 4.4 in [2]). Let a normed space X, a subset $E \subseteq X$ and a totally asymptotically weakly contractive map $T : E \to E$ be given. In particular, there are non-decreasing functions $\psi, \phi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ both positive on $\mathbb{R}^+$ together with two sequences $(k_n)_n, (l_n)_n$ of non-negative reals such that $k_n \to 0$ with rate f and $l_n \to 0$ with rate g, and for all $x, y \in E$ and $n \in \mathbb{N}$ we have

$$\|T^n x - T^n y\| \leq \|x - y\| + k_n\phi(\|x - y\|) - \psi(\|x - y\|) + l_n.$$

Suppose that there is a fixed point $q \in E$ of T and that $(x_n)_n$ is a sequence in E satisfying

$$x_{n+1} = (1 - \alpha_n)x_n + \alpha_n T^n x_n$$

for all $n \in \mathbb{N}$, where $(\alpha_n)_n$ is some sequence in $(0, \alpha]$ such that $\sum_{n=0}^{\infty}\alpha_n = \infty$ with rate r.

Let $\sum_{n=1}^{\infty}\alpha_n k_n$ and $\sum_{n=1}^{\infty}\alpha_n l_n$ be bounded by K and L, respectively, and we assume that there exist $M, M_0 \in \mathbb{R}^+$ with $\forall_{\lambda \geq M}\phi(\lambda) \leq m^{-1}\psi(\lambda) + M_0\lambda$, for some $m > 0$ with $m \geq \sup\{k_n \mid n \in \mathbb{N}\}$. Then $\|x_n - q\| \to 0$ for $n \to \infty$ with rate

$$\Phi_{K,L,M_0,M,x_0,q,\psi,\phi,\alpha,}(\varepsilon) :=$$
$$r\left(\sigma_{\phi,f,g}\left(\frac{1}{2}\min\left\{\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}, C_{K,L,M_0,M,x_0,q,\phi}\right), 2\int_{\frac{\varepsilon}{2}}^{C_{K,L,M_0,M,x_0,q,\phi}}\frac{dt}{\psi(t)}\right) + 1$$

where

$$C_{K,L,M_0,M,x_0,q,\phi} := e^{M_0 L}(\|x_0 - q\| + \phi(M)K + L)$$

and

$$\sigma_{\phi,f,g}(\delta, b) := \max\left\{ f\left(\frac{\delta}{2\phi(b)}\right), g\left(\frac{\delta}{2}\right)\right\}.$$

Proof. Since $\phi(\lambda) \leq m^{-1}\psi(\lambda) + M_0\lambda$ for all $\lambda \geq M$, we have

$$\begin{aligned} k_n\phi(\lambda) - \psi(\lambda) &\leq k_n(m^{-1}\psi(\lambda) + M_0\lambda + \phi(M)) - \psi(\lambda) \\ &\leq k_n\phi(M) + M_0 k_n\lambda \end{aligned}$$

for all $\lambda \in \mathbb{R}$ and $n \in \mathbb{N}$. It follows

$$\begin{aligned} \|x_{n+1} - q\| &\leq (1-\alpha_n)\|x_n - q\| + \alpha_n\|T^n x_n - T^n q\| \\ &\leq (1-\alpha_n)\|x_n - q\| + \\ &\quad \alpha_n\left(\|x_n - q\| + k_n\phi(\|x_n - q\|) - \psi(\|x_n - q\|) + l_n\right) \\ &\leq \|x_n - q\| + \alpha_n\left(k_n\phi(\|x_n - q\|) - \psi(\|x_n - q\|)\right) + \alpha_n l_n \\ &\leq \|x_n - q\| + \alpha_n(k_n\phi(M) - M_0 k_n\|x_n - q\|) + \alpha_n l_n \\ &\leq (1 + M_0\alpha_n k_n)\|x_n - q\| + \alpha_n k_n\phi(M) + \alpha_n l_n. \end{aligned}$$

Using Lemma 6.5.5, $(\|x_n - q\|)_n$ is bounded by $C_{K,L,M_0,M,x_0,q,\phi}$. By Lemma 6.3.11, $(T^n)_n$ is asymptotically weakly contractive with modulus $\sigma_{\phi,f,g}$. Hence, the second part of Theorem 6.5.2 gives the rate of convergence as above. □

Corollary 6.5.8 (Quantitative version of Theorem 4.2 in [2]). Let a normed space X, a subset $E \subseteq X$ and a totally asymptotically weakly contractive map T be given. In particular, there are non-decreasing functions $\psi, \phi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ both positive on $\mathbb{R}^+$ together with two sequences $(k_n)_n, (l_n)_n$ of non-negative reals such that $k_n, l_n \to 0$ for $n \to \infty$ with rate f and g, respectively, and for all $x, y \in E$ and $n \in \mathbb{N}$ we have

$$\|T^n x - T^n y\| \leq \|x - y\| + k_n\phi(\|x - y\|) - \psi(\|x - y\|) + l_n.$$

Suppose that there is a fixed point $q \in E$ of T, and let a Krasnoselski-Mann sequence $(x_n)_n$ in E w.r.t $(T^n)_n$ and $(\alpha_n)_n$ be given, where $(\alpha_n)_n$ is some sequence in $(0, \alpha]$ such that $\sum_{n=0}^{\infty}\alpha_n = \infty$ with rate r.

Suppose that $(k_n)_n$ is bounded by some $k \in [0,1)$, and there exists $M > 0$ such that $\phi(\lambda) \leq \psi(\lambda)$ for all $\lambda \geq M$. Then $\|x_n - q\| \to 0$ for $n \to \infty$ with rate

$$\Phi_{\psi,\phi,f,g,\alpha,r,k,(x_n)_n,q}(\varepsilon) := r\left(\sigma_{f,g,h,\psi}(\delta_{\alpha,\psi,k}(\varepsilon)), \frac{2}{1-k}\int_{\frac{\varepsilon}{2}}^{\left\|x_{\sigma_{f,g,h,\psi}(\delta_{\alpha,\psi,k}(\varepsilon))}-q\right\|} \frac{dt}{\psi(t)}\right) + 1,$$

where

$$\sigma_{f,g,\phi}(\delta) := \max\left\{f\left(\frac{\delta}{2\phi(M)}\right), g\left(\frac{\delta}{2}\right)\right\}$$

and

$$\delta_{\alpha,\psi,k}(\varepsilon) := \frac{1}{2}\min\left\{(1-k)\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}.$$

Proof. As $\phi(\lambda) \leq \psi(\lambda)$ for all $\lambda \geq M$, it follows

$$\phi(\lambda) \leq \phi(M) + \psi(\lambda)$$

for all $\lambda \in \mathbb{R}_0^+$. Therefore, for all $x \in E$, $\delta > 0$ and $n \geq \sigma_{f,g,\phi}(\delta)$ we have

$$\begin{aligned}\|T^n x - q\| &\leq \|x-q\| + k_n(\phi(M) + \psi(\|x-q\|)) - \psi(\|x-q\|) + l_n \\ &\leq \|x-q\| - (1-k)\psi(\|x-q\|) + k_n\phi(M) + l_n \\ &\leq \|x-q\| - (1-k)\psi(\|x-q\|) + \delta.\end{aligned}$$

Hence, T^n is quasi asymptotically $(1-k)\psi$-weakly contractive with unary modulus $\sigma_{f,g,\phi}$. By the first part of Theorem 6.5.2, $\|x_n - q\| \to 0$ for $n \to \infty$ with modulus $\Phi_{\psi,\phi,f,g,\alpha,r,k,(x_n)_n,q}$ □

Corollary 6.5.9 (Quantitative version of Theorem 4.3 in [2]). Let a normed space X, a subset $E \subseteq X$ and a totally asymptotically weakly contractive map $T : E \to E$ be given. In particular, there are non-decreasing functions $\psi, \phi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ both positive an $\mathbb{R}^+$ together with two sequences $(k_n)_n, (l_n)_n$ of non-negative reals such that $k_n, l_n \to 0$ for $n \to \infty$ with rate f and g, respectively, and for all $x, y \in E$ and $n \in \mathbb{N}$ we have

$$\|T^n x - T^n y\| \leq \|x-y\| + k_n\phi(\|x-y\|) - \psi(\|x-y\|) + l_n.$$

Suppose that there is a fixed point $q \in E$ of T, and let a Krasnoselski-Mann sequence $(x_n)_n$ in E w.r.t $(T^n)_n$ and $(\alpha_n)_n$ be given, where $(\alpha_n)_n$ is some sequence in $(0, \alpha]$ such that $\sum_{n=0}^{\infty} \alpha_n = \infty$ with rate r.

Finally we assume that there exist $M, M_0 \in \mathbb{R}^+$ with $\forall_{\lambda \geq M} \phi(\lambda) \leq M_0 \lambda$, and that $\sum_{i=0}^{\infty} \alpha_n k_n$ is bounded by $K \geq 0$. Then $\|x_n - q\| \to 0$ for $n \to \infty$ with rate

$$\Phi_{f,g,M,M_0,K,\phi,\psi,\alpha,(x_n)_n,q}(\varepsilon) := r\left(\sigma_{f,g,h,M,\phi}\left(\delta_{M_0,K,\psi,\alpha}(\varepsilon)\right), 2e^{M_0 K} \int_{\frac{\varepsilon}{2}}^{\left\|x_{\sigma_{f,g,h,M,\phi}(\delta_{M_0,K,\psi,\alpha}(\varepsilon))} - q\right\|} \frac{dt}{\psi(t)}\right) + 1,$$

where

$$\sigma_{f,g,h,M,\phi}(\delta) := \max\left\{f\left(\frac{\delta}{2\phi(M)}\right), g\left(\frac{\delta}{2}\right)\right\}$$

and

$$\delta_{M_0,K,\psi,\alpha}(\varepsilon) := \frac{1}{2}\min\left\{e^{-M_0 K}\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}.$$

Proof. As $\phi(\lambda) \leq M_0 \lambda$ for all $\lambda \geq M_0$ and ϕ is non-decreasing, it follows

$$\phi(\lambda) \leq M_0 \lambda + \phi(M)$$

for all $\lambda \in \mathbb{R}_0^+$. Therefore, for all $x \in E$, $\delta > 0$ and $n \geq \sigma_{f,g,h,M,\phi}(\delta)$ we have

$$\begin{aligned}\|T^n x - q\| &\leq \|x - q\| + k_n \phi(\|x - q\|) - \psi(\|x - q\|) + l_n \\ &\leq (1 + M_0 k_n)\|x - q\| - \psi(\|x - q\|) + k_n \phi(M) + l_n \\ &\leq (1 + M_0 k_n)\|x - q\| - \psi(\|x - q\|) + \delta.\end{aligned}$$

Hence, $(T^n)_n$ is quasi asymptotically ψ-weakly contractive w.r.t. q and $(M_0 k_n)_n$ and unary modulus $\sigma_{f,g,M,\phi}$. Furthermore is

$$\prod_{i=0}^{n}(1 + M_0 k_i) \leq e^{\sum_{i=0}^{n} \ln(1 + M_0 k_i)} \leq e^{\sum_{i=0}^{n} M_0 k_i} \leq e^{M_0 K}$$

for all n. By the first part of Theorem 6.5.2 it follows $\|x_n - q\| \to 0$ for $n \to \infty$ with modulus $\Phi_{f,g,M,M_0,K,\phi,\psi,\alpha,(x_n)_n,q}$. □

6.5.2 Approximate weakly contractive maps

Motivation 6.5.10. The next two corollaries are a computational version of two theorems in [3, Section 3], which consider sequences $(A_n)_n$ of operators which are weakly contracive in the limit. It turns out that those theorems are simple applications of Theorem 6.5.2.

