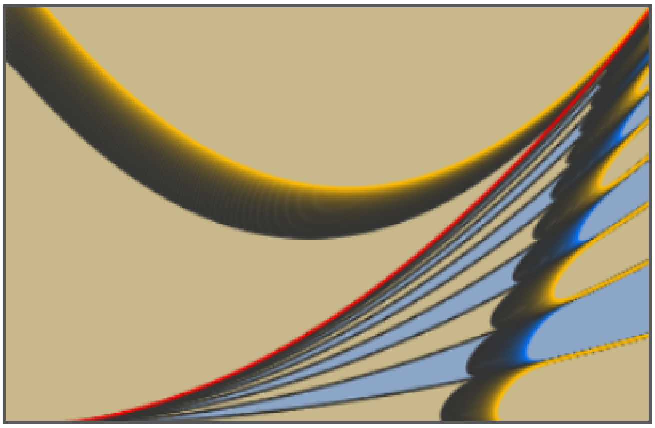
LEOPOLD-FRANZENS-UNIVERSITÄT INNSBRUCK

MASTER THESIS

# Decidability of algebra-valued tensor networks



*Author:*
Joshua GRAF

*Supervisor:*
Ass.-Prof. Gemma DE LAS CUEVAS
Univ.-Prof. Tim NETZER

Institut für Theoretische Physik

July 19, 2021

**Eidesstattliche Erklärung**

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Magister-/Master-/Diplomarbeit/Dissertation eingereicht.

| Datum | Unterschrift |
|---|---|

LEOPOLD-FRANZENS-UNIVERSITÄT INNSBRUCK

# *Abstract*

Institut für Theoretische Physik

Submitted in fulfillment of the requirements
for the degree of Master of Science

**Decidability of algebra-valued tensor networks**

by Joshua GRAF

Contrary to the growing importance of computational methods in natural sciences, there are problems proven to be unsolvable by means of algorithms. In case of a decision problem, i.e. the task of deciding whether a property is fulfilled or not, we call an uncomputable problem undecidable. In recent years, algorithmic undecidability has enjoyed growing attention as a tool in the investigation of physical problems. Motivated by an undecidable problem for 1D translationally invariant tensor networks, we introduce a generalization to algebra-valued tensor networks. It questions the membership of the network for any size in a predetermined convex cone. We call it the "moments-membership" problem due to a direct correspondence of these networks to the moments of the local tensors, i.e. the traces of its powers. We investigate some exemplary cases of this problem with main focus on polynomial algebras. As a result, we prove the undecidabilty of the problem for certain non-commutative algebras, including tensor power algebras. A more detailed discussion is dedicated to the moments-membership problem for the cone of sum of squares polynomials. Though leading to interesting insights, the reduction ansatz chosen in this work did not suffice to solve the problem. In view of the commutative case, we draw a connection to constant-recursive sequences by establishing a recursion relation for the moments. This yields a base for future work on the moments-membership problem for commutative algebras. Also, an investigation of the problem for non-commutative sum of squares using another reduction ansatz, e.g. starting from a relative of the matrix mortality problem, may be more promising.

# List of Symbols

| Symbol | Description | Notes |
|---|---|---|
| $\mathbb{N}$ | Set of natural numbers excuding 0 | - |
| $\mathbb{N}_0$ | Set of natural numbers including 0 | - |
| $[n]$ | Set of natural numbers $1, \ldots, n$ | $n \in \mathbb{N}$ |
| $\mathbb{Q}_i$ | Field extension of $\mathbb{Q}$ by the imaginary unit | $i^2 = -1$ |
| $\mathbb{A}$ | Algebraic numbers, algebraic closure of $\mathbb{Z}$ | - |
| $\overline{\lambda}$ | Complex conjugation | $\lambda \in \mathbb{Q}_i, \mathbb{A}, \mathbb{C}$ |
| $\mathcal{H}$ | (Pre-)Hilbert space | - |
| $\mathcal{H}^*$ | Dual (pre-)Hilbert space | - |
| $\lvert v\rangle$ | Dirac notation of a vector | $v \in \mathcal{H}$ |
| $\langle v\rvert$ | Dirac notation of a dual vector | $v \in \mathcal{H}^*$ |
| $\mathcal{L}(V)$ | Linear operators on $V$ | $V$ vector space |
| $\mathrm{Mat}_d$ | Quadratic $(d \times d)$-matrices | $d \in \mathbb{N}$ |
| $\mathrm{Her}, \mathrm{Her}_d$ | Hermitian operators$/(d \times d)$-matrices | $d \in \mathbb{N}$ |
| $\mathrm{PSD}, \mathrm{PSD}_d$ | Positive semidefinite operators$/(d \times d)$-matrices | $d \in \mathbb{N}$ |
| $\mathrm{supp}(f)$ | Set of arguments with non-vanishing $f$ | $f$ function |
| $\underline{X}$ | Set of indeterminates $\underline{X} = (X_1, \ldots, X_d)$ | $d \in \mathbb{N}$ |

# *Acknowledgements*

First and foremost, I would like to express my sincere gratitude to my research supervisors, Ass.-Prof. Dr. Gemma De las Cuevas and Univ.-Prof. Dr. Tim Netzer for this opportunity and their guidance through this fascinating project. When things did not seem to work out, Gemma's inspiring vision helped me to get back my focus and motivation. I am truly thankful for the knowledge and methodology she shared with me in order to conduct this research. No less, I want to appreciate Tim for his support as a mentor in mathematical issues and for his joy when it came to enlightening a physicist's mind with the beauty of mathematics. I am grateful he always had an open door and a lot of patience when my knowledge of algebra was exceeded.

I would also like to thank my colleagues for all those inspiring and enlightening discussions. I really enjoyed being part of the beautiful atmosphere in our office. Moreover, I am grateful for my family and friends, for their unconditional support throughout this project.

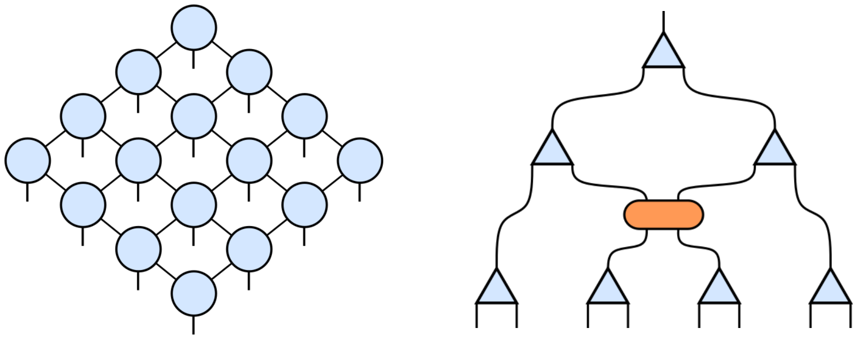
Last but not least, I would like to express my thanks to Jura for working solid as a rock even when the hours got late. Although only a coffee machine, you really knew how to support me on this journey.

# Contents

# Introduction

Hand in hand with the vast advances in computational power since the mid 20th century, we were to observe a similar growth in the importance of algorithmic methods. Computational techniques started to rapidly pull through all the natural sciences - and beyond - to become a central tool for investigation as well as generation of new knowledge. Simulations and other extensive calculations using computation units are nowadays a crucial tool in many branches of physics. However, those methods are still limited by resources such as computation time and memory. From this point of view, it is not very surprising that domains of computer science, for example computational complexity theory, experienced growing attention by the physics community. In essence, complexity theory strives for a classification of algorithms according to their requirement of certain computational resources. Beside that, there is another part that took foot in the physics discussion rather late. The subject of so-called decision problems is less interested in the precise amount of resources in need for a certain calculation, but questions the very existence of a solving algorithm instead. A decision problem is thereby usually concerned with binary questions that are answered by "Yes" or "No". That there are indeed problems unsolvable by algorithms was perceived early on. The first of this kind was posed in vicinity of Hilbert's program under the name "*Entscheidungsproblem*" in 1928 [16]. It was dedicated to the question whether an algorithm could decide over the deducability of logical statements in a specific set of axioms. In the following decades, the landscape of these so-called undecidable problems slowly became richer in appearance. These days, undecidable problems were formulated in various areas of mathematics [28]. Moreover, undecidable problems recently experienced growing attention by the physics community [6, 2]. We will show special interest in its occurrence in [8], where an undecidable problem is used as a tool to show non-existence of certain structures.

A central part of the paper just mentioned are so called Tensor Networks. The idea of those may be traced back to the early 70's, when R. Penrose introduced his notion of "abstract tensor systems" [26]. We can think of Tensor Networks as a tool to describe an object of a composite system in a decomposition to local degrees of freedom. The developements of Quantum Computational Networks by D. Deutsch [10] and R. Feynman [12] applied a related diagramatic concept to the theory of quantum circuits in 1987/88. Yet it took another two decades for its full potential to be discovered. Most remarkably, the idea of tensor network found a much broader audience, with applications far beyond quantum circuit theory. In various fields, Tensor networks led to significant developements. In quantum information theory as



(A) PEPS network (B) MERA network

FIGURE 1: Prominent examples of tensor networks

well as condensed matter physics, tensor networks are well known in their appearance as matrix product states (MPS) [1] and projected entangled pair states (PEPS) [33]. The latter network is illustrated in figure (1a). In the theory of many-body physics, the celebrated Multiscale Entanglement Renormalization Ansatz (MERA) [34, 11], as illustrated in figure (1b), allowed for new insight in renormalization group methods. The MERA network found further application in AdS/CFT-correspondence [24] and the construction of spacetime emerging from entanglement [13]. These relations turn out fruitful for our understanding of entanglement in large systems [18]. There are countless other branches of physics that benefit from the developements in tensor network theory, but the success story of tensor networks was not meant to stop at the boundaries of physics. Instead, it started to influence other disciplines such as machine learning [31] and language processing [27].