Corollary 6.5.11 (Quantitative version of Theorem 3.4 of [3]). Let X be a normed space, $E \subseteq X$ be a subset and $A : E \to E$ be a map with a fixed point $q \in E$. Suppose that there are sequences of positive numbers $(h_n)_n, (\delta_n)_n, (\mu_n)_n$ and $(\nu_n)_n$ converging to 0 with rates f_1, f_2, f_3 and f_4 respectively, another sequence $(\beta_n)_n$ of positive numbers with $\sum_{i=0}^{\infty} \beta_i$ bounded by K, a function $g : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, a sequence of functions $(\psi_n : \mathbb{R}_0^+ \to \mathbb{R}_0^+)_n$, a non-decreasing function $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ positive on $\mathbb{R}^+$ and a sequence $(A_n : E \to E)_n$ of maps such that for all $x, y \in E$, $n \in \mathbb{N}$ and $t \geq 0$,

$$\begin{aligned} \|A_n x - A_n y\| &\leq (1+\beta_n)\|x-y\| - \psi_n(\|x-y\|) + \mu_n, \\ \|A_n x - Ax\| &\leq h_n g(\|x\|) + \delta_n \\ |\psi_n(t) - \psi(t)| &\leq \nu_n. \end{aligned}$$

Finally, let $(x_n)_n$ be a sequence in E such that $x_{n+1} = A_n x_n$ for all $n \in \mathbb{N}$ and we assume that $\sum_{n=0}^{\infty} \beta_n$ is bounded by $K > 0$. Then $\|x_n - q\|$ converges to 0 for $n \to \infty$ with rate

$$\begin{aligned} &\Phi_{\psi,f_1,f_2,f_3,f_4,g,K,(x_n)_n,q}(\varepsilon) := \\ &\sigma_{f_1,f_2,f_3,f_4,g,q}(\delta_{\psi,K}(\varepsilon)) + \left\lceil 2e^K \int_{\frac{\varepsilon}{2}}^{\left\|x_{\sigma_{f_1,f_2,f_3,f_4,g,q}(\delta_{\psi,K}(\varepsilon))} - q\right\|} \frac{dt}{\psi(t)} \right\rceil + 1, \end{aligned}$$

where

$$\sigma_{f_1,f_2,f_3,f_4,g,q}(\delta) := \max\left\{ f_1\left(\frac{\delta}{4g(\|q\|)}\right), f_2\left(\frac{\delta}{4}\right), f_3\left(\frac{\delta}{4}\right), f_4\left(\frac{\delta}{4}\right)\right\}$$

and

$$\delta_{\psi,K}(\varepsilon) := \frac{1}{2}\min\left\{ e^{-K}\psi\left(\frac{\varepsilon}{2}\right), \varepsilon\right\}.$$

Proof. Using the three inequalities we have

$$\begin{aligned} \|A_n x - q\| &\leq \|A_n x - A_n q\| + \|A_n q - q\| \\ &\leq (1+\beta_n)\|x-q\| - \psi_n(\|x-q\|) + \mu_n + h_n g(\|q\|) + \delta_n \\ &\leq (1+\beta_n)\|x-q\| - \psi(\|x-q\|) + \nu_n + \mu_n + h_n g(\|q\|) + \delta_n \\ &\leq (1+\beta_n)\|x-q\| - \psi(\|x-q\|) + \delta \end{aligned}$$

for all $x \in E$, $\delta > 0$ and $n \geq \sigma_{f_1,f_2,f_3,f_4,g,q}(\delta)$. Therefore, $(A_n)_n$ is quasi asymptotically ψ-weakly contractive w.r.t. q and $(\beta_n)_n$ and unary modulus $\sigma_{f_1,f_2,f_3,f_4,g,q}$. Furthermore, $(x_n)_n$ is a Krasnoselski-Mann sequence w.r.t. $(1)_n$ and $(A_n)_n$ and a

rate of divergence for $\sum_{i=0}^{\infty} 1$ is given by $r(n,x) := n + \lceil x \rceil$. Finally, a bound of $\prod_{n=0}^{\infty}(1+\beta_n)$ is given by e^K (proven similar as in the proof of the last corollary). Hence, we apply Theorem 6.5.2 and get that $\|x_n - q\| \to 0$ for $n \to \infty$ with rate $\Phi_{\psi,f_1,f_2,f_3,f_4,g,K,(x_n)_n,q}$. □

Corollary 6.5.12 (Quantitative version of Theorem 3.6 of [3]). Let X be a normed space, $E \subseteq X$ be a subset and $A : E \to E$ be a weakly contractive map, i.e. there is a non-decreasing map $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ which is positive on $\mathbb{R}^+$ such that

$$\forall_{x,y\in E}\|Ax - Ay\| \leq \|x - y\| - \psi(\|x - y\|).$$

Furthermore, let $q \in E$ be a fixed point of A, $(A_n : E \to E)_n$ be a sequence of maps and $(x_n)_n$ be a sequence in E with $x_{n+1} = A_n x_n$ for all $n \in \mathbb{N}$. We assume that there exists a non-decreasing map $g : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ and two sequences $(\delta_n)_n$ and $(h_n)_n$, which converge to 0 with moduli f_1 and f_2, repetitively, such that

$$\|A_n x - Ax\| \leq h_n g(\|x\|) + \delta_n.$$

Then:

1. If the sequence $(\|x_n - q\|)_n$ is bounded by some $c > 0$, it converges to 0 with rate

$$\Phi_{\psi,c,f_1,f_2,g,q}(\varepsilon) := \sigma_{f_1,f_2,g,q}\left(\frac{1}{2}\min\left\{\psi\left(\frac{\varepsilon}{2}\right), \frac{\varepsilon}{\alpha}\right\}, c\right) + \left\lceil 2\int_{\varepsilon/2}^{c} \frac{dt}{\psi(t)}\right\rceil + 1,$$

   where

$$\sigma_{f_1,f_2,g,q}(\delta, b) := \max\left\{f_1\left(\frac{\delta}{2}\right), f_2\left(\frac{\delta}{2g(b + \|q\|)}\right)\right\}.$$

2. If $h_n = 0$ for all n, the sequence $(\|x_n - q\|)_n$ converges to 0 with rate

$$\Phi_{\psi,(x_n)_n,q,f_1}(\varepsilon) := f_1(\delta_\psi(\varepsilon)) + \left\lceil 2\int_{\varepsilon/2}^{\|x_{f_1(\delta_\psi(\varepsilon))} - q\|} \frac{dt}{\psi(t)}\right\rceil + 1,$$

   where

$$\delta_\psi(\varepsilon) := \frac{1}{2}\min\left\{\psi\left(\frac{\varepsilon}{2}\right), \varepsilon\right\}.$$

Proof. First we observe that $(x_n)_n$ is a Krasnoselski-Mann sequence w.r.t. $(1)_n$ and $(A_n)_n$ and a rate of divergence for $\sum_{i=0}^{\infty} 1$ is given by $r(n,x) := n+\lceil x \rceil$. Furthermore, for all $n \in \mathbb{N}$ and $x \in E$ we have

$$\begin{aligned}\|A_n x - q\| &\leq \|A_n x - Ax\| + \|Ax - Aq\| \\ &\leq h_n g(\|x\|) + \delta_n + \|x - q\| - \psi(\|x - q\|).\end{aligned}$$

We now consider the first case and assume that $(\|x_n - q\|)_n$ is bounded by c. For all $x \in E$, $b, \delta > 0$ and $n \geq \sigma_{f_1,f_2,g,q}(\delta, b)$ with $\|x - q\| \leq b$ we have

$$\begin{aligned}\|A_n x - q\| &\leq h_n g(\|x\|) + \delta_n + \|x - q\| - \psi(\|x - q\|) \\ &\leq \|x - q\| - \psi(\|x - q\|) + h_n g(b + \|q\|) + \delta_n \\ &\leq \|x - q\| - \psi(\|x - q\|) + \delta,\end{aligned}$$

and the first part is proven by the second part of Theorem 6.5.2.

In the second case we have

$$\begin{aligned}\|A_n x - q\| &\leq \delta_n + \|x - q\| - \psi(\|x - q\|) \\ &\leq \|x - q\| - \psi(\|x - q\|) + \delta,\end{aligned}$$

for all $x \in E$, $\delta > 0$ and $n \geq f_1(\delta)$, and the statement follows by the first part of Theorem 6.5.2. □

6.6 Case study: asymptotically d-weakly contractive maps

6.6.1 Duality selection maps

Motivation 6.6.1. The following definition is a generalisation of the normalised duality map in Definition 6.3.14 and goes back to [97].

Definition 6.6.2. Let X be a normed space and $J : X \to X^*$. J is called *duality selection map* if

$$\forall_{x \in X} \langle x, Jx \rangle = \|x\|^2 = \|Jx\|^2.$$

Furthermore, we say that J is *uniformly continuous (on bounded sets)* with modulus $\omega : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ if

$$\forall_{x,y \in E} \forall_{d,\epsilon > 0}. \|x\|, \|y\| \leq d \wedge \|x - y\| \leq \omega(d, e) \Rightarrow \|Jx - Jy\| \leq \varepsilon.$$

Note that $\|\cdot\|$ denotes both the norm in X and the induced norm in X^*.

Lemma 6.6.3 (Lemma 3.5 of [97]). Let X be a normed space and $J : X \to X^*$ be a duality selection map. Then

$$\forall_{x,y\in X}\|x+y\|^2 \leq \|x\|^2 + 2\langle y, J(x+y)\rangle.$$

Proof. This follows from

$$\begin{aligned}\|x+y\|^2 &= \langle x, J(x+y)\rangle + \langle y, J(x+y)\rangle \leq \|x\|\|x+y\| + \langle y, J(x+y)\rangle \\ &\leq \frac{1}{2}(\|x\|^2 + \|x+y\|^2) + \langle y, J(x+y)\rangle.\end{aligned}$$

□

Motivation 6.6.4. Using Definition 6.6.2 we are able to generalise the notion of being asymptotically d-weakly contractive, i.e. Definition 6.3.16.

Definition 6.6.5. Let X be a normed space, $J : X \to X^*$ be a duality selection map, and $E \subseteq X$ be a subset. A sequence $(A_n : E \to X)_n$ of maps is called *asymptotically d-weakly contractive* with modulus $\sigma : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ if there exists some non-decreasing $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ which is positive on $\mathbb{R}^+$ such that for all $x, y \in E$, $b, \delta > 0$ and $n \geq \sigma(\delta, b)$

$$\|x-y\| \leq b \Rightarrow |\langle A_n x - A_n y, J(x-y)\rangle| \leq \|x-y\|^2 - \psi(\|x-y\|) + \delta.$$

We say that $(A_n)_n$ is *quasi asymptotically d-weakly contractive* w.r.t. $q \in X$ if for all $x \in E$, $b, \delta > 0$ and $n \geq \sigma(\delta, b)$

$$\|x-q\| \leq b \Rightarrow |\langle A_n x - q, J(x-q)\rangle| \leq \|x-q\|^2 - \psi(\|x-q\|) + \delta.$$

6.6.2 Convergence theorem in spaces with uniformly continuous selction map

Theorem 6.6.6. Let X be a normed space and J be a uniformly continuous duality selection map with modulus ω. Let $(A_n : E \to X)_n$ be a sequence of maps, $q \in X$ and $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be some non-decreasing function positive on $\mathbb{R}^+$. Suppose that $(A_n)_n$ is quasi asymptotically d-weakly contractive w.r.t. q, and let $(x_n)_n$ be a P-Krasnoselski-Mann sequence w.r.t. $(A_n)_n$ and some $(\alpha_n)_n$, where $P : X \to E$ is a non-expansive retraction and $(\alpha_n)_n$ is a sequence in $(0, \alpha]$ such that $\alpha_n \to 0$ with rate f and $\sum_{i=0}^n \alpha_n$ divergent with rate r. Finally, let $c_1, c_2, c_3 > 0$ be given such

that $\|x_n - q\| \leq c_1$ and $\|A_n x_n - x_n\| \leq c_2$ for all n and $\|q\| \leq c_3$. Then $\|x_n - q\| \to 0$ for $n \to \infty$ with rate

$$\Phi_{\omega,c_1,c_2,c_3,\alpha,f,r,\sigma}(\varepsilon) := r\left(N_{\omega,c_1,c_2,c_3,f,\sigma}\left(\min\left\{\psi\left(\frac{\varepsilon}{\sqrt{2}}\right), \frac{\varepsilon^2}{2\alpha}\right\}\right), 2\int_{\frac{\varepsilon}{\sqrt{2}}}^{c_1} \frac{u\,du}{\psi(u)}\right) + 1,$$

where

$$N_{\omega,c_1,c_2,c_3,f,\sigma}(\delta) := \max\left\{\sigma\left(\frac{\delta}{4}, c_1\right), f\left(\frac{1}{c_2}\omega\left(c_1 + c_3, \frac{\delta}{4c_2}\right)\right)\right\}.$$