In this thesis, we will discuss some specific decision problems for tensor networks. We will focus on translationally invariant networks with values in a general algebra instead of the quantum mechanical operator-algebra. The first chapter is devoted to a more extensive presentation of the different domains present in this thesis. We will begin with an introduction to some branches of complexity theory starting with Turing machines as an abstract articulation of algorithms. After a short revision of some aspects of quantum theory, we will discuss the undecidable problem for tensor networks presented in the already mentioned paper [8]. These results will accompany us for the rest of the thesis - both as a motivation as well as a starting point for some proofs of undecidability. Chapter 1 ends with an introduction to algebras in genearal followed by a closer discussion of polynomial algebras. This may be seen as a preperation to the following chapter, which incorparates the investigation of the generalization of the tensor network decision problem to general algebras. We will discuss some special cases focused on polynomial algebras and establish some results of undecidability for non-commutative algebras. In a seperate treatment of commutative algebras, we establish a recursion formula which restricts the significant degrees of freedom. This relation suggest an investigation of the problem from the perspecive of constant-recursive sequences. Moreover, an investigation of non-existance theorems induced by undecidability similar to [8] remains to be done.

# Chapter 1

# Computational Complexity Theory and Tensor Networks

## 1.1 Decision problems in computational complexity theory

This section is devoted to an introduction of the basic concepts of computability and the different notions of undecidability. We will shortly discuss the Halting problem, which was historically one the first problem to be shown undecidable. Another decision problem will be presented in the following sections. The latter shall be the foundation of and the motivation behind this work. Similarly to the introduction, the following discussion will incoperate some historical notes.

### 1.1.1 Paradoxical self-reference and axiomatic undecidability

In contrast to the rest of this thesis, where we mainly focus on so-called algorithmic undecidability, we shall begin our journey with some related concepts: paradoxical self-reference and axiomatic undecidability. This should not be seen as a detailed introduction to this domain but more as a gentle getting-to-know before diving into the broad expanse of the field of undecidability.

Before we step to axiomatic undecidability, we shall discuss the concept of paradoxical self-reference. It will help us to understand both sides of undecidability more clearly. Moreover, it will be used later-on as the central element of Cantor's diagonal argument. We start the discussion with Russell's paradox, also known as the Barber's paradox. It was stated by B. Russell in 1918 to explain a similar paradox in set theory [30]:

> *"A barber is one who shaves all those, and only those, who do not shave themselfs. Does the barber shave himself?"*

When trying to answer this question, one finds oneself confronted with the problem that each answer automatically implies the contrary. When we express it in a mathematical system, the property "shaves" sets two objects in relation by assignment of a truth value, i.e. the expression "$x$ shaves $y$" is either true or false. The barber is introduced to the set of persons $P$ by the definition

$$x \text{ is a barber} :\Leftrightarrow \forall y \in P : (x \text{ shaves } y \Leftrightarrow \neg(y \text{ shaves } y))\,.$$

Given such a barber $x$, there seems to be an inconsistancy concerning the truth value of the expression

$$x \text{ shaves } x.$$

The root of this issue is located in the definition of the barber as it includes a statement about the barber her- or himself. It is the universal quantifier that also incorpates the barber resulting in

$$x \text{ shaves } x \Leftrightarrow \neg(x \text{ shaves } x)$$

being a true statement for any barber $x$. The truth value of "$x$ shaves $x$" seems to be ill-defined. However, this does not mean that there is an inconsistancy in this system but simply that the predicate "*is a barber*" is trivial. More specifically, there does not exist any person that satisfies the requirements of being a barber. The crucial point in this example is the appearance of a paradoxical statement due to self-reference. If we were to exclude the barber in her/his definition by demanding

$$(x \text{ shaves } y \Leftrightarrow \neg(y \text{ shaves } y))$$

for other individuals only, we would not run into a contradiction ultimately. There are various similar examples of such paradoxes which are based on the principle of self-references, for example the liar's paradox:

*"I am a liar."*

Well-known mathematical results based on paradoxical self-reference are Cantor's proof of the uncountability of the reals as well as Russell's set paradox. The latter was used to highlight an inconsistency in an early form of the axioms of set theory. The issue was later resolved by a modification of the axiom that describes how a set can be defined via a property.

In contrast to paradoxical self-reference, which aims to highlight an inconsistency or contradiction, the nature of axiomatic undecidability concerns the very provability of a statement. More specifically, an axiomatically undecidable statement is one that can neither be proven nor negated in a given axiomatic system. Thus, this may be seen as a synonymous to axiomatic independency. A well-known example is the axiom of choice, which is axiomatically undecidable in the Zermelo-Fraenkel axiom system of set theory.

While the examples so far were more of the pedagogical type, we now want to discuss probably one of the most striking results concerning axiomatic undecidability. It is the result of Gödel's observations in the early 20th century, during which time the axiomatization of mathematics was sophisticatedly driven by mathematicians around David Hilbert [37]. The main goal of Hilbert's program was the full unification of the different branches of mathematics under a single set of fundamental axioms and to show their consistency explicitly. Mathematical theories were reinvented in the way they are taught in universities nowadays. Going along with this intention was the very belief that such a complete description of mathematics was possible at all. This, however, turned out to be a losing game when Kurt Gödel published his incompleteness theorems of axiomatic systems. Gödel showed [14] that each sufficiently complex system suffers either of incompleteness or inconsistency. Moreover, and even more dooming for Hilbert's program, Gödel's second incompleteness theorem states that even if such a system is consistent in its axioms, this consistency is an unprovable statement. In other words, from a certain complexity level upwards, a system gets incapable of proving its own consistency [14, 37].

We will soon get back to another example of paradoxical self-reference in the vicinity of algorithms. This will provide an important example in the intersection of axiomatic and algebraic undecidability. To this end, we shall first introduce the notion of Turing machines as an abstract implementation of algorithms.

### 1.1.2 Algorithms and Turing machines

As mentioned before, we will set our focus on algorithmic undecidability - a domain of theoretical computer science dealing with the computability of problems. More precisely, we question the existence of an algorithm able to solve a specific problem. The intuition of an algorithm is an instruction scheme following certain rules to process an input to an output value. Consider for example the famous Gauss algorithm, a procedure allowing us to transform any input matrix in row echelon form. This algorithm can be defined on a set of basic rules, a tool kit containing basic arithmetic and some composition rules such as if and do-while commands. Another example is illustrated in figure (1.1) where we find an instruction protocol to calculate the modulo of two numbers. Algorithmic undecidability basically denotes the non-existence of such an algorithm to solve a specific problem - given a clear set of rules of course.

There are many ways of putting the idea of an algorithm in a dress of mathematical ideas. Fortunately, it turns out that there is some ambiguity in the precise description of an algorithm and various different ansatzes all lead to the same notion of computability. Each of these descriptions have their own strengths and weaknesses. To begin with, we will focus on a single model, the Turing machine, which was introduced by the mathematician Alan Turing in 1936. We chose this description due to its close intuition to classical

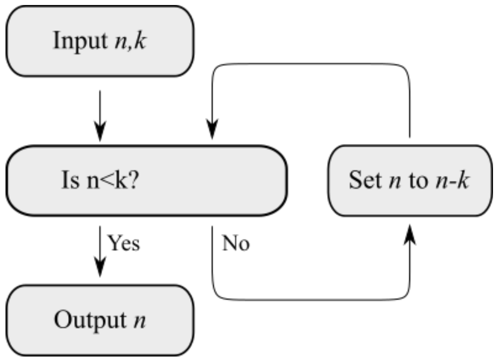



FIGURE 1.1: Intuitive form of an algorithm for the calculation of the remainder of a devision.

computers but introduce recursive functions as a more axiomatic model lateron. Based on these abstract devices, we will introduce the notion of Turing computability, or simply computability, as this description is equal in a bigger class of automata. See [22], [17] or the lecture notes of M. Wolf [35] for a broader overview of different descriptions.

A crucial point when talking about abstract automata is the encoding of information. An encoding is usually done in a certain set of symbols that are put together in strings forming words.

**Definition 1** (Alphabet and words)**.** *An **alphabet** is a non-empty set $\Sigma$ of symbols. A **word** $\omega$ of length n in an alphabet $\Sigma$ is a tuple $\omega = (\omega_1, \ldots, \omega_n) \in \Sigma^n$. We denote the set of words in $\Sigma$ of arbitrary lenght by $\Sigma^*$.*

To get a feeling for this, let us think of a classical computer. A CPU typically works with two different voltage levels representing an "on"-state "1" and an "off"-state "0". These two levels represent the alphabet the automata is working with: the binary alphabet $\Sigma_{\text{bin}} = \{0, 1\}$. Due to our physical characteristics, namely possessing ten fingers, humankind tends to use another alphabet to encode numbers. We typically use the decimal representation of numbers expressed in the alphabet

$$\Sigma_{\text{dec}} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

The representation of any natural number $n \in \mathbb{N}$ in decimals can be seen as an encoding of the information $n$ in the alphabet $\Sigma_{\text{dec}}$. The notion of computability is insensitive to the precise amount of symbols in the alphabet as long as it stays finite.

The idea of a Turing machine is that of an idealized automaton with an arbitrary but finite amount of resources. It essentially consists of two parts. On the one hand, there is an arbitrarily extended tape of cells, each containing a symbol of a so-called tape alphabet. On the other, it possesses a reading/writing head with a set of internal states $Q$. This is illustrated in figure (1.2), where the head in the internal state $q \in Q$ is represented by the gray triangle. One may think of this picture as a representation of the state of the Turing machine at an instance of time.

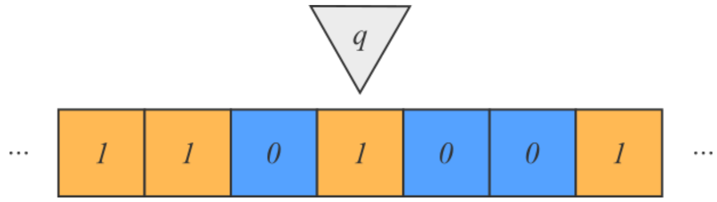



FIGURE 1.2: Schematic representation of a Turing machine with binary alphabet by its tape and reading head (gray) in the internal state $q$. The dots at both ends of the tape indicate its arbitrary extension.

Since a Turing machine describes some kind of process, there has to be dynamics to manipulate the input to the respective output. This is given by an instruction protocol, which determines the next step of the Turing machine based on its current state. The automata is thereby only allowed to use the tape-data at the position of the reading head. At each step, the Turing machine may change the symbol on the tape at the head's position as well as its internal state. At the same time, to provide mobility of the head along the tape, it can move either one cell to the left or to the right. There are some alternative definitions for a Turing machine with slight modifications. However, the notion of computability will again be insensitive to those changes.

The instruction protocol may be defined by a map. It takes the symbol on the tape at the head's position together with the internal state as arguments and returns a new symbol to write on the tape, a new internal state as well as the direction the head is moving. Putting everything together we now have collected the data needed to define such a Turing machine.

**Definition 2** (Turing machine)**.** *A **Turing machine** $T = (\Sigma, \Box, Q, q_i, q_f, P)$ is a collection of*

- *a finite **tape alphabet** $\Sigma$ with an **empty symbol** $\Box \in \Sigma$,*
- *a finite set of **internal states** $Q$ with an **initial state** $q_i \in Q$ and a **final state** $q_f \in Q$ as well as*
- *an **instruction protocol** $P : \Sigma \times (Q \setminus \{q_f\}) \to \Sigma \times Q \times \{R, L\}$.*

Although we have obtained the abstract definition of a Turing machine, we still did not discuss its operation method in full extent. What is missing is how it starts and how it stops operating. The Turing machine needs an input encoded in the tape alphabet without empty symbol, i.e. a word of finite lenght in the alphabet $\Sigma \setminus \{\Box\}$. The initial state of the tape is the input word embedded in a tape of empty symbols. The reading head is thereby positioned at the first symbol of the input word with the initial state $q_i$ assigned to. From this starting configuration of a Turing machine, it evolves as described above according to the instruction protocol $P$ until the internal state arrives at the final state $q_f$. We say the Turing machine **halts** for the respective input if it arrives at this state after finitely many steps. Note that it is not guaranteed that the automaton halts for a specific input. If it does, the word on the tape starting from the head's position to the first empty symbol is its output.

An important detail about Turing machines is that for a given input they do not have to halt after a finite amount of steps. As a consequence, a Turing machine $T$ cannot be associated with a function $F_T : \Sigma^* \to \Sigma^*$. Instead, it defines a so-called partial function.

**Definition 3** (Partial function)**.** *A **partial function** $f : X \nrightarrow Y$ is a function $f : D_f \to Y$ with $D_f \subseteq X$, i.e. a function that does not have to be defined on each element of $X$.*

One may think of a partial function as one that may map onto the value "*undefined*". The partial function $f_T : \Sigma^* \nrightarrow \Sigma^*$ associated to a Turing machine $T$ is defined to return the output of $T$ for arguments where the Turing machine halts. The necessity for partial functions is based on $T$ not halting on all input values. In the following, we will abbreviate $T(\omega) := f_T(\omega)$ for simplicity of notation.

To get a feeling for such Turing machines, let us discuss a simple example. We want a Turing machine describing the successor function

$$\begin{aligned} s : \mathbb{N}_0 &\to \mathbb{N}_0 \\ n &\mapsto n + 1. \end{aligned}$$

We use a binary tape alphabet $\Sigma_{\text{bin}}$ with $0$ as empty symbol and encode a input $n \in \mathbb{N}$ as word of $n + 1$ times the symbol $1 \in \Sigma_{\text{bin}}$. This is know as the **unary encoding**. The task of the Turing machine is thus to add one times the symbol 1 at the end of the input word followed by moving the head back to the beginning. This can be implemented with 3 internal states: the initial state $q_0$, an intermediate state $q_1$ and the final state $q_2$. The instruction protocol is defined by

$$\begin{aligned} (q_0, 1) &\mapsto (q_0, 1, R) \\ (q_0, 0) &\mapsto (q_1, 1, L) \end{aligned}$$

for the initial state. This moves the head along the tape till it hits the first empty symbol, which is changed to a 1 while the internal state switches to the intermidiate state $q_1$ and the head moves left. In the state $q_1$, we let the head move back to the left along the tape till it hits an empty symbol again. At this point, it moves a last step right and changes to the final state:

$$\begin{aligned} (q_1, 1) &\mapsto (q_1, 1, L) \\ (q_1, 0) &\mapsto (q_2, 0, R). \end{aligned}$$

The resulting word consists of $n + 2$ times the 1 symbol which encodes the natural number $n + 1$. Such an instruction protocol can be illustrated by a graph where the internal states are the vertices and the possible steps are the

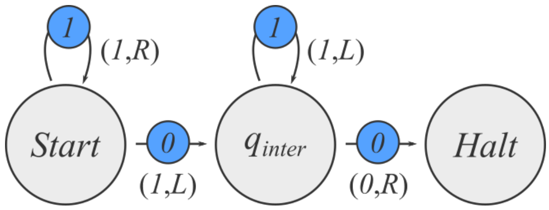



FIGURE 1.3: Visualization of an algorithm by the graph corresponding to the instruction protocol. Big circles illustrate internal states while the small ones indicate the tape read-value leading to the corresponding transition. Write-value as well as head movement direction are added in brackets.

edges. Note that the encoding map

$$\begin{aligned} \mathrm{ec} : \mathbb{N}_0 &\to \Sigma_{\mathrm{bin}}^* \\ n &\mapsto \mathrm{ec}(n) := \underbrace{11\ldots1}_{n+1} \end{aligned}$$

as well as the (partial) decoding map

$$\begin{aligned} \mathrm{dc} : \Sigma_{\mathrm{bin}}^* &\to \mathbb{N}_0 \\ \underbrace{11\ldots1}_{n+1} &\mapsto n \end{aligned}$$

are necessary to connect the Turing machine working on the alphabet $\Sigma_{\mathrm{bin}}$ to the in- and output in the natural numbers.

### 1.1.3 Computability and decidability

Now that we have found an abstract way to describe an algorithm by Turing machines, it is a natural question to ask for their limitations. In other words, we may ask whether any function can be simulated by a Turing machine. To deal with this question, we have to fix an alphabet and an encoding as well as a decoding. Specially the latter two have great influence on the resulting notion of computability. Since we will work exclusively with functions that have a natural relation to the natural numbers, we will restrict the following discussion to functions of the form $f : \mathbb{N}_0^k \to \mathbb{N}_0^m$ with a $n$-ary encoding in an alphabet

$$\Sigma = [n] \cup \{\perp, \square\}.$$

The symbol $\perp$ is used as a spacer to separate the $k$ input numbers in the encoding. For example, a rational number $q \in \mathbb{Q}$ may be represented by a triple of natural numbers $a, b, c \in \mathbb{N}$ with

$$q = \frac{a-b}{c}.$$

The encoding of $q$ then is obtained by using the $n$-ary encodings $\omega_a, \omega_b, \omega_c$ of the natural numbers together, which yields

$$\mathrm{ec}(q) = ''\omega_a \perp \omega_b \perp \omega_c''.$$

The decoding is constructed in a similar manner. This way, by defining computability for functions on cartesian powers of the naturals, we can choose a single pair of encoding and decoding for a big number of problems with an input set related to natural numbers.

**Definition 4** (Computable function)**.** *A partial function $f : \mathbb{N}_0^k \to \mathbb{N}_0^m$ is called (Turing) computable if there is a Turing machine $T$ that halts on* $\mathrm{dom}(f)$ *only and such that*

$$\begin{tikzcd} \mathbb{N}_0^k \arrow[r, "\mathrm{ec}"] \arrow[d, "f"] & \Sigma^* \arrow[d, "T"] \\ \mathbb{N}_0^m & \Sigma^* \arrow[l, "\mathrm{dc}"] \end{tikzcd}$$

*is a commutative diagram.*

In fact, it was already shown in the early times of the abstract theory of automata that the amount of computable functions is evanescent compared to the uncomputable ones. This is immediate to see if we employ some cardinality arguments. As the description of a Turing machine is finite, one can show that the set of Turing machines is an enumerable one. On the other hand, the set of functions $f : \mathbb{N}_0 \to \{0,1\}$ for example has the same cardinality as the power set of the naturals. This is due to a bijection between subsets $A \subseteq \mathbb{N}$ and characteristic functions $\chi_A : \mathbb{N}_0 \to \{0,1\}$ defined by

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

It is a well-known fact that the cardinality of a set is always smaller then the cardinality of its power set. We can conclude that only an enumerable subset of all functions can be modelled by a Turing machine. In other words, almost every function is uncomputable by a Turing machine. Still, uncomputability of a function can be used as a tool providing interesting insight in physical problems, see for example [2, 6, 8].

The concept of computability can be extended to sets with a reasonably relation to a Cartesian power of the natural numbers, for example the rationals as treated above. We will typically deal with matrices over an enumerable subfield of $\mathbb{C}$, i.e. a Cartesian power of the respective field. Note that we can also adapt the definition of computability to functions with a restricted domain $A \subsetneq \mathbb{N}^k$. However, it is immediate that a function with a finite domain is always computable, since we can build an algorithm that comprises all possible inputs with their respective output. It would be programmed to read the input and just write the right output. Such a Turing machine would not use any inherent structure of the problem and would still have a finite description. Thus, our interest focuses on infinite enumerable sets.

In this thesis, we will focus on the computability of a special kind of functions that declare a binary property to the elements of an enumerable set. More specifically, these functions can be thought of as characteristic functions of the property. For each element, they return the value 1 if it has the property and 0 if it does not. An example of such a property that we have seen earlier is the predicate "*is a barber*". To give a more fitting example in the virtue of computability, we may think of the property "*is prime*" on the natural numbers. We shall soon get back to this and show its computability explicitly.

**Definition 5** (Decision problem)**.** *Let $\mathcal{I}$ be a (enumerable) set and*

$$\mathcal{Q} : \mathcal{I} \to \{0, 1\}$$

*a map describing a binary property of elements in $\mathcal{I}$. We call the question of computability of $\mathcal{Q}$ a **decision problem**. A decision problem $\mathcal{Q} : \mathcal{I} \to \{0, 1\}$ is called*

1. ***recursively enumerable** iff there is a Turing machine $T$ that halts for all $\mathcal{Q}^{-1}(\{1\})$ and satisfies $T(a) = \mathcal{Q}(a)$ whenever it halts and*

2. ***co-recursively enumerable** iff there is a Turing machine $T$ that halts for all $\mathcal{Q}^{-1}(\{0\})$ and satisfies $T(a) = \mathcal{Q}(a)$ whenever it halts.*

*We call the decision problem **decidable** whenever $\mathcal{Q}$ is computable.*

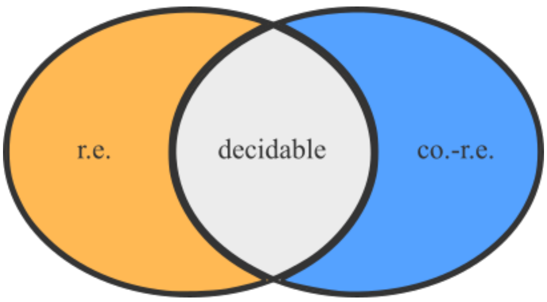



FIGURE 1.4: The set of decidable problems illustrated as the intersection of recursively enumerable and recursively co-enumerable problems.