Proof. As $(\|x_n - q\|)_n$ is bounded by c_1 we have for all $\delta > 0$ and $n \geq \sigma\left(\frac{\delta}{4}, c_1\right)$:

$$\begin{aligned}\langle A_n x_n - q, J(x_n - q)\rangle &\leq \|x_n - q\|^2 - \psi(\|x_n - q\|) + \frac{\delta}{4} \\ &= \langle x_n - q, J(x_n - q)\rangle - \psi(\|x_n - q\|) + \frac{\delta}{4}\end{aligned}$$

and therefore

$$\langle Ax_n - x_n, J(x_n - q)\rangle \leq -\psi(\|x_n - q\|) + \frac{\delta}{4}. \tag{6.10}$$

Independently of this, we define

$$y_{n+1} := (1 - \alpha_n)x_n - \alpha_n A_n x_n$$

and observe that

$$\|y_{n+1} - x_n\| = \|\alpha_n(A_n x_n - x_n)\| \leq \alpha_n c_2.$$

Hence, since $\|x_n\| \leq \|x_n - q\| + \|q\| \leq c_1 + c_3$, for all $n \geq f\left(\frac{1}{c_2}\omega\left(c_1 + c_3, \frac{\delta}{4c_2}\right)\right)$ it follows

$$\|y_{n+1} - x_n\| \leq \omega\left(c_1 + c_3, \frac{\delta}{4c_2}\right)$$

and therefore

$$\|J(y_{n+1} - q) - J(x_n - q)\| \leq \frac{\delta}{4c_2}.$$

Using this we get

$$\begin{aligned}\langle Ax_x - x_n, J(y_{n+1} - q) - J(x_n - q)\rangle &\leq \|Ax_n - x_n\|\|J(y_{n+1} - q) - J(x_n - q)\| \\ &\leq c_2\|J(y_{n+1} - q) - J(x_n - q)\| \\ &\leq \frac{\delta}{4}\end{aligned} \tag{6.11}$$

for all $n \geq f\left(\frac{1}{c_2}\omega\left(c_1 + c_3, \frac{\delta}{4c_2}\right)\right)$.

Putting (6.10) and (6.11) together leads to

$$\begin{aligned}
&\langle A_n x_n - x_n, J(y_{n+1} - q)\rangle \\
&\quad \leq \langle A_n x_n - x_n, J(x_n - q)\rangle + \langle A_n x_n - x_n, J(y_{n+1} - q) - J(x_n - q)\rangle \\
&\quad \leq -\psi(\|x_n - q\|) + \frac{\delta}{2}
\end{aligned}$$

for all $\delta > 0$ and $n \geq N_{\omega,c_1,c_2,c_3,f,\sigma}(\delta)$. Thus for all these δ and n we have

$$\begin{aligned}
\|x_{n+1} - q\|^2 &= \|P((1-\alpha_n)x_n + \alpha_n A_n x_n) - Pq\|^2 \\
&\leq \|x_n - q + \alpha_n(A_n x_n - x_n)\|^2 \\
&\leq \|x_n - q\|^2 + 2\alpha_n\langle A_n x_n - x_n, J(x_{n+1} - q)\rangle \\
&\leq \|x_n - q\|^2 - 2\alpha_n\psi(\|x_n - q\|) + \alpha_n\delta,
\end{aligned}$$

where the second inequality follows from Lemma 6.6.3. In particular, $\mu_n := \|x_n - q\|^2$ satisfies

$$\mu_{n+1} \leq \mu_n - \alpha_n(\overline{\psi}(\mu_n) + \delta),$$

where $\overline{\psi} := 2\psi(\sqrt{\cdot})$, for all $\delta > 0$ and $n \geq N_{\omega,c_1,c_2,c_3,f,\sigma}(\delta)$. Thus, by Lemma 6.4.6 $\|x_n - q\|^2 \to 0$ for $n \to \infty$ with rate

$$\overline{\Phi}_{\omega,c_1,c_2,c_3,\alpha,f,r,\sigma}(\varepsilon) := r\left(N_{\omega,c_1,c_2,c_3,f,\sigma}\left(\min\left\{\psi\left(\frac{\sqrt{\varepsilon}}{\sqrt{2}}\right), \frac{\varepsilon}{2\alpha}\right\}\right), 2\int_{\frac{\varepsilon}{2}}^{c_1^2} \frac{dt}{2\psi(\sqrt{t})}\right) + 1$$

Therefore, $\|x_n - q\| \to 0$ for $n \to \infty$ with rate

$$\Phi_{\omega,c_1,c_2,c_3,\alpha,f,r,\sigma}(\varepsilon) := r\left(N_{\omega,c_1,c_2,c_3,f,\sigma}\left(\min\left\{\psi\left(\frac{\varepsilon}{\sqrt{2}}\right), \frac{\varepsilon^2}{2\alpha}\right\}\right), \int_{\frac{\varepsilon^2}{2}}^{c_1^2} \frac{dt}{\psi(\sqrt{t})}\right) + 1.$$

By the substitution rule we obtain

$$\int_{\frac{\varepsilon^2}{2}}^{c_1^2} \frac{dt}{\psi(\sqrt{t})} = \int_{\frac{\varepsilon}{\sqrt{2}}}^{c_1} \frac{2udu}{\psi(u)}$$

which leeds to the rate stated in the theorem. □

6.6.3 d-weakly contractive maps in uniformly smooth spaces

Motivation 6.6.7. As an application of Theorem 6.6.6 we consider d-weakly contractive maps (as in Motivation 6.3.15) in uniformly smooth spaces (defined below). It turns out that uniformly smooth spaces are special cases of normed spaces with uniformly continuous duality selection map (see Lemma 6.6.9). Furthermore, T being d-weakly contracive implies that T is quasi asymptotically d-weakly contractive w.r.t. any point q and modulus $\sigma(\delta) = 0$. In fact, we get a simple corollary, which forms a computational version of Theorem 3.1 of [37].

Definition 6.6.8. Let X be a Banach space. X is called *uniformly smooth* if

$$\forall_{\varepsilon>0}\exists_{\delta>0}\forall_{x,y\in X}.\|x\| = 1 \wedge \|y\| \leq \delta \Rightarrow \|x+y\| + \|x-y\| \leq 2 + \varepsilon\|y\|$$

A function $\tau : \mathbb{R}^+ \to \mathbb{R}^+$ is called *modulus of uniform smoothness* if

$$\forall_{\varepsilon>0}\forall_{x,y\in X}.\|x\| = 1 \wedge \|y\| \leq \tau(\varepsilon) \Rightarrow \|x+y\| + \|x-y\| \leq 2 + \varepsilon\|y\|.$$

Lemma 6.6.9. Let X be a uniformly smooth space with modulus τ. Then the normalised duality map $J : X \to 2^{X^*}$ from Definition 6.3.14 is single-valued and we consider J as a map from X to X^*. Furthermore, J is uniformly continuous on bounded sets with modulus $\omega_\tau : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ given by

$$\omega_\tau(d,\varepsilon) := \begin{cases} \frac{\varepsilon^2}{12d}\tau\left(\frac{\varepsilon}{2d}\right) & \text{if} \quad \varepsilon \in (0,2], d \geq 1, \\ \frac{\varepsilon^2}{12}\tau\left(\frac{\varepsilon}{2}\right) & \text{if} \quad \varepsilon \in (0,2], d < 1, \\ \frac{1}{3d}\tau\left(\frac{1}{d}\right) & \text{if} \quad \varepsilon > 2, d \geq 1, \\ \frac{1}{3}\tau\left(1\right) & \text{if} \quad \varepsilon > 2, d < 1. \end{cases}$$

Proof. By [64, Theorem 1] uniformly smoothness of X implies uniform convexity of X^*, and by [59, Proposition 12.3b] uniform convexity of X^* implies that J is single-valued. That ω_τ is a modulus of uniform continuity follows from [97, Proposition 2.5]. □

Corollary 6.6.10 (Quantitative version of Theorem 3.1 of [37]). Let X be a uniformly smooth space with modulus τ, $E \subseteq X$ be a subset of X and $T : E \to X$ be a d-weakly contractive map, i.e. there is a non-decreasing $\psi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ positive on $\mathbb{R}^+$ such that

$$\forall_{x,y\in E}|\langle Tx - Ty, J(x-y)\rangle| \leq \|x-y\|^2 - \psi(\|x-y\|^2). \tag{6.12}$$

We suppose that $(x_n)_n$ is a sequence in X satisfying

$$x_{n+1} = P((1-\alpha_n)x_n + \alpha_n T x_n) \tag{6.13}$$

where $P : X \to E$ is some non-expansive retraction and $(\alpha_n)_n$ is some sequence in $(0, \alpha]$ such that $\alpha_n \to 0$ with rate of convergence f and $\sum_{n=0}^{\infty} \alpha_n = \infty$ with rate r. Then whenever $c_1, c_2, c_3 > 0$ are such that $\|x_n - q\| \leq c_1$, $\|Tx_n - x_n\| \leq c_2$ for all $n \in \mathbb{N}$ and $\|q\| \leq c_3$, we have $\|x_n - q\| \to 0$ with rate

$$\Phi_{\tau,c_1,c_2,c_3,\alpha,f,r}(\varepsilon) := r\left(N_{\tau,c_1,c_2,c_3,f}\left(\min\left\{\psi\left(\frac{\varepsilon}{\sqrt{2}}\right), \frac{\varepsilon^2}{2\alpha}\right\}\right), 2\int_{\frac{\varepsilon}{\sqrt{2}}}^{c_1} \frac{u\,du}{\psi(u)}\right) + 1.$$

where

$$N_{\tau,c_1,c_2,c_3,f}(\delta) := f\left(\frac{1}{c_2}\omega_\tau\left(c_1 + c_3, \frac{\delta}{4c_2}\right)\right)$$

and ω_τ is given as in Lemma 6.6.9.

Proof. By Lemma 6.6.9, ω_τ is a modulus of uniform continuity of the normalised duality map J. Furthermore, because of (6.12) the constant sequence $(T)_n$ is asymptotically d-weakly contractive with modulus $\sigma(\delta) = 0$, and by (6.13) the sequence $(x_n)_n$ is a P-Krasnoselski-Mann sequence w.r.t. $(T)_n$ and $(\alpha_n)_n$. Hence, Theorem 6.6.6 leads to the desired rate of convergence. □

Remark 6.6.11. In the corollary above we have assumed that the bounds c_1 and c_2 exists, whereas in Theorem 3.1 of [37] it is proven that these bounds exists, e.g. it is shown that c_1 can be replaced by $\|x_1 - q\|$. It is more complex (if it is even possible) to construct the bound c_2 out of the proof in [37] and could be the topic of a new work. However, for our purpose it sufficient to assume that these bounds exist.