Essentially, a decision problem is recursively enumerable if there is an algorithm able to determine whether an instance has the property, but not necessarily whether it does not have it. Thus, it is only required to halt on the instances $\mathcal{Q}^{-1}(\{1\})$. The same is true for recursively co-enumerable problems with the no-instances $\mathcal{Q}^{-1}(\{0\})$. As one may suspect, a decision problem is decidable if and only if it is recursively enumerable as well as recursively co-enumerable as illustrated in figure (1.4). We will typically define such a decision problem for a property in a sentence of the following form:

*Given $A \in \mathcal{I}$, decide whether $A$ has the property $\mathcal{Q}$.*

The intuitive way to show the decidability of a decision problem is the explicit construction of an algorithm that solves the problem. However, it is a quite ungrateful task to construct an explict Turing machine for more complex properties. It is more practical to have a construction kit of building blocks together with some composition rules. The following theorem gives a crutial footing for such an approach.

**Theorem 6.** *The set of computable functions is closed under composition of functions.*

Moreover, projection and algebraic operations on the naturals, i.e. addition as well as multiplication, can be shown to be computable, see for example [35]. We can now use these results to construct other computable functions that rely on algebraic operations, for example the standard scalar product or the matrix multiplication. However, the full class of computable functions is larger than the set we reach by this construction.

We can solve this issue by introducing two additional constructions that can be shown to be Turing computable - the primitive recursion as well as the minimalization operation. From a programmers point of view these may be seen as a FOR- and a DO-WHILE-operation respectively. The discussion of those will closely follow the line of M. Wolf in his lecture notes [35]. The primitive recursion is a map of functions. It demands two functions $f : \mathbb{N}^k \to \mathbb{N}$ and $g : \mathbb{N}_0^{k+2} \to \mathbb{N}_0$ as arguments in return for another, $\mathrm{Pr}[f,g] : \mathbb{N}_0^{k+1} \to \mathbb{N}_0$. The latter is satisfying the initial condition $\mathrm{Pr}[f,g](0,x) = f(x)$ as well as the recursion relation

$$\mathrm{Pr}[f,g](n+1,x) = g(\mathrm{Pr}[f,g](n,x),n,x)$$

for all $x \in \mathbb{N}_0^k$. The first argument of $\mathrm{Pr}[f,g]$ yields the number of recursions of applying $g$. If the functions $f$ and $g$ are computable then so is the function $\mathrm{Pr}[f,g]$, see for example [35]. The class of functions we can construct by means of algebraic operations on the naturals together with the primitive recursion construction are called primitive recursive functions. They yield a proper subset of the full Turing computable functions and are typically defined axiomatically. The basic primitive recursive building blocks are

(PR1) the zero function $\mathbf{0} : n \mapsto 0$,

(PR2) the projections $p_i^k : (n_1, \ldots, n_k) \mapsto n_i$, for $k \in \mathbb{N}$ and $i \in [k]$, as well as

(PR3) the sucessor function $s : n \mapsto n+1$.

Those basic PR functions together with the two axioms concerning rules of building new PR functions, i.e.

(PR4) let $g : \mathbb{N}_0^m \to \mathbb{N}_0$ and $f_1, \ldots, f_m : \mathbb{N}_0^n \to \mathbb{N}_0$ be PR, then the composition $g(f_1, \ldots, f_m) : \mathbb{N}_0^n \to \mathbb{N}_0$ is PR and


FIGURE 1.5: The set of natural numbers devided in primes (amber) and non-primes (blue) as an example of a decidable problem.

(PR5) let $f : \mathbb{N}_0^k \to \mathbb{N}_0$ and $g : \mathbb{N}_0^{k+2} \to \mathbb{N}_0$ be PR, then the primitive recursion $\mathrm{Pr}[f, g]$ is PR,

define an axiomatic system for primitive recursive functions. This establishes a connection between axiomatic and computational undecidability by putting computability in an axiomatic system.

The full set of Turing computable functions can be obtained by adding one more ingredient, namely the minimization operation

$$\mathrm{Mn}[f](x) := \begin{cases} n & \text{if } f(n,x) = 0 \wedge (\forall n' < n : f(n',x) \in \mathbb{N}) \\ \text{undefined} & \text{elsewise} \end{cases}$$

for a function $f : \mathbb{N}_0 \times \mathbb{N}_0^k \to \mathbb{N}_0$. Thus, if defined, the minimalization gives the smallest natural number such that $f$ under conditions $x \in \mathbb{N}_0^k$ is vanishing. This allows for the construction of conditional functions, for example a WHILE-loop. Again, one can show that for a computable function $f$, the minimization $\mathrm{Mn}[f]$ is again computable. One can show that the primitive recursive functions together with the minimization operation provide the full set of Turing computable functions.

**Example: Prime numbers.** As an example of a decidable problem we may consider the task of determining whether a natural number $n \in \mathbb{N}$ is prime or not. Clearly, the set of instances is enumerable as it is the naturals themselves.

**Problem 1** (Prime)**.** *Given a natural number $p \in \mathbb{N}$, decide whether $p$ is a prime number.*

We will set up a rather primitive algorithm to show the decidability of this problem. The principle idea is to check the remainder of the devision of $n$ with all numbers with square less-equal $n$. A central element of this construction will be the remainder function

$$\begin{aligned} r : \mathbb{N}_0 \times \mathbb{N} &\to \mathbb{N}_0 \\ (a,b) &\mapsto a \mod b \end{aligned}$$

which we shall show to be computable. We already gave an intuitive algorithm to calculate the remainder in the beginning of this section, see figure (1.1). Now we want to translate it to the language of recursive functions. The main part of this instruction is essentially a while loop. It subtracts the second argument from the first one as many times as possible, which will be achieved by a minimization operation. There are however some basic operations that have to be defined first. To be precise, we have to construct addition, multiplication as well as subtraction of natural numbers. The addition is well defined by $+(0,x) := p_1^1(x) = x$ and $+(n+1,x) := s(+(n,x))$ which corresponds to a primitive recursion with $f \equiv p_1^1$ and $g \equiv s \circ p_1^3$. In the same manner, we construct the multiplication by $\cdot(0,x) := 0$ and $\cdot(n+1,x) := \cdot(n,x) + x$, i.e. $f \equiv \mathbf{0}$ and $g \equiv +(p_1^3, p_3^3)$. Similar to the former two functions, the substraction is a primitive recursive function, but we first define the predecessor function pre as an auxiliary function. This primitive recursive function is obtained by $\mathrm{pre}(0) = 0$ and $\mathrm{pre}(n+1) = n$. The subtraction is then constructed with some similarity to the addition by $\dot{-}(x,0) := x$ and $\dot{-}(x,n+1) := \mathrm{pre}(\dot{-}(x,n))$. Note that we have swapped the arguments concerning the primitive recursion for the sake of better readability. Moreover, the dot's duty is to highlight the subtraction of natural numbers which vanishes if the first argument is greater than the second one.

In the next step, we will use these basic functions as a toolkit for the construction of the remainder function. As final result, we will obtain the recursive function

$$\begin{aligned} r(a,b) : &= \dot{-}(p_1^2, \cdot(\div, p_2^2))(a,b) \\ &= a \dot{-} (a \div b) \cdot b \end{aligned}$$

where $\div$ denotes the division of natural numbers, i.e. the largest $n \in \mathbb{N}$ such that $a > n \cdot b$. This will be the only function that uses the minimization operation. More precisely, it is obtained as the minimization of the function $f(n,a,b) = a \dot{-} (n+1) \cdot b$, i.e.

$$\div(a,b) : = \mathrm{Mn}[f](a,b).$$

To this point, we have established the remainder as a recursive function and thus Turing computable.

In the last step, we will show that the property IsPrime is indeed computable. To achieve this, we shall use the primitive recursion construction one last time to define the function $\mathrm{div}(p)$ which yields the number of divisors of $p$; the trivial cases 1 and $p$ excluded. We begin with an auxiliary function defined by $f(0,p) = 0$ and $f(n+1,p) = f(n,p) + (1 \dot{-} r(p, s(n)))$. The term $1 \dot{-} r(p,s(n))$ yields 1 if $p$ is divided by $n+1$ and vanishes otherwise. Using this recursive function, the number of non-trivial divisors is given by $\mathrm{div}(p) = f(\mathrm{pre}(\mathrm{pre}(p)), p)$. Last but not least, we truncate all non-vanishing values to 1 to obtain the property $\mathrm{IsPrime}(p) = 1 \dot{-} (1 \dot{-} \mathrm{div}(p))$. We conclude that IsPrime is decidable.

### 1.1.4 Undecidability - Cantor's Argument and Reductions

In this section we focus on showing undecidability of a decision problem. The main tools we will discuss are Cantor's argument, also known as the diagonalization procedure, on the one hand and the reduction procedure on the other. The former proves the undecidability from scratch by using self-reference and negation to form a contradiction similar to the one seen earlier in the Barber's paradox. The latter on the contrary uses structure preserving maps, i.e. homomorphisms of decision problems called reductions, to broadcast the undecidability of one problem to another.

We will start the discussion with Cantor's argument since it does not rely on any knowledge of another decision problem. It is an approach that exploits a form of self-reference concerning Turing machines to generate a contradiction to computability. Cantor himself used this method to show the non-enumerability of the real numbers. The crucial structure for Cantor's argument to work out is some kind of self-reference in the system which, however, has many different faces and can occur in various ways. A collection of such instances can be found in [36]. We shall treat Cantor's argument on an important example, the Halting problem, which was proven to be undecidable by A. Church and A. Turing independently. This was the very first problem to be proven algorithmically undecidable.

**Problem 2** (Halting problem). *Given a description of a Turing machine $T$ as well as an input $x$, decide whether $T$ halts on the input $x$.*

Essentially, this problem is asking for the existence of a Turing machine able to determine whether any Turing machine halts for any input. For this purpose, the finite description of Turing machines is crucial since they have to be fed as an input, thus as a natural number. There are different ways to encode a Turing machine $T$ in a natural number. If such a system is fixed, the corresponding number is usually called the **Gödel number** $\ulcorner T \urcorner$ of the automaton $T$. With this in mind, we will state the famous proof of the undecidability of the Halting problem by paradoxical self-reference [36]. The proof will follow the line of Cantor's diagonal argument. We denote by $G : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbf{2}$ the function that specifies whether a Turing machine halts on a specific input. Note that the function $G(\lfloor T \rfloor, \bullet)$ is total since - for any input - the Turing machine $T$ either halts or not. In the following, we will assume that $G$ is computable and construct a paradoxical statement based on this.

Following Cantor's argument, we construct a Turing machine $g$ according to the diagram

$$\begin{tikzcd} \mathbb{N} \arrow[r, "g"] \arrow[d, "\Delta"'] & \mathbf{2} \\ \mathbb{N} \times \mathbb{N} \arrow[r, "G"] & \mathbf{2} \arrow[u, "\alpha"'] \end{tikzcd}$$

where $\Delta(n) := (n, n)$ is the diagonal function and $\alpha$ is a special partial function. The latter is defined by $\alpha(0) = 1$ and $\alpha(1)$ undefined. This can be realized by leading the head in an endless loop if $G$ returns the value 1. This way, a Turing machine corresponding to $\alpha \circ G$ halts if and only if the input automata is not terminating. Note that the construction $\alpha \circ G \circ \Delta$ yields a Turing machine only because of the assumtion of computability of $G$. Since $G(\lfloor g \rfloor, \bullet)$ is total, it is specifically defined for the input $\lfloor g \rfloor$. By definition of $G$, we find that $G(\lfloor g \rfloor, \lfloor g \rfloor) = 1$ if and only if $g$ halts on $\lfloor g \rfloor$, i.e. $g(\lfloor g \rfloor) = 1$. Due to the precise construction of $g$ however, this further implies the equivalence

$$G(\lfloor g \rfloor, \lfloor g \rfloor) = 1 \Leftrightarrow G(\lfloor g \rfloor, \lfloor g \rfloor) = 0.$$

This strongly reminds of the situation in the barbers paradox where the existence of a barber resulted in a similar statement. Again, this contradiction is a consequence of self-reference, this time in form of the Gödel number, which allows a Turing machine to be its own input. Since the computability of $G$ was the only assumption made in this discussion, we conclude that the halting problem 2 is undecidable.

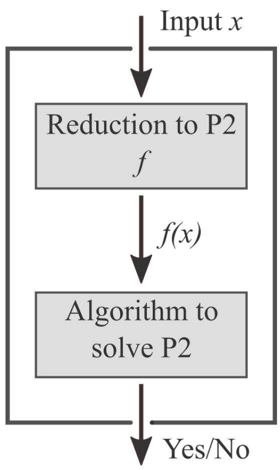



FIGURE 1.6: The algorithm constructed to solve problem 1 by using a hypothetical algorithm for problem 2.

Before we proceed with the discussion of reductions of decision problems, let us devote some words on the connection of axiomatic and algorithmic undecidability as well as the role of self-reference. The diagonal argument of Cantor may be seen as a tool that can be used to generate a contradiction if there is enough complexity. This can be applied to a variety of situations. Though originally used to show the existence of different cardinalities of infinity, it has its influence in the seemingly distant field of decidability. We refer to [36] for an overview of different problems that where tackled with this technique.

Concerning undecidability, from a technical perspective, the two notions we introduced are of a fundamentally different nature. On one hand, we got the question of axiomatic independence, i.e. formal non-deducability of a single statement or its negation in a specified set of axioms. This is technically established by formal proof theory, which took a central part in Gödel's work on incompleteness. In essence, this formalizes the way we prove statements in an axiomatic system by representing statements and proofs by words in a finite alphabet. On the other hand, algorithmic undecidability is concerned with non-computability, the lack of an algorithm capable of reflecting a property, which may be seen as a family of statements, one for each input. However, they in fact share an intimate relationship. Essentially, any algorithmically undecidable property $\mathcal{Q}$ needs an infinite amount of axiomatically undecidable statements. To see this, assume that each statement of the form "$\mathcal{Q}(x) = 1$" would be axiomatically decidable, i.e. for any input $x$ there is a proof for the property to be true or false. In this case, we could design an algorithm that runs through all sentences in the proof language

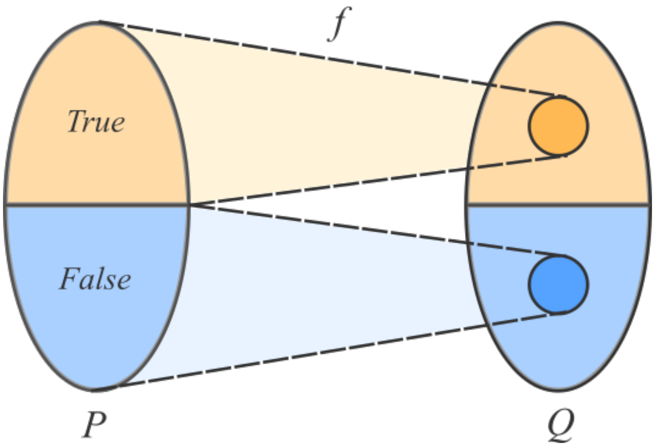



FIGURE 1.7: Schematic representation of a reduction of the binary decision problem $P$ to $Q$ by the (computable) function $f$. As illustrated, the reduction has to match both the true as well as the false instances.

checking whether they prove "$\mathcal{Q}(x) = 1$" or its negation. Due to the assumption that any of these statements are axiomatically decidable, the algorithm is guaranteed to halt - a contradiction to the assumption of algorithmic undecidability.[1] The same argument can be applied for a finite amount of axiomatically undecidable instances. They can be treated separately by a straight implementation of the corresponding answer. There are more examples that suggest that though those notions of undecidability concern different objects, they are in fact deeply intertwined.

The Halting problem was historically the first instance of algorithmic undecidability. Almost all other instances of algorithmic undecidability can be traced back to the undecidability of this single problem [28]. The central tool used for later proofs of undecidability are so-called reductions, the homomorphisms of decision problems. A reduction is a computable map that relates two decision problems. Essentially, instances of one problem are mapped to instances of the other such that both have the same outcome in $\{0,1\}$. This relation can be utilized to prove the undecidability of a problem. If the related problem, call it problem 2, is decidable, we can build an algorithm that decides the original problem, call it problem 1. As illustrated in figure (1.6), this algorithm simply uses the reduction to assign the input to a corresponding input for problem 2 with the same outcome. Applying the solving algorithm to the latter input then yields the right output for the first problem. On the other hand, this means that undecidability of the original problem implies undecidability of the second one. This often allows us to determine the undecidability of a problem without having to deal with Turing machines directly.

[1]This example was stated by Joel David Hamkins on the platform MathOverflow to elaborate a question about the relation of these concepts of undecidability. See *https://mathoverflow.net/questions/130789/are-the-two-meanings-of-undecidable-related*; access date: 05.07.2021

**Definition 7** (Reduction of decision problems). *Let $(\mathcal{I}_1, \mathcal{Q}_1)$ and $(\mathcal{I}_2, \mathcal{Q}_2)$ be decision problems. We call a computable map $\varphi : \mathcal{I}_1 \to \mathcal{I}_2$ such that the diagram*

$$\begin{tikzcd}
\mathcal{I}_1 \arrow[rr, "\varphi"] \arrow[dr, "\mathcal{Q}_1"'] & & \mathcal{I}_2 \arrow[dl, "\mathcal{Q}_2"] \\
& \mathbf{2} &
\end{tikzcd}$$

*commutes a reduction of decision problems.*

The essence of this definition is visualized in figure (1.7). As mentioned earlier, a reduction transfers decidability properties from one problem to another. How this works in detail is explained in the following theorem.

**Theorem 8.** *Let $(\mathcal{I}_1, \mathcal{Q}_1)$ and $(\mathcal{I}_2, \mathcal{Q}_2)$ be decision problems and $f : (\mathcal{I}_1, \mathcal{Q}_1) \to (\mathcal{I}_2, \mathcal{Q}_2)$ a reduction. Then*

1. *$(\mathcal{I}_1, \mathcal{Q}_1)$ is decidable if $(\mathcal{I}_2, \mathcal{Q}_2)$ is decidable*
2. *$(\mathcal{I}_1, \mathcal{Q}_1)$ is recursively enumerable if $(\mathcal{I}_2, \mathcal{Q}_2)$ is recursively enumerable*
3. *$(\mathcal{I}_1, \mathcal{Q}_1)$ is co-recursively enumerable if $(\mathcal{I}_2, \mathcal{Q}_2)$ is co-recursively enumerable.*

*Proof.* In all three cases, the statement follows immediately from the composition rule of computable functions. □

Though the case of undecidability is not explicitly stated in the theorem, it can be immediately obtained by negation of the first statement. This yields

$$(\mathcal{I}_1, \mathcal{Q}_1) \text{ undecidable } \Rightarrow (\mathcal{I}_2, \mathcal{Q}_2) \text{ undecidable}\,. \tag{1.1}$$

Thus, undecidability is transfered by a reduction while decidability is pulled-back, i.e. decidability of the image problem implies the decidability of the domain problem.

A more or less trivial yet important example of a reduction is the inclusion map, as presented in the following lemma.

**Lemma 9.** *Let $(\mathcal{I}, \mathcal{Q})$ be a decision problem and $\mathcal{S} \subseteq \mathcal{I}$ non-empty. The inclusion map $\imath : \mathcal{S} \hookrightarrow \mathcal{I}$ is a reduction of $(\mathcal{S}, \mathcal{Q}|_{\mathcal{S}})$ to $(\mathcal{I}, \mathcal{Q})$.*

*Proof.* The restriction of a function satisfies $Q \circ \imath = Q|_{\mathcal{D}}$. Furthermore, the inclusion is clearly computable. □

Though simple in appearance, this lemma has important consequences. Clearly, if the problem could be solved on the full set, we could use the very same algorithm to solve it on a subset. We will later employ the negation of this statement; undecidability on a subset implies undecidability of the full problem. On the other hand, as typically fulfilled by homomorphisms of any kind, the reductions are closed under the composition of maps.