Appendix A

Haskell program

```
{-# LANGUAGE FlexibleInstances, TupleSections  #-}

-- The following part is from 'Konstruktive Analysis mit
-- exakten reellen Zahlen':

-- Authors: Quirin F. Schroll and Franziskus Wiesnet

import Data.Ratio

-- | For elegant display of infinite objects
display :: Show a => Int -> a -> IO ()
display n = putStrLn . take n . show

data Sd = SdL | SdM | SdR
  deriving (Eq, Ord)

instance Enum Sd where
  fromEnum SdL = (-1)
  fromEnum SdM =   0
  fromEnum SdR =   1

  toEnum (-1) = SdL
  toEnum   0  = SdM
  toEnum   1  = SdR
  toEnum   n  = error $ "cannot cast " ++ show n ++ " to enum"
```

```
instance Show Sd where
  show SdL = "-1"
  show SdM = "0"
  show SdR = "+1"

infixr 5 :~:
data Str = Sd :~: Str

instance Show Str where
  show (x :~: xs) = show x ++ " " ++ show xs

strCoRec :: t -> (t -> (Sd, Either Str t)) -> Str
strCoRec t f = let (d, strt) = f t in d :~: case strt of
  Left str -> str
  Right t0 -> strCoRec t0 f

mOne  = SdL :~: mOne
zero  = SdM :~: zero
pOne  = SdR :~: pOne
phalf = SdR :~: zero
mhalf = SdL :~: zero
third = SdM :~: SdR :~: third
sixth = SdM :~: third
ninth = SdM :~: SdM :~: SdM :~: SdR :~: SdR :~: SdR :~: ninth
twoThird = SdR :~: third

cCoINegToCoIPlusOne, cCoIPosToCoIMinusOne :: Str -> Str
cCoINegToCoIPlusOne (SdR :~: u) = pOne
cCoINegToCoIPlusOne (SdM :~: u) = SdR :~: cCoINegToCoIPlusOne u
cCoINegToCoIPlusOne (SdL :~: u) = SdR :~: u

cCoIPosToCoIMinusOne (SdR :~: u) = SdL :~: u
cCoIPosToCoIMinusOne (SdM :~: u) = SdL :~: cCoIPosToCoIMinusOne u
cCoIPosToCoIMinusOne (SdL :~: u) = mOne

cCoIToCoIDouble :: Str -> Str
cCoIToCoIDouble (SdR :~: u) = cCoINegToCoIPlusOne u
cCoIToCoIDouble (SdM :~: u) = u
```

```
cCoIToCoIDouble (SdL :~: u) = cCoIPosToCoIMinusOne u

cCoIToCoIQuad :: Str -> Str
cCoIToCoIQuad = cCoIToCoIDouble . cCoIToCoIDouble

k n | n >  2   =  1
    | n < -2   = -1
    | otherwise = 0
j   6  =  2
j   5  =  1
j   4  =  0
j   3  = -1
j   2  =  2
j   1  =  1
j   0  =  0
j (-1) = -1
j (-2) = -2
j (-3) =  1
j (-4) =  0
j (-5) = -1
j (-6) = -2

cCoIAverage :: Str -> Str -> Str
cCoIAverage (d :~: u) (e :~: v) =
  strCoRec (fromEnum d + fromEnum e, u, v) step where
  step :: (Int, Str, Str) -> (Sd, Either Str (Int, Str, Str))
  step (t, d :~: u, e :~: v) =
    (toEnum (k num), Right (j num, u, v)) where
      num = fromEnum d + fromEnum e + 2*t

cCoIUMinus :: Str -> Str
cCoIUMinus (SdL :~: s) = SdR :~: cCoIUMinus s
cCoIUMinus (SdM :~: s) = SdM :~: cCoIUMinus s
cCoIUMinus (SdR :~: s) = SdL :~: cCoIUMinus s

cCoIDivSatCoIClAux1 u0 u1 =
  cCoIToCoIQuad $ cCoIAverage u0 (SdM :~: cCoIUMinus u1)
cCoIDivSatCoIClAux4 u0 u1 =
```

```
  cCoIToCoIQuad $ cCoIAverage u0 (SdM :~:            u1)


cCoIDiv u@(SdR :~:                 _) v =
  SdR :~: cCoIDiv (cCoIDivSatCoIClAux1 u v) v
cCoIDiv u@(SdM :~: SdR :~:         _) v =
  SdR :~: cCoIDiv (cCoIDivSatCoIClAux1 u v) v
cCoIDiv u@(SdM :~: SdM :~: SdR :~: _) v =
  SdR :~: cCoIDiv (cCoIDivSatCoIClAux1 u v) v
cCoIDiv u@(SdM :~: SdM :~: SdM :~: _) v =
  SdM :~: cCoIDiv (cCoIToCoIDouble u)       v
cCoIDiv u@(SdL :~:                 _) v =
  SdL :~: cCoIDiv (cCoIDivSatCoIClAux4 u v) v
cCoIDiv u@(SdM :~: SdL :~:         _) v =
  SdL :~: cCoIDiv (cCoIDivSatCoIClAux4 u v) v
cCoIDiv u@(SdM :~: SdM :~: SdL :~: _) v =
  SdL :~: cCoIDiv (cCoIDivSatCoIClAux4 u v) v
{-
cCoIDiv u0 u1 = strCoRec u0 func where
  func u2@(d :~: e :~: f :~: _) =
    case d of
      SdR -> (SdR, Right (cCoIDivSatCoIClAux1 u2 u1))
      SdL -> (SdL, Right (cCoIDivSatCoIClAux4 u2 u1))
      SdM -> case e of
SdR -> (SdR, Right (cCoIDivSatCoIClAux1 u2 u1))
SdL -> (SdL, Right (cCoIDivSatCoIClAux4 u2 u1))
SdM -> case f of
  SdR -> (SdR, Right (cCoIDivSatCoIClAux1 u2 u1))
  SdL -> (SdL, Right (cCoIDivSatCoIClAux4 u2 u1))
  SdM -> (SdM, Right (cCoIToCoIDouble u2))
-}

divide :: Str -> Str -> Str
divide u v@(SdM :~: _        ) =
  divide (cCoIToCoIDouble u) (cCoIToCoIDouble v)
divide u v@(SdL :~: _        ) =
  divide (cCoIUMinus u) (cCoIUMinus v)
divide u v@(SdR :~: SdL :~: _) =
```

```
  divide (cCoIToCoIDouble u) (cCoIToCoIDouble v)
divide u v = cCoIDiv u v


infix 4 :+:
data SdReal = [Sd] :+: Str

instance Show SdReal where
  show ([] :+: v) = "(0) | " ++ show v
  show (l  :+: v) = showAux (reverse l) v
    where
      showAux []     v = "| " ++ show v
      showAux (d:ds) v = show d ++ " " ++ showAux ds v

sdToInt :: [Sd] -> Integer
sdToInt = sdToIntAux 0 0 where
  sdToIntAux acc i []     = acc
  sdToIntAux acc i (d:ds) =
    sdToIntAux (acc + toInteger (fromEnum d) * 2^i) (i + 1) ds


-- shift (k + v) i == (k + v) * 2 ^^ i
shift :: SdReal -> Int -> SdReal
shift x 0 = x
shift x i | i > 0  = shl x i where
  shl x                        0 = x
  shl ([] :+: SdM :~: u) i = shl ([ ] :+: u) (i - 1)
  shl (l  :+:  d  :~: u) i = shl (d:l :+: u) (i - 1)
shift x i | i < 0  = shr x i where
  shr x                  0 = x
  shr ([]   :+: u) i = shr ([] :+: SdM :~: u) (i + 1)
  shr (d:ds :+: u) i = shr (ds :+:  d  :~: u) (i + 1)

instance Num [Sd] where
  fromInteger 0    = []
  fromInteger n | n < 0  = negate (fromInteger (abs n))
  fromInteger n = case r of
  0 -> SdM : fromInteger q
```

```
  1 ->   d : fromInteger q
    where
      (q, r) = divMod n 2
      d = if n < 0 then SdL else SdR
  signum  = fromInteger . signum . sdToInt
  negate = map negateSd where
    negateSd SdL = SdR
    negateSd SdM = SdM
    negateSd SdR = SdL
  ds + es = fromInteger (sdToInt ds + sdToInt es)
  ds * es = fromInteger (sdToInt ds * sdToInt es)

  abs     = undefined

instance Enum [Sd] where
  fromEnum = fromEnum . toInteger
  toEnum   = toEnum . fromEnum . toInteger

instance Real [Sd] where
  toRational = toRational . toInteger

instance Integral [Sd] where
  toInteger = toIntegerAux 0 0 where
    toIntegerAux acc i [] = acc
    toIntegerAux acc i (d:ds) =
      toIntegerAux (acc + toInteger (fromEnum d) * 2^i) (i + 1) ds
  quotRem = error "Not implemented"

instance Num SdReal where
  fromInteger k = fromInteger k :+: zero
  negate (l :+: u) = negate l :+: cCoIUMinus u
  abs = undefined
  signum = undefined
  (k :+: v) + (l :+: w) = k + l + [d] :+: u where
    (d :~: u) = cCoIAverage v w
  x * y = undefined

-- intInvToStr :: Sd -> Integer -> Str
```

```
-- intInvToStr d x =  intInvToStrAux $ 1 / toRational x where

infixl 7 //
(//) :: Integer -> Integer -> SdReal
x // y = fromRational (x % y)

instance Fractional SdReal where
  fromRational rat
    | rat == 0       = 0 :+: zero
    | rat  < 0       = fromInteger w :+: intInvToStr SdL r
    | otherwise      = fromInteger w :+: intInvToStr SdR r
    where
      (w, r) = wholeRest rat

      wholeRest :: Rational -> (Integer, Rational)
      wholeRest rat = let (q, r) =
(numerator rat) `divMod` (denominator rat) in
(q, r % denominator rat)

      intInvToStr :: Sd -> Rational -> Str
      intInvToStr d r
| rest == 0   =  d  :~: zero
| whole > 0   =  d  :~: intInvToStr d rest
| otherwise   = SdM :~: intInvToStr d rest
where
  (whole, rest) = wholeRest (2 * r)


-- w > 1/4
sdDiv :: Str -> Str -> SdReal
sdDiv v w = shift ([] :+: divide (SdM :~: SdM :~: v) w) 2

sdMultSdReal :: [Sd] -> SdReal -> SdReal
sdMultSdReal = sdMultAux 0 0 where
  sdMultAux acc i    []    x = acc
  sdMultAux acc i (SdL:ds) x =
    sdMultAux (acc - shift x i) (i + 1) ds x
  sdMultAux acc i (SdR:ds) x =
```

```
    sdMultAux (acc + shift x i) (i + 1) ds x
  sdMultAux acc i (SdM:ds) x =
    sdMultAux  acc              (i + 1) ds x


sdRealDivSd :: SdReal -> Str -> SdReal
sdRealDivSd v         w@(SdM :~: _        ) =
  sdRealDivSd (shift v 1) (cCoIToCoIDouble w)
sdRealDivSd v         w@(SdL :~: _        ) =
  sdRealDivSd (-v) (cCoIUMinus w)
sdRealDivSd v         w@(SdR :~: SdL :~: _) =
  sdRealDivSd (shift v 1) (cCoIToCoIDouble w)
sdRealDivSd (k :+: v) w                     =
  sdMultSdReal k (sdDiv pOne w) + sdDiv v w


sdRealDiv :: SdReal -> SdReal -> SdReal
sdRealDiv x y@(l :+: _) | ([] :+: w) <- shift y (-(length l))  =
  shift x (-(length l)) `sdRealDivSd` w

-----------------------------------------------------------------

-- Here starts the new part of the work:

rAux :: Str -> Bool
rAux (SdR :~: SdR :~: v) = True
rAux (SdR :~: SdM :~: v) = True
rAux (SdR :~: SdL :~: SdR :~: v) = True
rAux (SdR :~: SdL :~: SdM :~: v) = True
rAux (SdM :~: SdR :~: SdR :~: v) = True
rAux (SdM :~: SdR :~: SdM :~: v) = True
rAux v = False

lAux :: Str -> Bool
lAux (SdL :~: SdL :~: v) = True
lAux (SdL :~: SdM :~: v) = True
lAux (SdL :~: SdR :~: SdL :~: v) = True
lAux (SdL :~: SdR :~: SdM :~: v) = True
```

```
lAux (SdM :~: SdL :~: SdL :~: v) = True
lAux (SdM :~: SdL :~: SdM :~: v) = True
lAux v = False

qplus :: Str -> Str
qplus(SdL :~: s) = SdM :~: SdM :~: s
qplus(SdM :~: s) = SdM :~: SdR :~: s
qplus(SdR :~: s) = SdR :~: SdM :~: s

qminus :: Str -> Str
qminus(SdL :~: s) = SdL :~: SdM :~: s
qminus(SdM :~: s) = SdM :~: SdL :~: s
qminus(SdR :~: s) = SdM :~: SdM :~: s

funcR :: (Int -> Int) -> (Int -> Str) -> Int -> Str
funcR m f n = cCoIToCoIQuad (qminus ( f (max (m 4) n)))

funcL :: (Int -> Int) -> (Int -> Str) -> Int -> Str
funcL m f n = cCoIToCoIQuad (qplus ( f (max (m 4) n)))

funcM :: (Int -> Int) -> (Int -> Str) -> Int -> Str
funcM m f n = cCoIToCoIDouble $ f (max (m 4) n)

cCoILim :: (Int -> Int) -> (Int -> Str) -> Str
cCoILim m f
| rAux (f (m 4)) = SdR :~: (cCoILim n (funcR m f))
| lAux (f (m 4)) = SdL :~: (cCoILim n (funcL m f))
| otherwise  = SdM :~: (cCoILim n (funcM m f))
where n = \p -> (m (p+1))

heron :: Str-> Int -> Str
heron s 0 = pOne
heron s n = cCoIAverage (heron s (n-1)) (cCoIDiv s (heron s (n-1)))

auxlog :: Int -> Int -> Int
auxlog p n = if (p <= 2^n) then n else auxlog p (n+1)

poslog p = auxlog p 0
```

```
{-
cCoILim poslog (heron phalf)
+1 +1 0 -1 +1 -1 +1 -1 0 0 0 0 +1 -1 +1 -1 0 0 -1 +1 +1 -1 -1 +1
+1 -1 -1 +1 +1 Interrupted.
-}

cCoIsqrt :: Str -> Str
cCoIsqrt (SdL :~: u) = zero
cCoIsqrt (SdM :~: SdL :~: u) = zero
cCoIsqrt (SdM :~: SdM :~: u) = SdM :~: (cCoIsqrt u)
cCoIsqrt (SdM :~: SdR :~: SdL :~: u) =
  SdM :~: (cCoIsqrt (SdR :~: u))
cCoIsqrt (SdR :~: SdL :~: SdL :~: u) =
  SdM :~: (cCoIsqrt (SdR :~: u))
cCoIsqrt u = cCoILim id (heron u)

facul :: Int -> Int
facul n = faculacc n 1
  where faculacc n accu =
  if (n == 0) then accu else faculacc (n-1) (n*accu)

liouvilleAux :: (Int -> Int) -> Int -> Int -> Str
liouvilleAux f m n = if (f(m) == n)
   then (SdR :~: liouvilleAux f (m+1) (n+1))
   else  (SdM :~: liouvilleAux f m (n+1))

liouville f = liouvilleAux f 1 1

{-
cCoILim poslog (heron (liouville facul))
+1 +1 +1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 +1 -1 0 +1 -1 0
+1 -1 0 +1 -1 Interrupted.
-}
```

Appendix B

Real numbers

Motivation B.1. There are many alternatives to define real numbers and even various ways to define real numbers as Cauchy sequences like [30, Chapter 2]. We present a definition from [155] which is also the definition of real numbers in Minlog. This definition can be done in **TCF** as well as in $\mathbf{HA}^\omega$ and $\mathbf{PA}^\omega$. For **TCF** note that all objects we consider here are by default total.

In this chapter we do not give any proofs but we state the Minlog name of each theorem and definition. Therefore, the reader can look the proof up in the Minlog file `lib/rea.scm`.

In this dissertation, real numbers are used in Chapter 5 and Chapter 6. In both chapters real numbers can be seen as abstract objects which fulfil certain properties. Therefore, this excursion about the definition of real numbers in not necessary to understand the theory in this dissertation.

Definition B.2. A pair $x := ((a_n)_n, M) \in (\mathbb{N} \to \mathbb{Q}) \times (\mathbb{Z}^+ \to \mathbb{N})$ is a *real number* if

$$\forall_p \forall_{n,m \geq M(p)} |a_n - a_m| \leq 2^{-p}$$

and M is non-decreasing. In Minlog this is formulated as a predicate denoted by `Real` of arity $(\mathbb{N} \to \mathbb{Q}) \times (\mathbb{Z}^+ \to \mathbb{N})$.

Remark B.3. The definitions of the positive integers $\mathbb{Z}^+$, the integers $\mathbb{Z}$ and $\mathbb{Q}$ in **TCF** are already known from Example 2.1.5. Here and in the following we assume that the standard functions like $+, -, \cdot, /, \ldots$ on the natural numbers, the (positive) integers and the rational numbers are already known. For our purpose an intuitive understanding is sufficient. Concrete definitions can be found in the corresponding library files of Minlog.

In **TCF** positive integers are binary numbers which simplifies the calculation with these numbers and should therefore be preferred. However, in $\mathbf{HA}^\omega$ and $\mathbf{PA}^\omega$ we need another definition of (positive) integers. In this case $\mathbb{Z}^+$ can be identified with $\mathbb{N}$, where a natural number n is interpreted as the positive number $n+1$. Then define $\mathbb{Z} := \mathbb{Z}^+ + \mathbb{U} + \mathbb{Z}^+$ and $\mathbb{Q} := \mathbb{Z} \times \mathbb{Z}^+$.

Lemma B.4 (`RealBdProp`). Let $x := ((a_n)_n, M)$ be a real number. Then there is a positive integer p_x such that $\forall_n |a_n| \leq 2^{p_x}$. In Minlog p_x is denoted by `RealBd as M`.

Definition B.5. Let $x := ((a_n)_n, M)$ and $y := ((b_n)_n, N)$ be real numbers. We say that x is smaller or equal than y and write $x \leq y$ if

$$\forall_p a_{M(p+1)} \leq b_{N(p+1)} + 2^{-p}.$$

In Minlog this relation is denoted by `RealLe` or as infix by <<=.

Lemma B.6 (`RealLeRefl`, `RealLeAntiSym`, `RealLeTrans`). The relation $\leq$ on the real numbers is reflexive, antisymmetric and transitive.

Definition B.7. Let $x := ((a_n)_n, M)$ and $y := ((b_n)_n, N)$ be real numbers. We say that x and y are equal, and write $x = y$ if

$$x \leq y \wedge y \leq x.$$

In Minlog the real equality is denoted by `RealEq` or as infix by `===`.

Lemma B.8 (`RealEqRefl`, `RealEqSym`, `RealEqTrans`). The relation $=$ on the real numbers is reflexive, symmetric and transitive.

Remark B.9. The real equality is in general not decidable in a constructive setting like **TCF** and $\mathbf{HA}^\omega$. To see this, we fix an arbitrary Turing machine TM and define the real number $x_{\mathsf{TM}} := ((a_n)_n, M)$ as follows

$$a_n := \begin{cases} 0 & \text{if } \mathsf{TM} \text{ does not terminate in at most } n \text{ steps} \\ 2^{-k} & \text{if } \mathsf{TM} \text{ terminates in } k \leq n \text{ steps} \end{cases}$$

and $M(p) := p + 1$. One can easily check that x_{TM} is a real numbers, and x_{TM} is equal to 0 if and only if TM terminates. Hence, if $(x_{\mathsf{TM}} = 0) \vee (x_{\mathsf{TM}} \neq 0)$ were constructively provable for all Turing machines TM, we would get a solution to the Halting problem.

Definition B.10. Let $x := ((a_n)_n, M)$ be a real number and p be a positive number, then x is called *p-positive* if

$$2^p \leq a_{M(p+1)}.$$

In this case we write $x \in_p \mathbb{R}^+$. We write $x \in \mathbb{R}^+$ if there exists a p with $x \in_p \mathbb{R}^+$. In Minlog $x \in_p \mathbb{R}^+$ is denoted by `RealPos x p`.

Lemma B.11 (`RealLeCompat`). The relation $\leq$ is compatible with the real equality.

Definition B.12. Let $x := ((a_n)_n, M)$ and $y := ((b_n)_n, N)$ be real numbers. For each arithmetic function, we define a new real number $z := ((c_n)_n, L)$ as follows:

z	c_n	$L(p)$	Minlog name
$\lvert x\rvert$	$\lvert a_n\rvert$	$M(p)$	`RealAbs`
$x + y$	$a_n + b_n$	$\max\{M(p+1), N(p+1)\}$	`RealPlus`
$-x$	$-a_n$	$M(p)$	`RealUMinus`
xy	$a_n b_n$	$\max\{M(p+p_y+1), N(p+p_x+1)\}$	`RealTimes`
$\frac{1}{x}$	$\begin{cases} \frac{1}{a_n} & \text{if } a_n \neq 0 \\ 0 & \text{else} \end{cases}$	$M(2q+2+p)$	`RealUDiv`

Here, p_x and p_y are defined as in Lemma B.4. Note that the definition of $\frac{1}{x}$ depends on a given positive integer q which is suppressed in the notation. If we want to point out the dependence, we write $\left(\frac{1}{x}\right)_q$. Of course, in Minlog q is always explicit by writing `RealUDiv x q`.

Lemma B.13 (`RealAbsReal`, `RealPlusReal`, `RealUMinusReal`, `RealTimesReal`, `RealUDivReal`). Let $x := ((a_n)_n, M)$ and $y := ((b_n)_n, N)$ be real numbers. Then $\lvert x\rvert$, $x + y$, $-x$, xy are real numbers. If $\lvert x\rvert \in_q \mathbb{R}^+$ for some positive integer q then also $\left(\frac{1}{x}\right)_q$ is a real number.

Lemma B.14. Let x, y, z, z_0 be real numbers, then the following statements hold. Note that this is just a selection of the most used properties.

`RealPlusAssoc`	$x + (y + z) = (x + y) + z$
`RealPlusComm`	$x + y = y + x$
`RealPlusZero`	$x + 1 = x$
`RealPlusMinusZero`	$x + (-x) = 0$

`RealTimesAssoc`	$x(yz) = (xy)z$
`RealTimesComm`	$xy = yx$
`RealTimesOne`	$x1 = x$
`RealTimesZero`	$x0 = 0$
`RealTimesUDivR`	$x \in_p \mathbb{R}^+ \to x\frac{1}{x} = 1$
`RealTimesPlusDistr`	$x(y + z) = xy + xz$
`RealAbsTimes`	$\lvert xy\rvert = \lvert x\rvert\lvert y\rvert$
`RealAbsAbs`	$\lvert\lvert x\rvert\rvert = \lvert x\rvert$
`RealAbsUMinus`	$\lvert -x\rvert = \lvert x\rvert$
`RealLeMonPlus`	$x \leq y \to z \leq z_0 \to x + z \leq y + z_0$
`RealLeMonTimesR`	$0 \leq x \to y \leq z \to xy \leq xz$
`RealLeAbsPlus`	$\lvert x + y\rvert \leq \lvert x\rvert + \lvert y\rvert$
`RealLeIdAbs`	$x \leq \lvert x\rvert$
`RealPosMonPlus`	$x \in \mathbb{R}^+ \to y \in \mathbb{R}^+ \to x + y \in \mathbb{R}^+$
`RealLeMonTimesTwo`	$0 \leq x \leq y \to 0 \leq z \leq z_0 \to xz \leq yz_0$
`RealPosLe`	$x \leq y \to x \in_p \mathbb{R}^+ \to y \in_{p+2} \mathbb{R}^+$

Lemma B.15 (`RealAbsCompat`, `RealPlusCompat`, `RealUMinusCompat`, `RealTimesCompat`, `RealUDivCompat`). The arithmetic functions from Definition B.12 are compatible with the real equality. In particular, for the division we have: if $|x|$ is p-positive, $|y|$ is q-positive and $x = y$ then $\left(\frac{1}{x}\right)_p = \left(\frac{1}{y}\right)_q$.