**Lemma 10** (Composition of reductions)**.** *Let* $(\mathcal{I}_1, \mathcal{Q}_1)$,$(\mathcal{I}_2, \mathcal{Q}_2)$ *and* $(\mathcal{I}_3, \mathcal{Q}_3)$ *be decision problems. Moreover, consider homomorphisms* $\varphi_{12} : (\mathcal{I}_1, \mathcal{Q}_1) \to (\mathcal{I}_2, \mathcal{Q}_2)$ *and* $\varphi_{23} : (\mathcal{I}_2, \mathcal{Q}_2) \to (\mathcal{I}_3, \mathcal{Q}_3)$*. Then the composition*

$$\varphi_{23} \circ \varphi_{12} : \mathcal{I}_1 \to \mathcal{I}_3$$

*is a reduction of* $(\mathcal{I}_1, \mathcal{Q}_1)$ *to* $(\mathcal{I}_3, \mathcal{Q}_3)$*.*

*Proof.* Consider $c \in \mathcal{I}_1$, then

$$\begin{aligned}(\mathcal{Q}_3 \circ (\varphi_{23} \circ \varphi_{12}))(c) &= (\mathcal{Q}_3 \circ \varphi_{23})(\varphi_{12}(c)) \\ &= \mathcal{Q}_2(\varphi_{12}(c)) = \mathcal{Q}_1(c).\end{aligned}$$

Since the composition of computable maps is again computable, $\varphi_{23} \circ \varphi_{12}$ is a reduction. □

## 1.2 Tensor Networks in Quantum Physics

As already mentioned in the introduction, tensor networks have become an important tool in various branches of quantum physics. We will now introduce and discuss a special kind of tensor networks which will accompany us troughout this thesis: the 1D translationally invariant tensor network. To begin this journey, let us put these networks in the phyiscal context.

To begin, we shall discuss the physical context of tensor networks and how the structure of quantum theory advantaged their developement. As mentioned in the introduction, tensor networks originated in the vicinity of quantum physics, which is due to the precise structure of composed quantum systems. We will assume our systems to be represented by the so-called Dirac-von Neumann axioms. A central element of this description is a Hilbert space $\mathcal{H}$ over the complex numbers. For simplicity and due to our focus on finite dimensional systems, we will assume that $\dim \mathcal{H} < \infty$. The Hilbert space induces both the space of states as well as the space of observables, which are special subspaces of the linear opereators $\mathcal{L}(\mathcal{H})$ on $\mathcal{H}$. In this formalism, a **state** $\rho$ of a quantum system is a positive semidefinite operator of unit trace. The **space of states** is the space of such operators

$$S(\mathcal{H}) := \{\rho \in \mathcal{L}(\mathcal{H}) \mid \rho \geqslant 0 \text{ and } \operatorname{tr}(\rho) = 1\}.$$

On the other hand, the **space of observables** is the set of self-adjoint, or similarily hermitian[2], operators acting on the Hilbert space, i.e.

$$\mathcal{O} := \operatorname{Her}(\mathcal{H}) = \{A \in \mathcal{L}(\mathcal{H}) \mid A^\dagger = A\}.$$

[2]In the present case of a finite dimensional Hilbert space, the terms self-adjoint and hermitian describe the same structure. This is why they are often used interchangable in the physical discussion of such systems. In general systems, however, self-adjointness is a stronger requirement.

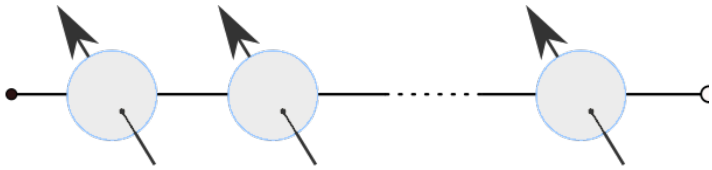


FIGURE 1.8: Illustration of the spin chain as composite quantum system. Here, the spheres indicate the local Hilbert space representing a spin system, i.e. a finite dimensional quantum system.

Our main focus is on the description of states and observables in systems equipped with symmetries. This is why we shall not be concerned with the principles of quantum dynamics, neither unitary nor projective. We just note that the time evolution of an unmeasured quantum system is generated by a special observable, namely the Hamiltonian operator $H \in \mathrm{Her}(\mathcal{H})$. This observable defines the character of a quantum system. Most importantly for our purpose, it specifies the proper symmetries of the quantum system. However, instead of working with an explicit Hamilton operator, we shall just assume specific symmetry properties. At this point, there are two crucial differences of such a system compared to a classical one. On the one side, the quatum space of states is based on a Hilbert space construction and allows for superposition of states, i.e. combinations of different observed states. On the other, more important for us, the composition of quantum systems is of a fundamentally different nature. Recall that classical systems are composed by a direct sum construction, providing a linear growth of degrees of freedom in the composite system. Quantum systems, however, show a different behaviour. A composite system is obtained by a tensor product construction of the Hilbert spaces $\mathcal{H}_A \otimes \mathcal{H}_B$. In contrast to the classical system, this leads to an exponential growth of the systems degrees of freedom, which makes it a lot harder to describe such systems. Tensor networks are a tool developed to get this issue under control. They are particularly powerful when dealing with additional symmetries, which will be the case for the states we are interested in. We will focus in this work on translational symmetries in one dimension, i.e. finite cyclic symmetries. Systems with this kind of symmetry can be decomposed in a $n$-fold copy of one and the same Hilbert space:

$$\mathcal{H}_{\mathrm{sys}} = \mathcal{H}^{\otimes n} := \underbrace{\mathcal{H} \otimes \cdots \otimes \mathcal{H}}_{n \text{ times}}.$$

A popular example of such a system is the spin chain illustrated in figure (1.8). Here, the system is described by the $n$-fold tensor product of a unitary representation of the spin group $SU(2)$. For such a system, we introduce a translation operator $T$ acting on $\psi_1 \otimes \cdots \otimes \psi_n \in \mathcal{H}^{\otimes n}$ by

$$T(\psi_1 \otimes \cdots \otimes \psi_n) := \psi_n \otimes \psi_1 \otimes \cdots \otimes \psi_{n-1}.$$

This definition extends to arbitrary vectors by linearity. It is immediate that $T$ is a unitary map and that $T^n$ is the identity map on $\mathcal{H}^{\otimes n}$. Therefore, $T$ is

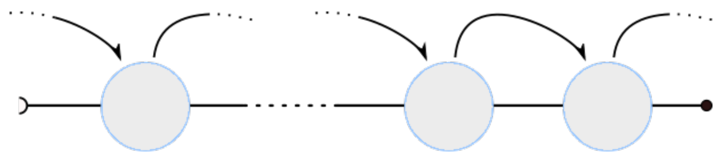


FIGURE 1.9: Illustrtion of the action of the translation operator $T$ on a spin chain. The arrows indicate the isomorphic mapping of the local Hilbert space to its respective neighbor.

a generator of a unitary $\mathbb{Z}_n$-action on $\mathcal{H}^{\otimes n}$. This action transfers to the linear operators on $\mathcal{H}^{\otimes n}$ according to

$$\mathcal{T}(A_1 \otimes \cdots \otimes A_n) := T^\dagger[A_1 \otimes \cdots \otimes A_n]T = A_n \otimes A_1 \otimes \cdots \otimes A_{n-1}$$

where $A_1, \ldots, A_n \in \mathcal{L}(\mathcal{H})$ and $\mathcal{T}$ is calligraphic to distinguish it from $T$. We call an operator $A \in \mathcal{L}(\mathcal{H}^{\otimes n})$ **translationally invariant** if it is a fixed point of $\mathcal{T}$, i.e. $\mathcal{T}(A) = A$. In the following we will be interested in translationally invariant states of the system. These are of special interest from various perspectives, especially if the Hamilton operator itself is translationally invariant. It is straight forward that the thermal states of the system, given by the Gibbs state

$$\rho(\beta) \propto \exp\left(-\beta H\right),$$

inherit this invariance due to $T^\dagger \exp(A) T = \exp(T^\dagger A T)$ for unitary $T$. The same is true for the ground state of the system if it is non-degenerate. Since these are important classes of states, we are interested in an efficient description of translationally invariant operators. Tensor networks are structures providing such an efficient way to work with operators which are invariant under an enumerable symmetry group. In the following, we will focus on the description of $\mathbb{Z}_n$-invariant states by so-called translationally invariant tensor networks.

**Definition 11** (Translationally invariant tensor network)**.** *Let $\mathcal{H}$ be a Hilbert space and $A \in \mathrm{Mat}_r(\mathcal{L}(\mathcal{H}))$. We define the **translationally invariant tensor network** $\tau_n(A) \in \mathcal{L}(\mathcal{H}^{\otimes n})$ of order $n \in \mathbb{N}$ with **local tensor** $A$ by*

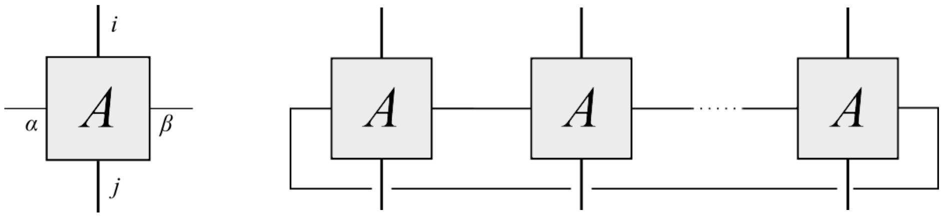
$$\tau_n(A) := \sum_{\alpha \in [r]^n} A_{\alpha_1\alpha_2} \otimes A_{\alpha_2\alpha_3} \otimes \cdots \otimes A_{\alpha_n\alpha_1} \tag{1.2}$$

*and call $r$ the **rank** or **bond dimension** of the network.*

Applying the translation operator to such a tensor network operator yields

$$\begin{aligned} T^\dagger \tau_n(A) T &= \sum_{\alpha \in [r]^n} T^\dagger [A_{\alpha_1\alpha_2} \otimes A_{\alpha_2\alpha_3} \otimes \cdots \otimes A_{\alpha_n\alpha_1}] T \\ &= \sum_{\alpha \in [r]^n} A_{\alpha_n\alpha_1} \otimes A_{\alpha_1\alpha_2} \otimes \cdots \otimes A_{\alpha_{n-1}\alpha_n} \end{aligned}$$

which corresponds to a simple renaming of the indexing. We conclude that these tensor networks are indeed translationally invariant by construction.