Motivation B.16. In the constructive setting we have a weaker version of decidability of $\leq$, which suffices in many cases.

Lemma B.17 (`ApproxSplit`). Let x, y, z be real numbers with $x < y$, i.e. $y - x \in \mathbb{R}^+$, then $x \leq z$ or $z \leq y$.

Bibliography

[1] Ya. I. Alber. Recurrence relations and variational inequalities. In *Doklady Akademii Nauk*, volume 270, pages 11–17. Russian Academy of Sciences, 1983.

[2] Ya. I. Alber, C.E. Chidume, and H. Zegeye. Approximating fixed points of total asymptotically nonexpansive mappings. *Fixed Point Theory and Applications*, 2006(1):1–20, 2006.

[3] Ya. I. Alber and S. Guerre-Delabriere. Principle of Weakly Contractive Maps in Hilbert Spaces. In Gohberg I. and Lyubich Y., editors, *New Results in Operator Theory and Its Applications*, volume 98 of *Operator Theory: Advances and Applications*, pages 7–22. Birkhäuser, 1997.

[4] Ya. I. Alber and S. Guerre-Delabriere. On the projection methods for fixed point problems. *Analysis*, 21(1):17–40, 2001.

[5] Ya. I. Alber and S. Reich. An iterative method for solving a class of nonlinear operator equations in Banach spaces. *Panamericam Mathematical Journal*, 4(2):39–54, 1994.

[6] Yakov Alber, Simeon Reich, and Jen-Chih Yao. Iterative methods for solving fixed-point problems with nonself-mappings in Banach spaces. *Abstract and Applied Analysis*, 2003(4):193–216, 2003.

[7] Yakov I. Alber and A.N. Iusem. Extension of subgradient techniques for nonsmooth optimization in banach spaces. *Set-Valued Analysis*, 9(4):315–335, 2001.

[8] Maria Emilia Alonso, Thierry Coquand, and Henri Lombardi. Revisiting Zariski Main Theorem from a constructive point of view. *Journal of Algebra*, 406:46–68, 2014.

[9] Emil Artin. Über die Zerlegung definiter Funktionen in Quadrate. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 5, pages 100–115. Springer, 1927.

[10] Emil Artin and Otto Schreier. Algebraische konstruktion reeller Körper. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 5, pages 85–99. Springer, 1927.

[11] Federico Aschieri and Stefano Berardi. Interactive Learning-Based Realizability Interpretation for Heyting Arithmetic with EM 1. In Pierre-Louis Curien, editor, *Typed Lambda Calculi and Applications*, pages 20–34. Springer Berlin Heidelberg, 2009.

[12] Michael F. Atiyah and Ian G. Macdonald. *Introduction to Commutative Algebra*. Addison–Wesley, Reading, MA, 1969.

[13] Alborz Azarang. A simple proof of Zariski's Lemma. *Bulletin of the Iranian Mathematical Society*, 43(5):1529–1530, 2017.

[14] B. Banaschewski and J.J.C. Vermeulen. Polynomials and radical ideals. *Journal of pure and applied algebra*, 113(3):219–227, 1996.

[15] Sami Barhoumi. Seminormality and polynomial rings. *Journal of Algebra*, 322(6):1974–1978, 2009.

[16] Sami Barhoumi and Henri Lombardi. An algorithm for the Traverso–Swan theorem on seminormal rings. *Journal of Algebra*, 320(4):1531–1542, 2008.

[17] Sami Barhoumi, Henri Lombardi, and Ihsen Yengui. Projective modules over polynomial rings: a constructive approach. *Mathematische Nachrichten*, 282(6):792–799, 2009.

[18] Gianluigi Bellin. Ramsey interpreted: a parametric version of Ramsey's theorem. *Logic and computation (Pittsburgh, PA, 1987)*, 106:17–37, 1990.

[19] Ulrich Berger. A Computational Interpretation of Open Induction. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 326–334. IEEE, 2004.

[20] Ulrich Berger. From Coinductive Proofs to Exact Real Arithmetic. In E. Grädel and R. Kahle, editors, *Computer Science Logic*, Lecture Notes in Computer Science, pages 132–146. Springer, 2009.

[21] Ulrich Berger. From coinductive proofs to exact real arithmetic: theory and applications. *Logic Methods in Computer Science*, 7(1):1–24, 2011.

[22] Ulrich Berger, Wilfried Buchholz, and Helmut Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114(1-3):3–25, 2002.

[23] Ulrich Berger and Tie Hou. Coinduction for Exact Real Number Computation. *Theory of Computing Systems*, 43:394–409, 2008.

[24] Ulrich Berger, Andrew Lawrence, Fredrik Nordvall Forsberg, and Monika Seisenberger. Extracting verified decision procedures: DPLL and resolution. *Logical Methods in Computer Science*, 11(1):1–18, 2015.

[25] Ulrich Berger, Kenji Miyamoto, Helmut Schwichtenberg, and Monika Seisenberger. Minlog – A Tool for Program Extraction Supporting Algebras and Coalgebras. In A. Corradini, B. Klin, and C. Cirstea, editors, *Algebra and Coalgebra in Computer Science*, volume 6859 of *Lecture Notes in Computer Science*, pages 393–399. Springer, 2011.

[26] Ulrich Berger, Kenji Miyamoto, Helmut Schwichtenberg, and Hideki Tsuiki. Logic for Gray-code Computation. In *Concepts of proof in mathematics, philosophy, and computer science*, pages 69–110. De Gruyter, 2016.

[27] Ulrich Berger and Helmut Schwichtenberg. Program extraction from classical proofs. In Daniel Leivant, editor, *Logic and Computational Complexity*, volume 960 of *Lecture Notes in Computer Science*, pages 77–97. Springer Berlin Heidelberg, 1994.

[28] Ulrich Berger and Monika Seisenberger. Proofs, programs, processes. *Theory of Computing Systems*, 51(3):313–329, 2012.

[29] Errett Bishop. *Foundations of constructive analysis*. McGraw-Hill Book Company, 1967.

[30] Errett Bishop and Douglas Bridges. *Constructive Analysis*, volume 279 of *A Series of Comprehensive Studies in Mathematics*. Springer-Verlag Berlin Heidelberg, 1985.

[31] Ingo Blechschmidt. Garben und Logik. `https://www.youtube.com/watch?v=Sij1jIL2gEg&list=PLR-3Jx6BfhkgjagoGM3Z15G0j5hv5i66v`, 2017. Recorded lecture at the Universität Augsburg [Accessed April 21, 2021].

[32] Siegfried Bosch. *Algebra*. Springer-Verlag, 5 edition, 2004.

[33] D. S. Bridges, F. Richman, W. H. Julian, and R. Mines. Extensions and Fixed Points of Contractive Maps in $\mathbb{R}^n$. *Journal of Mathematical Analysis and Applications*, 165(2):438–456, 1992.

[34] Eyvind Martol Briseid. *On Rates of Convergence in Metric Fixed Point Theory*. PhD thesis, Technische Universität Darmstadt, 2009.

[35] Jan Cederquist and Thierry Coquand. Entailment Relations and Distributive Lattices. In Samule R. Buss, Petr Hajek, and Pavel Pudlak, editors, *Logic Colloquium '98*, volume 98 of *Lectire Notes in Logic*, pages 127–139. Association for Symbolic Logic, Cambridge University Press, 2000.

[36] Jan Cederquist and Sara Negri. A constructive proof of the Heine-Borel covering theorem for formal reals. In *Types for Proofs and Programs*, volume 1158 of *Lecture Notes in Computer Science*, pages 62–75. Springer, 1996.

[37] C.E. Chidume, H. Zegeye, and S.J. Aneke. Approximation of fixed points of weakly contractive nonself maps in Banach spaces. *Journal of Mathematical Analysis and Applications*, 270(1):189–199, 2002.

[38] Alberto Ciaffaglione and Pietro Di Gianantonio. A certified, corecursive implementation of exact real numbers. *Theoretical Computer Science*, 351(1):39–51, 2006.

[39] Francesco Ciraulo, Davide Rinaldi, and Peter Schuster. Lindenbaum's lemma via open induction. In *Advances in Proof Theory*, volume 28 of *Progress in Computer Science and Applied Logic*, pages 65–77. Springer, 2016.

[40] Thierry Coquand. A note on the open induction principle. Technical report, Göteborg University, 1997.

[41] Thierry Coquand. Sur un théorème de Kronecker concernant les variétés algébriques. *Comptes Rendus Mathematique*, 338(4):291–294, 2004.

[42] Thierry Coquand. On seminormality. *Journal of Algebra*, 305(1):577–584, 2006.

[43] Thierry Coquand. Space of valuations. *Annals of Pure and Applied Logic*, 157(2-3):97–109, 2009.

[44] Thierry Coquand. Constructive Algebra. `https://www.youtube.com/watch?v=Guy60QCTNrk`, 2018. Recored talk at the Hausdorff Trimester Program: Types, Sets and Constructions, Bonn [Accessed April 21, 2021].

[45] Thierry Coquand, Lionel Ducos, Henri Lombardi, and Claude Quitté. L'idéal des coefficients du produit de deux polynômes. *Revue des Mathématiques de l'enseignement Supérieur*, 113(3):25–39, 2003.

[46] Thierry Coquand, Lionel Ducos, Henri Lombardi, and Claude Quitté. Constructive Krull Dimension. I: Integral Extensions. *Journal of Algebra and Its Applications*, 8(1):129–138, 2009.

[47] Thierry Coquand and Henri Lombardi. Hidden constructions in abstract algebra: Krull dimension of distributive lattices and commutative rings. In M. Fontana, S.-E. Kabbaj, and S. Wiegand, editors, *Commutative Ring Theory and Applications*, volume 231 of *Lecture Notes in Pure and Applied Mathematics*, pages 477–499, Reading, MA, 2002. Addison-Wesley.

[48] Thierry Coquand and Henri Lombardi. A Short Proof for the Krull Dimension of a Polynomial Ring. *The American Mathematical Monthly*, 112(9):826–829, 2005.

[49] Thierry Coquand and Henri Lombardi. A logical approach to abstract algebra. *Mathematical Structures in Computer Science*, 16(5):885–900, 2006.

[50] Thierry Coquand, Henri Lombardi, and Stefan Neuwirth. Lattice-ordered groups generated by an ordered group and regular systems of ideals. *Rocky Mountain Journal of Mathematics*, 49(5):1449–1489, 2019.

[51] Thierry Coquand, Henri Lombardi, and Claude Quitté. Generating non-Noetherian Modules Constructively. *manuscripta mathematica*, 115(4):513–520, 2004.

[52] Thierry Coquand, Henri Lombardi, and Claude Quitté. Dimension de heitmann des treillis ditributifs et des anneaux commutatifs. *Publications Mathématiques de Besançon - Algèbre et Théorie des Nombres*, pages 57–100, 2006.

[53] Thierry Coquand, Henri Lombardi, and Marie-Françoise Roy. An elementary characterization of Krull dimension. In L. Crosilla and P. Schuster, editors, *From Sets and Types to Topology and Analysis*, volume 48 of *Oxford Logic Guides*, pages 239–244. Oxford University Press, 2005.

[54] Thierry Coquand, Henri Lombardi, and Peter Schuster. A nilregular element property. *Archiv der Mathematik*, 85(1):49–54, 2005.

[55] Thierry Coquand, Henri Lombardi, and Peter Schuster. Spectral schemes as ringed lattices. *Annals of Mathematics and Artificial Intelligence*, 56(3):339–360, 2009.

[56] Thierry Coquand and Henrik Persson. Valuations and Dedekind's Prague theorem. *Journal of Pure and Applied Algebra*, 155(2-3):121–129, 2001.