(A) Local tensor (B) Translationally invariant tensor network

FIGURE 1.10: Visual representation of tensor networks: a translationally invariant tensor network (B) build up by a single local tensor (A). Regarding the local tensor $\{A^{ij}_{\alpha\beta}\}$ in (A), the Greek letters indicate the virtual indices while Latin letters are used for the physical ones, i.e. the ones acting on the local Hilbert space.

A particular strength of tensor networks is the way they can be visualized. Such diagramatic approaches often turn out useful to keep an overview over complex structures. The most famous example of such a procedure certainly is Feynman's approach to perturbative quantum field theory and its adaptations. For tensor networks, such a visualization is achieved by representing the local tensors by boxes with a leg for each index. Contraction is then indicated by a connection of two such legs via a line. Non-contracted legs usually indicate the physical degrees of freedom where the orientation, mostly up vs. down, distinguishes between $\mathcal{H}$ and $\mathcal{H}^*$, i.e. between ket and bra. For the translationally invariant tensor network this is illustrated in figure (1.10). This visualization of tensor networks is a powerful tool especially when the networks grow bigger and more complicated with multiple tensors involved. In the case at hand, the network structure is quite simple, so that we will mostly work without visulized tensor networks.

According to [9], any state which is invariant under a symmetry action can be decomposed and represented in form of a tensor network respecting that symmetry group. That allows us to focus on these local tensors and still cover all possible observables. However, we are mostly going the other direction and face the question whether a given local tensor corresponds to a quantum state, which is necessarily positive semidefinite. The latter is not the case in general but there are ways to deal with this issue. It is possible to choose another, bigger network architecture that incorparates the symmetry group and ensures positive semidefiniteness by construction. We will introduce an example of such a network in the following section. In any case, our focus will remain on the simple translationally invariant network.

## 1.3 A first decision problem for tensor networks

In this section, we will catch up with the question whether a translationally invariant tensor network represents a state. More precisely, we will discuss a decision problem based on this issue. But instead of being concerned with the positivity of a single translationally invariant network, we pose the question for the full set of tensor network operators induced by a local tensor. We will then discuss the undecidability of that problem as well as its broader implications.

Before we introduce the decision problem we have to deal with a technical detail. Due to the definition of computability with a finite alphabet, the set of instances of a decision problem has to be enumerable to be non-trivial. This, however, is in contradiction with the uncountable Hilbert spaces used in quantum machanics. To fit the requirements of non-trivial decision problems, we have to thin out a representative enumerable space. In this thesis, this is achieved most of the time by using a pre-Hilbert space over the algebraic numbers $\mathbb{A}$ instead of a Hilbert spaces over $\mathbb{C}$. In contrast to a Hilbert space, a pre-Hilbert space does not require completeness, i.e. the convergence of Cauchy sequences. The completeness of a Hilbert space is in close relation with uncountability. This is why we have to let loose this condition. To minimize the loss of comfort, we will mostly use the algebraic numbers as a replacement for the complex ones. The **algebraic numbers** $\mathbb{A}$ are all those numbers that may appear as a root of a polynomial with coefficients in the rationals, or - similarily - in the integers. Those are indeed countable and to avoid getting entangled in too many details, we define these numbers by

$$\mathbb{A} := \{z \in \mathbb{C} \mid \exists p \in \mathbb{Z}[x] \setminus \{0\} : p(z) = 0\}$$

The advantage of the algebraic numbers over the rationals and their complex extension $\mathbb{Q}_i$ is the participation of algebraic expressions such as square roots. Non-algebraic numbers are for example the circle number $\pi$ as well as the Euler number $e$. In many cases, a result obtained for these unphysical pre-Hilbert spaces can be broadcasted to the physical case of Hilbert spaces. Thus, it is more of a technical detail needed to use the theory of (un)decidability for our purpose.

Having clarified this issue, we are ready to introduce our first decision problem concerning tensor networks. It is essentially dealing with the question whether a local tensor yields a positive semidefinite translationally invariant tensor network for all system sizes $n \in \mathbb{N}$.

**Problem 3** (PSD problem). *Let $d, D \in \mathbb{N}$. Given a tensor $A \in \mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{Z}))$, decide whether*

$$\tau_n(A) := \sum_{\alpha \in [D]^n} A_{\alpha_1\alpha_2} \otimes A_{\alpha_2\alpha_3} \otimes \cdots \otimes A_{\alpha_n\alpha_1}$$

*is positive semidefinite for all $n \in \mathbb{N}$.*

The reason this is formulated for the integers rather than the algebraic numbers as announced just before is that this is a more general result. We have seen that an embedding is a reduction and therefore the result is valid in the same way for algebraic numbers.

**Theorem 12.** *The* PSD *problem is undecidable for $d, D \geq 7$. This is true even if we restrict the problem to $A \in \mathrm{Mat}_D(\mathrm{Mat}_d^{Diag}(\mathbb{Z}))$.*

The undecidability of this problem was proven in [8] by construction of a reduction from the zero-in-the-upper-left-corner problem [4]. The latter is a close relative of the matrix mortality problem where only the entry in the upper left corner has to be zero. The undecidability of the PSD problem was an intermediate step to prove a limitation to the purification of matrix product states. Purification forms can be used to deal with the uncertainty of positive semidefiniteness. One introduces auxillary purification degrees of freedom to ensure the tensor network to be positive semidefinite for all system sizes. We can think of these as another tensor network architecture which, also, yields translationally invariant states. More specifically, the idea is to construct a translationally invariant element $|\psi\rangle$ in the purification Hilbert space $\mathcal{H}_{\mathrm{sys}} \otimes \mathcal{H}_{\mathrm{aux}}$ which corresponds to a translationally invariant state $\sigma_n = \mathrm{Tr}_{\mathrm{aux}} |\psi_n\rangle\langle\psi_n|$ on the system Hilbert space $\mathcal{H}_{\mathrm{sys}}$. In this case, our local tensor $B$ is not operator valued but takes its values in the purification Hilbert space, i.e. $B \in \mathrm{Mat}_D(\mathcal{H}_{\mathrm{sys}} \otimes \mathcal{H}_{\mathrm{aux}})$. The translationally invariant vector $|\psi_n\rangle$ is constructed in a similar fashion to the tensor network $\tau_n$:

$$|\psi_n\rangle := \sum_{\alpha \in [D]^n} B_{\alpha_1\alpha_2} \otimes B_{\alpha_2\alpha_3} \otimes \cdots \otimes B_{\alpha_n\alpha_1}.$$

As described above, we obtain the corresponding state by applying the partial trace of over the auxillary Hilbert space which yields the translationally invariant state

$$\sigma_n(B) := \mathrm{Tr}_{\mathrm{aux}}\left(|\psi_n\rangle\langle\psi_n|\right).$$

Since this construction naturally provides a positive semidefinite state for all system sizes $n \in \mathbb{N}$, it is definitely the more practical network to describe states of a quantum system. However, if we consider a series of states $\{\tau_n(A)\}_{n\in\mathbb{N}}$ given by a local tensor $A \in \mathrm{Mat}_D(\mathcal{L}(\mathcal{H}))$, it turns out that, in general, there is no universal local tensor $B \in \mathrm{Mat}_{D'}(\mathcal{H}_{\mathrm{sys}} \otimes \mathcal{H}_{\mathrm{aux}})$ such that

$$\tau_n(A) \propto \sigma_n(B)$$

for all systems sizes $n \in \mathbb{N}$. This was shown to be a consequence of the undecidability of the PSD problem in [8]. The basic idea behind such a proof of non-existence is to show that its existence could be used to solve an undecidable problem. Remember that we already saw such an ansatz earlier in the discussion of reductions. In the present case, the purification form of the translationally invariant network could be used to provide an algorithm to solve the PSD problem (3). Clearly, the purification form - if existing - has to be computable to be usable in an algorithm. It was shown in the referred paper that this is indeed the case. Note that, as mentioned in the beginning of this section, the decision problem has to be formulated in a subfield of the complex numbers, for example the algebraic numbers. The purification form has to be formulated over the same field since we want to make a statement about its computability. As described, the universal purification form over $\mathbb{A}$ cannot exist in general due to the undecidability of the PSD problem. As a consequence, the physical universal purification form, i.e. over the complex numbers $\mathbb{C}$, cannot exist either since they include the ones over $\mathbb{A}$.

## 1.4 Algebra

In this section we will shortly introduce the notion of an algebra and discuss some of the related concepts used in this thesis. We also devote a subsection to the discussion of polynomial algebras as an example of special interest in this thesis.

### 1.4.1 Basics on algebras

An algebra is a commonly used structure appearing in almost every branch of physics. In the same way as a group or a vector space, an algebra is a set equipped with certain operations similar to addition or multiplication. More precisely, algebras are closely related to vector spaces, as they carry a vector space structure but with an additional operation, a multiplication $* : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$ of algebra elements. In this section, we will introduce some of the basic definitions for algebras.

**Definition 13** (Algebra)**.** *An **algebra** $(\mathcal{A}, *)$ is a vector space over a field $F$ together with a $F$-bilinear pairing $* : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$. We call $\mathcal{A}$ an **associative algebra** if $*$ is associative and **unital** if there is a neutral element with respect to $*$.*

Typical examples of algebras are the 3-dimension real space $\mathbb{R}^3$ together with the cross product

$$\times : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$$

as well as the vector space of square matrices together with matrix multiplication. To be able to characterize the structure just defined, we introduce the structure preserving maps, i.e. homomorphisms.