[57] Michel Coste, Henri Lombardi, and Marie-Françoise Roy. Dynamical method in algebra: effective Nullstellensätze. *Annals of Pure and Applied Logic*, 111(3):203–256, 2001.

[58] Laura Crosilla and Peter Schuster. Finite Methods in Mathematical Practice. In G. Link, editor, *Formalism and Beyond. On the Nature of Mathematical Discourse*, volume 23 of *Logos*, pages 351–410. Walter de Gruyter, Boston and Berlin, 2014.

[59] Klaus Deimling. *Nonlinear Functional Analysis*. Springer-Verlag, 1985.

[60] Jean Della Dora, Claire Dicrescenzo, and Dominique Duval. About a new method for computing in algebraic number fields. In *EUROCAL '85 European Conference on Computer Algebra*, volume 204 of *Lecture Notes in Computer Science*, pages 289–290. Springer, 1985.

[61] C. N. Delzell. A continuous, constructive solution to Hilbert's 17^{th} problem. *Inventiones mathematicae*, 76(3):365–384, 1984.

[62] Charles N. Delzell. Kreisel's unwinding of Artin's proof - Part I. In Piergiorgio Odifreddi, editor, *Kreiseliana. About and Around Georg Kreisel*, pages 113–245. A K Peters, Wellesley, MA, 1996.

[63] Pietro Di Gianantonio. An abstract data type for real numbers. *Theoretical Computer Science*, 221(1-2):295–326, 1999.

[64] Joseph Distel. *Geometry of Banach Spaces - Selected Topics*, volume 485 of *Lecture Notes in Mathematics*. Springer-Verlag, 1975.

[65] Lionel Ducos. Sur les théorèmes de Serre, Bass et Forster–Swan. *Comptes Rendus Mathematique*, 339(8):539–542, 2004.

[66] Lionel Ducos, Henri Lombardi, Claude Quitté, and Maimouna Salou. Théorie algorithmique des anneaux arithmétiques, des anneaux de Prüfer et des anneaux de Dedekind. *Journal of Algebra*, 281(2):604–650, 2004.

[67] Michael Edelstein. On fixed and periodic points under contractive mappings. *Journal of the London Mathematical Society*, 37(1):74–79, 1962.

[68] Harold M. Edwards. *Divisor theory*. Springer Science & Business Media, 2012.

[69] David Eisenbud. *Commutative Algebra: with a View Toward Algebraic Geometry*, volume 150 of *Graduate Texts in Mathematics*. Springer Science & Business Media, 2013.

[70] Yu. L. Ershov. Model C of Partial Continuous Functionals. In R.O. Gandy and J.M.E. Hyland, editors, *Logic Colloquium 76*, volume 87 of *Studies in Logic and the Foundations of Mathematics*, pages 455–467. Elsevier, 1977.

[71] L. Espanol. The spectrum lattice of Baer rings and polynomials. In *Categorical Algebra and its Applications*, volume 1348 of *Lecture Notes in Mathematics*, pages 118–124. Springer, 1988.

[72] Solomon Feferman. Hilbert's program relativized; Proof-theoretical and foundational reductions. *The Journal of Symbolic Logic*, 53(2):364–384, 1988.

[73] Giulio Fellin, Peter Schuster, and Daniel Wessel. The Jacobson radical of a propositional theory. In *Proceedings of Third Tübingen Conference on Proof-Theoretic Semantics*, volume 15, pages 1–16, 2019.

[74] Harvey Friedman. Classically and intuitionistically provably recursive functions. In Gert H. Müller and Dana S. Scott, editors, *Higher Set Theory*, volume 669 of *Lecture Notes in Mathematics*, pages 21–27. Springer, Berlin, Heidelberg, 1978.

[75] Philipp Gerhardy. A quantitative version of Kirk's fixed point theorem for asymptotic contractions. *Journal of Mathematical Analysis and Applications*, 316(1):339–345, 2006.

[76] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[77] Kurt Gödel. Über eine bisher noch nicht benutzte Erweiterung des finiten Standpunktes. *Dialectica*, 12(3-4):280–287, 1958.

[78] Yuri Gurevich. Sequential abstract – state machines capture sequential algorithms. *ACM Transactions on Computational Logic*, 1(1):77–111, 2000.

[79] Matthew Hendtlass. The intermediate value theorem in constructive mathematics without choice. *Annals of Pure and Applied Logic*, 163(8):1050–1056, 2012.

[80] Paul Hertz. Über Axiomensysteme für beliebige Satzsysteme. Teil II. Sätze höheren Grades. *Mathematische Annalen*, 89(1):76–102, 1923.

[81] David Hilbert. Die Grundlegung der elementaren Zahlenlehre. *Mathematische Annalen*, 104(1):485–494, 1931.

[82] Klaus Hulek. *Elementary Algebraic Geometry*, volume 20 of *Student Mathematical Libary*. American Mathematical Society, 2003.

[83] Klaus Hulek. *Elementare algebraische Geometrie: grundlegende Begriffe und Techniken mit zahlreichen Beispielen und Anwendungen*. Springer-Verlag, 2012.

[84] William Julian H., Ray Mines, and Fred Richman. The intermediate value theorem: Preimages of compact sets under uniformly continuous functions. *The Rocky Mountain Journal of Mathematics*, 18(1):25–36, 1988.

[85] William Art Kirk. Fixed points of asymptotic contractions. *Journal of Mathematical Analysis and Applications*, 277(2):645–650, 2003.

[86] Manfred Knebusch and Claus Scheiderer. *Einführung in die reelle Algebra*, volume 63 of *Aufbaukurs Mathematik*. Vieweg-Verlag, 1989.

[87] Ulrich Kohlenbach. Applied foundations: proof mining in analysis. *Newsletter of the Danish Mathematical Society (Mathilde)*, 13:7–9, 2002.

[88] Ulrich Kohlenbach. Some computational aspects of metric fixed-point theory. *Nonlinear Analysis: Theory, Methods & Applications*, 61(5):823–837, 2005.

[89] Ulrich Kohlenbach. Some logical metatheorems with applications in functional analysis. *Transactions of the American Mathematical Society*, 357(1):89–128, 2005.

[90] Ulrich Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Monographa in Mathematics. Springer Science & Business Media, 2008.

[91] Ulrich Kohlenbach. A note on the monotone functional interpretation. *Mathematical Logic Quarterly*, 57(6):611–614, 2011.

[92] Ulrich Kohlenbach. Proof-theoretic Methods in Nonlinear Analysis. In *Proceedings of the International Congress of Mathematicians*, volume 2, pages 61–82. World Scientific, 2018.

[93] Ulrich Kohlenbach and Angeliki Koutsoukou-Argyraki. Rates of convergence and metastability for abstract Cauchy problems generated by accretive operators. *Journal of Mathematical Analysis and Applications*, 423(2):1089–1112, 2015.

[94] Ulrich Kohlenbach and Laurenţiu Leuştean. The approximate fixed point property in product spaces. *Nonlinear Analysis: Theory, Methods & Applications*, 66(4):806–818, 2007.

[95] Ulrich Kohlenbach and Laurenţiu Leuştean. Asymptotically nonexpansive mappings in uniformly convex hyperbolic spaces. *Journal of the European Mathematical Society*, 12(1):71–92, 2009.

[96] Ulrich Kohlenbach and Laurenţiu Leuştean. Effective metastability of Halpern iterates in CAT(0) spaces. *Advances in Mathematics*, 231(5):2526–2556, 2012.

[97] Ulrich Kohlenbach and Laurenţiu Leuştean. On the computational content of convergence proofs via Banach limits. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1971):3449–3463, 2012.

[98] Ulrich Kohlenbach and Paulo Oliva. Proof Mining: A Systematic Way of Analyzing Proofs in Mathematics. *Proceedings of the Steklov Institute of Mathematics*, 242:136–164, 2003.

[99] Ulrich Kohlenbach and Paulo Oliva. Proof mining in L_1-approximation. *Annals of Pure and Applied Logic*, 121(1):1–38, 2003.

[100] Ulrich Kohlenbach and Thomas Powell. Rates of convergence for iterative solutions of equations involving set-valued accretive operators. *Computers & Mathematics with Applications*, 80(3):490–503, 2020.

[101] Ulrich Kohlenbach and Andrei Sipoş. The finitary content of sunny nonexpansive retractions. *Communications in Contemporary Mathematics*, 23(01):1950093, 2021.

[102] Nils Köpp. Automatically verified program extraction from proofs with applications to constructive analysis. MSc Thesis, Ludwig Maximilians University Munich, 2018.

[103] Daniel Körnlein and Ulrich Kohlenbach. Effective rates of convergence for Lipschitzian pseudocontractive mappings in general Banach spaces. *Nonlinear Analysis: Theory, Methods & Applications*, 74(16):5253–5267, 2011.

[104] Georg Kreisel. On the Interpretation of Non-Finitist Proofs: Part I. *The Journal of Symbolic Logic*, 16(4):241–267, 1951.

[105] Georg Kreisel. On the Interpretation of Non-Finitist Proofs: Part II. Interpretation of Number Theory. Applications. *The Journal of Symbolic Logic*, 17(1):43–58, 1952.

[106] Georg Kreisel. Hilbert's programme. *Dialectica*, 12(3-4):346–372, 1958.

[107] Georg Kreisel. Mathematical Significance of Consistency Proofs. *The Journal of Symbolic Logic*, 23(2):155–182, 1958.

[108] Joseph Liouville. Nouvelle démonstration d'un théoreme sur les irrationnelles algébriques inséré dans le compte rendu de la derniere séance. *CR Acad. Sci. Paris*, 18:910–911, 1844.

[109] Henri Lombardi. Platitude, localisation et anneaux de Prüfer: une approche constructive. *Publications Mathématiques de Besançon - Algèbre et Théorie des Nombres*, pages 1–65, 2001.

[110] Henri Lombardi. Dimension de Krull, Nullstellensätze et Évaluation Dynamique. *Mathematische Zeitschrift*, 242(1):23–46, 2002.

[111] Henri Lombardi. Hidden constructions in abstract algebra (1): integral dependance. *Journal of Pure and Applied Algebra*, 167(2-3):259–267, 2002.

[112] Henri Lombardi. Structures algébriques dynamiques, espaces topologiques sans points et programme de Hilbert. *Annals of Pure and Applied Logic*, 137(1-3):256–290, 2006.

[113] Henri Lombardi and Claude Quitté. Constructions cachées en algebre abstraite (2) le principe local global. In M. Fontana, S.-E. Kabbaj, and S. Wiegand, editors, *Commutative Ring Theory and Applications*, volume 231 of *Lecture Notes in Pure and Applied Mathematics*, pages 461–476. CRC Press, 2002.

[114] Henri Lombardi and Claude Quitté. Seminormal rings (following Thierry Coquand). *Theoretical computer science*, 392(1-3):113–127, 2008.

[115] Henri Lombardi and Claude Quitté. *Commutative algebra: Constructive methods. Finite Projective Modules*, volume 20 of *Algebra and Applications*. Springer, Dordrecht, 2015.

[116] Henri Lombardi and Marie-Françoise Roy. Elementary constructive theory of ordered fields. In T. Mora and C. Traverso, editors, *Effective methods in algebraic geometry*, volume 94 of *Progress in Mathematics*, pages 249–262. Birkhäuser, Boston, MA, 1991.

[117] Horst Luckhardt. Herbrand-Analysen zweier Beweise des Satzes von Roth: Polynomiale Anzahlschranken. *Journal of Symbolic Logic*, 54(1):234–263, 1989.

[118] John McCabe. A Note on Zariski's Lemma. *The American Mathematical Monthly*, 83(7):560–561, 2018.

[119] Albert Menne and Joseph M. Bochéski. *Grundriß der formalen Logik*. Universitäts-Taschen-Bücher-Verlag, 1983.

[120] William Messing and Victor Reiner. A universal coefficient theorem for Gauss's Lemma. *Journal of Commutative Algebra*, 5(2):299–307, 2013.

[121] Ray Mines, Fred Richman, and Wim Ruitenburg. *A Course in Constructive Algebra*. Springer, New York, 1988.

[122] Kenji Miyamoto. The Minlog System. `http://www.mathematik.uni-muenchen.de/~logik/minlog/index.php`, 2017. [Accessed on April 25, 2021].

[123] Kenji Miyamoto and Helmut Schwichtenberg. Program extraction in exact real arithmetic. *Mathematical Structures in Computer Science*, (Special issue 8):1692–1704, 2015.

[124] Christopher J. Mulvey and Joan Wick Pelletier. A globalization of the Hahn-Banach theorem. *Advances in Mathematics*, 89(1):1–59, 1991.

[125] Sara Negri, Jan Von Plato, and Thierry Coquand. Proof-theoretical analysis of order relations. *Archive for Mathematical Logic*, 43(3):297–309, 2004.

[126] Stefan Neuwirth. Lorenzen's reshaping of Krull's Fundamentalsatz for integral domains (1938–1953). *arXiv preprint arXiv:2007.08625*, 2020.

[127] Paulo Oliva. *Proof Mining in Subsystems of Analysis*. PhD thesis, University of Aarhus, 2003.

[128] Paulo Oliva and Thomas Powell. A Game-Theoretic Computational Interpretation of Proofs in Classical Analysis. In Rathjen M. Kahle R., editor, *Gentzen's Centenary*, pages 501–531. Springer, 2015.

[129] Paulo Oliva and Thomas Powell. Bar recursion over finite partial functions. *Annals of Pure and Applied Logic*, 168(5):887–921, 2017.

[130] Hervé Perdry. Strongly Noetherian rings and constructive ideal theory. *Journal of Symbolic Computation*, 37(4):511–535, 2004.

[131] Hervé Perdry. Lazy bases: a minimalist constructive theory of Noetherian rings. *Mathematical Logic Quarterly*, 54(1):70–82, 2008.

[132] Henrik Persson. An application of the constructive spectrum of a ring. In *Type Theory and the Integrated Logic of Programs*. Chalmers University and University of Göteborg, 1999. PhD thesis.

[133] Thomas Powell. *On Bar Recursive Interpretations of Analysis*. PhD thesis, Queen Mary University of London, 2013.

[134] Thomas Powell. Gödel's functional interpretation and the concept of learning. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 136–145, 2016.

[135] Thomas Powell. Sequential algorithms and the computational content of classical proofs. *arXiv preprint arXiv:1812.11003*, 2018.

[136] Thomas Powell, Peter Schuster, and Franziskus Wiesnet. An algorithmic approach to the existence of ideal objects in commutative algebra. In R. Iemhoff, M. Moortgat, and R. de Queiroz, editors, *Logic, Language, Information and Computation*, volume 11541 of *Lectures Notes in Computer Science*. Springer-Verlag, July 2019.

[137] Thomas Powell, Peter Schuster, and Franziskus Wiesnet. A universal algorithm for Krull's theorem. *Information and Computation*, 2021. To appear.

[138] Thomas Powell and Franziskus Wiesnet. Rates of convergence for asymptotically weakly contractive mappings in normed spaces. *arXiv preprint arXiv:2104.14495*, 2021. submitted.

[139] J.L. Rabinowitsch. Zum Hilbertschen Nullstellensatz. *Mathematische Annalen*, 102(1):520, 1930.

[140] Jean-Claude Raoult. Proving open properties by induction. *Information Processing Letters*, 29(1):19–23, 1988.

[141] Michael Rathjen. The constructive Hilbert program and the limits of Martin-Löf type theory. In Segerberg K. Stoltenberg-Hansen V. Lindström S., Palmgren E., editor, *Logicism, Intuitionism, and Formalism*, volume 341 of *Synthese Library (Studies In Epistemology. Logic, Methodology, and Philosophy of Science)*, pages 397–433. Springer, 2009.

[142] Miles Reid. *Undergraduate Algebraic Geometry*, volume 12 of *London Mathematical Society Student Texts*. Cambridge University Press, 1988.

[143] Fred Richman. Nontrivial uses of trivial rings. *Proceedings of the American Mathematical Society*, 103(4):1012–1014, 1988.

[144] Davide Rinaldi and Peter Schuster. A universal Krull–Lindenbaum theorem. *Journal of Pure and Applied Algebra*, 220(9):3207–3232, 2016.

[145] Davide Rinaldi, Peter Schuster, and Daniel Wessel. Eliminating disjunctions by disjunction elimination. *Indagationes Mathematicae*, 29(1):226–259, 2018.

[146] Davide Rinaldi and Daniel Wessel. Extension by conservation. Sikorski's theorem. *Logical Methods in Computer Science*, 14(4:8):1–17, 2018.

[147] Giovanni Sambin. Intuitionistic Formal Spaces — A First Communication. In D. G. Skordev, editor, *Mathematical Logic and its Applications, Proc. Adv. Internat. Summer School Conf., Druzhba, Bulgaria, 1986*, pages 187–204. Plenum, New York, 1987.

[148] Sam Sanders. Metastability and higher-order computability. In Nerode A. Artemov S., editor, *Logical Foundations of Computer Science*, volume 10703 of *Lecture Notes in Computer Science*, pages 309–330. Springer, 2018.

[149] Erhard Scholz. Die Gödelschen Unvollständigkeitssätze und das Hilbertsche Programm einer „finiten“ Beweistheorie. In *Wolfgang Achtner: Künstliche Intelligenz und menschliche Person*, pages 15–38, 2006.

[150] Peter Schuster. Formal Zariski topology: Positivity and points. *Annals of Pure and Applied Logic*, 137(1-3):317–359, 2006.

[151] Peter Schuster. Induction in algebra: a first case study. In *2012 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia*, pages 581–585. IEEE Computer Society Publications, 2012. *Logical Methods in Computer Science* (3:20) 9 (2013).

[152] Peter Schuster and Helmut Schwichtenberg. Constructive solutions of continuous equations. In Godehard Link, editor, *One Hundred Years of Russell's Paradox*, volume 6 of *Logic and Its Applications*, pages 227–246. De Gruyter, 2003.

[153] Peter Schuster and Daniel Wessel. Resolving finite indeterminacy: A definitive constructive universal prime ideal theorem. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 820–830, New York, NY, USA, 2020. Association for Computing Machinery.

[154] Peter Schuster, Daniel Wessel, and Ihsen Yengui. Dynamic evaluation of integrity and the computational content of Krull's lemma. 2019. Preprint.

[155] Helmut Schwichtenberg. Constructive analysis with witnesses. volume 200 of *Series III: Computer and System Sciences*, pages 323–354. IOS Press, 2006.

[156] Helmut Schwichtenberg. Program Extraction in Constructive Analysis. In *Logicism, Intuitionism, and Formalism*, volume 341 of *Synthese Library (Studies In Epistemology. Logic, Methodology, and Philosophy of Science)*, pages 255–275. Springer, 2009.

[157] Helmut Schwichtenberg. Computational aspects of Bishop's constructive mathematics. In D. Bridges, H. Ishihara, H. Schwichtenberg, and M. Rathjen, editors, *Handbook of constructive mathematics*. Cambridge University Press, 2021. Submitted.

[158] Helmut Schwichtenberg, Monika Seisenberger, and Franziskus Wiesnet. Higman's lemma and its computational content. In Thomas Studer Reinhard Kahle, Thomas Strahm, editor, *Advances in Proof Theory*, volume 28 of *Progress in Computer Science and Applied Logic*. Birkhäuser, Cham, 2016.

[159] Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and Computation*. Perspectives in Logic. Cambridge University Press and Association for Symbolic Logic, 2012.

[160] Helmut Schwichtenberg and Franziskus Wiesnet. Logic for exact real arithmetic. *Logical Methods in Computer Science*, 12(2):7:1–7:22, April 2021.

[161] Dana Scott. *Outline of a mathematical theory of computation*. Oxford University Computing Laboratory, Programming Research Group Oxford, 1970.

[162] Yaghoub Sharifi. Zariski's Lemma. `https://ysharifi.wordpress.com/tag/zariskis-lemma/`, 2011. [Accessed April 26, 2021].

[163] William Simmons and Henry Towsner. Proof mining and effective bounds in differential polynomial rings. *Advances in Mathematics*, 343:567–623, 2019.

[164] Stephen G. Simpson. Partial realizations of Hilbert's program. *The Journal of Symbolic Logic*, 53(2):349–363, 1988.

[165] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Logic. Cambridge University Press and Association for Symbolic Logic, 2009.

[166] Clifford Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In J. C. E. Dekker, editor, *Recursive Function Theory*, volume 5 of *Proceedings of Symposia in Pure Mathematics*, pages 1–27. American Mathematical Society, Providence, RI, 1962.

[167] Terence Tao. Soft analysis, hard analysis, and the finite convergence principle. `https://terrytao.wordpress.com/2007/05/23/soft-analysis-hard-analysis-and-the-finite-convergence-principle/`, 2007. [Accessed on April 26, 2021].

[168] Alfred Tarski. Fundamentale begriffe der Methodologie der deduktiven Wissenschaften. I. *Monatshefte für Mathematik und Physik*, 37(1):361–404, 1930.

[169] Anne S. Troelstra and Dirk van Dalen. *Constructivism in Mathematics: an Introduction*, volume II of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1988.

[170] Hideki Tsuiki. Real number computation through Gray code embedding. *Theoretical Computer Science*, 284(2):467–485, 2002.

[171] Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[172] Till Überrück-Fries. Program extraction for exact real numbers: Stream multiplication. Bachelor's thesis, Ludwig-Maximalians Universität, 2016.

[173] Klaus Weihrauch. *Computable Analysis: An Introduction*. Texts in Theoretical Computer Science. Springer Science & Business Media, 2000.

[174] Daniel Wessel. *Choice, extension, conservation. From transfinite to finite proof methods in abstract algebra*. PhD thesis, University of Trento, 2018.

[175] Daniel Wessel. Ordering groups constructively. *Communications in Algebra*, 47(12):4853–4873, 2019.

[176] Daniel Wessel. Making the use of maximal ideals inductive, 2021. Talk at Workshop *Reducing complexity in algebra, logic, combinatorics*.

[177] Edwin Wiedmer. Computing with infinite objects. *Theoretical Computer Science*, 10(2):133–155, 1980.

[178] Franziskus Wiesnet. Konstruktive Analysis mit exakten reellen Zahlen. Master's thesis, Ludwig-Maximalians Universität, 2017.

[179] Franziskus Wiesnet. Introduction to Minlog. In Klaus Mainzer, Peter Schuster, and Helmut Schwichtenberg, editors, *Proof and Computation*, pages 233–288. World Scientific, 2018.

[180] Franziskus Wiesnet. An algorithmic version of Zariski's lemma. In L. De Mol, F. Manea, and A. Weiermann, editors, *Computability in Europe*, Lecture Notes in Computer Science. Springer, 2021. To appear.

[181] Franziskus Wiesnet and Nils Köpp. Limits of real numbers in the binary signed digit representation. *arXiv preprint arXiv:2103.15702*, 2021. submitted.

[182] Ihsen Yengui. An algorithm for the divisors of monic polynomials over a commutative ring. *Mathematische Nachrichten*, 260(1):93–99, 2003.

[183] Ihsen Yengui. Making the use of maximal ideals constructively. *Theoretical Computer Science*, 392(1-3):174–178, 2008.

[184] Ihsen Yengui. *Constructive Commutative Algebra. Projective Modules over Polynomial Rings and Dynamical Gröbner Bases*, volume 2138 of *Lecture Notes in Mathematics*. Springer, Cham, 2015.

[185] Richard Zach. Hilbert's program then and now. In Dale Jacquette, editor, *Philosophy of Logic*, Handbook of the Philosophy of Science, pages 411–447. North-Holland, Amsterdam, 2007.

[186] Oscar Zariski. A new proof of Hilbert's Nullstellensatz. *Bulletin of the American Mathematical Society*, 53(4):362–368, 1947.

Index