**Definition 14** (Homomorphism of algebras)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be algebras over $F$ and $\phi : \mathcal{A} \to \mathcal{B}$. We call $\varphi$ a **homomorphism of algebras** if it is a homomorphism of $F$-vector spaces and if for any $a, b \in \mathcal{A}$ it satisfies $\varphi(a * b) = \varphi(a) * \varphi(b)$.*

A bijective homomorphism is called an **isomorphism** and its existence indicates structural equivalence. One typically treats two isomorphic objects as one and the same since they cannot be distinguished by the respective structure. A famous example of such an equivalence is given by the algebra $(\mathbb{R}^3, \times)$ described above and the algebra

$$\mathfrak{su}(2) := (\mathrm{span}_{\mathbb{R}}(i\sigma_x, i\sigma_y, i\sigma_z), [\cdot, \cdot]),$$

where $\sigma_x, \sigma_y, \sigma_z \in \mathrm{Mat}_2(\mathbb{C})$ are the Pauli matices

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and $[\cdot, \cdot]$ the matrix commutator defined by $[A, B] := AB - BA$. There is an isomorphism of algebras given by the map

$$\begin{aligned} \phi : \mathbb{R}^3 &\to \mathfrak{su}(2) \\ \hat{e}_k &\mapsto i\sigma_k \end{aligned}$$

where $\{\hat{e}_k\}_{k=x,y,z}$ is the standard basis of $\mathbb{R}^3$. These concepts will turn out useful when we have results valid for a certain type of algebras.

As it is the case for groups and other structures, an algebra may possess subsets that are closed under all operations. We will call such a subset subalgebras.

**Definition 15** (Subalgebra). *A sub-vector space $S$ of an algebra $\mathcal{A}$ is called a **subalgebra** of $\mathcal{A}$ iff it is closed under algebra multiplication, i.e. $S * S \subseteq S$.*

A natural way of generating a subalgebra of $\mathcal{A}$ is to choose a subset $G \subseteq \mathcal{A}$ and take its closure under the algebra operations, i.e. the set of elements that can be reached by means of finitely many additions and multiplications, scalar as well as algebra, on the set $G$. These are all elements of the form

$$a = \sum_{n=1}^{k} \sum_{g \in G^n} c_g g_1 * \cdots * g_n$$

for any $k \in \mathbb{N}$ and $c \in F^G$ with $c_g \neq 0$ for only finitely many $g \in G$. This set is obviously closed under algebra operations and, thus, a subalgebra of $\mathcal{A}$. More specifically, it is the smallest subalgebra of $\mathcal{A}$ containing the set $G$. In the example presented above, each non-trivial element $v \in \mathbb{R}^3 \setminus \{0\}$ generates a one-dimensional subalgebra $S_v := \mathrm{span}_{\mathbb{R}}(v)$ with the trivial algebra multiplication due to $v \times v = 0$. If we take $G$ to consist of two lineary independent elements $v, w \in \mathbb{R}^3 \setminus \{0\}$, we already generate the whole algebra since the vectors $v, w$ and $v \times w$ are lineary independent. Thus, the smallest set of elements generating the algebra $(\mathbb{R}^3, \times)$ consists of two lineary independent vectors.

**Definition 16** (Generating set). *Let $\mathcal{A}$ be an algebra. We call a subset $G \subseteq \mathcal{A}$ a **generating set** of $\mathcal{A}$ if the smallest subalgebra of $\mathcal{A}$ containing $G$ is $\mathcal{A}$ itself. We call $\mathcal{A}$ **finitely generated** if it is generated by a finite set.*

**Example: Polynomial functions.** The set of polynomial functions in one variable provides a vector space structure as well as an algebra operation, namely the standard multiplication of polynomials. We will devote the next section to the introduction of abstract polynomials, however, for this example consider the set

$$\mathcal{P} := \left\{ p : \mathbb{R} \to \mathbb{R} \,\middle|\, \exists k \in \mathbb{N}\, \exists\, c_0, \ldots, c_k \in \mathbb{R} : \left( p : x \mapsto \sum_{n=0}^{k} c_n x^n \right) \right\}$$

of real functions of polynomial form. It is immediate from the definition that $\mathcal{P}$ is finitely generated by the constant function $\mathbb{1} : x \mapsto 1$ as well as the identity map $\mathrm{id} : x \mapsto x$. For multiple variable we get a similar result but with a projection function $p_i : (x_1, \ldots, x_d) \mapsto x_i$ for each variable. Note that in contrast to the former example, the algebra $\mathcal{P}$ is infinite-dimensional as a vector space. Moreover, each element $p \in \mathcal{P}$ has an associated natural number, the degree $\deg p$ of the polynomial. This is the smallest natural number $k$ such that

$$p : x \mapsto \sum_{n=0}^{k} c_n x^n$$

or in other words the order of the highest non-vanishing coefficient. One can convince oneself that this grading satisfies $\deg(\lambda p) = \deg(p)$, $\deg(p \cdot q) = \deg(p) + \deg(q)$ as well as $\deg(p + q) \leq \deg(p) + \deg(q)$ for any $p, q \in \mathcal{P}$ and $\lambda \in \mathbb{R} \setminus \{0\}$. In general, we can define such grading structures in the following way, which induces the (in)equalities for $\deg : \mathcal{P} \to \mathbb{N}$ above.

**Definition 17** ($\mathbb{N}$-graded algebra)**.** *We call an algebra $(\mathcal{A}, *)$ a* $\mathbb{N}$***-graded algebra*** *iff there is a family of sub-vector spaces $(\mathcal{A}_n)_{n \in \mathbb{N}}$ such that*

$$\mathcal{A} = \bigoplus_{n=0}^{\infty} \mathcal{A}_n$$

*as well as $\mathcal{A}_k * \mathcal{A}_l \subseteq \mathcal{A}_{k+l}$.*

In the previous section, we introduced tensor networks to describe states and observables of a quantum mechanical system. Decision problem 3 dealt with a series of tensor networks that represented operators on the $n$-fold copy of a base Hilbert space. Those spaces can be combined to the $\mathbb{N}$-graded algebra

$$\mathcal{L}(\mathcal{H})^{\otimes} := \bigoplus_{n=0}^{\infty} \mathcal{L}(\mathcal{H}^{\otimes n})$$

called the **tensor-power algebra** of $\mathcal{L}(\mathcal{H})$. This has the advantage that each tensor network in the series $\{\tau_n(A)\}_{n \in \mathbb{N}}$ takes its value in the same space $\mathcal{L}(\mathcal{H})^{\otimes}$. This will be a crucial point for the generalization to algebra-valued networks in the next chapter.

Besides the algebra structure, we will have to deal with different notions of positivity in the following chapter. In the tensor-power algebra we have a natural understanding of positive elements given by positive semidefinite operators. When generalizing the decision problem to arbitrary algebras, we lose this notion of positivity. Thus, in the general case, we will replace the PSD operators by another notion of positivity given by a convex cone. From now on, we will restrict the discussion to algebras over the number fields $F = \mathbb{Q}, \mathbb{A}, \mathbb{C}$ etc. Before defining the structure of a convex cone, we recall the definition of a convex set.

**Definition 18** (Convex set, Convex cone)**.** *Let $V$ be a vector space over $\mathbb{C}$ or one of its subfields. We call $\mathcal{C} \subseteq V$ **convex** iff it is closed under convex combinations, i.e. for each $v, w \in \mathcal{C}$ and $\lambda \in [0, 1]$ the vector $\lambda v + (1 - \lambda)w \in \mathcal{C}$. We call $\mathcal{C} \subseteq V$ a **convex cone** iff it is additionally closed under multiplications with positive scalars, i.e. $\lambda \cdot \mathcal{C} \subseteq \mathcal{C}$ for all $\lambda > 0$.*

Before going on with polynomial algebras in the next section, we shall introduce one last special structure on algebras.

**Definition 19** ($*$-Algebra)**.** *A **$*$-algebra** over $F$ is a algebra $\mathcal{A}$ over $F$ equipped with an involution, i.e. a map $\bullet^* : \mathcal{A} \to \mathcal{A}$ that is:*

- *conjugate-linear, i.e. $\forall a_1, a_2 \in \mathcal{A}\ \forall \lambda \in F : (a_1 + \lambda a_2)^* = a_1^* + \overline{\lambda} a_2^*$,*
- *an antihomomorphism, i.e. $\forall a_1, a_2 \in \mathcal{A} : (a_1 a_2)^* = a_2^* a_1^*$, and*
- *self-inverse: $(a^*)^* = a$.*

Typical examples of $*$-structures are the transpose map for matrices over $\mathbb{R}$, the adjoint map and simple complex conjugation. Elements invariant under the involution are usually called **self-adjoint** or **symmetric**.

### 1.4.2 Polynomial algebras

In the following, we introduce two types of abstract polynomials: the commutative and the non-commutative polynomial algebras. We will shortly discuss concepts, which will be of interest for the following sections. These types of algebras will accompany us throughout the rest of this thesis.

When talking about abstract polynomials, we mean expressions of the form

$$p = \sum_{k=0}^{n} c_k X^k$$

where $c_k$ are elements of a certain ring $R$ and $X$ is an indeterminate. We may also introduce multiple symbols as indeterminates $\{X_1, X_2, \dots\}$, which results in polynomials in multiple variables. However, for their formal definition, we have to specify how these different indeterminates interact. In the following, we shall discuss the two most important cases. On the one hand, we have got the case with no further structure of the indeterminates, ending up with $X_1 X_2$ and $X_2 X_1$ being different expressions. On the other hand, we have the case of full commutative variables, where the precise order of symbols is of no importance. Before we define polynomial algebras in multiple variables, we recall the definition for a single indeterminate.

**Definition 20** (Polynomial ring)**.** *We define the* ***polynomial ring*** $R[X]$ *over a ring* $R$ *to be the space*

$$R[X] := \left\{ (p_k)_{k\in\mathbb{N}_0} \in R^{\mathbb{N}_0} \,\middle|\, \operatorname{supp}(p) \textit{ finite} \right\}$$

*of sequences with finitely many non-vanishing entries together with standard addition and the algebra multiplication*

$$* : (p,q) \mapsto \left( \sum_{i=0}^{k} p_i q_{k-i} \right)_{k\in\mathbb{N}_0} .$$

*The element* $X := (\delta_{i,1})_{i\in\mathbb{N}_0}$ *is called the* ***indeterminate****. We can write each element* $p = (p_k)_{k\in\mathbb{N}_0} \in R[X]$ *in the* ***monomial representation****, i.e. as sum over all non-vanishing entries*

$$p = \sum_{k\in\mathbb{N}_0} p_k X^k$$

*where* $X^k = (\delta_{i,k})_{i\in\mathbb{N}_0}$ *is the k-fold product of X.*

The connection of these abstract polynomials to polynomial functions as discussed earlier is given by the evaluation homomorphism. Let $\mathcal{A}$ be an algebra over $R$, then we define the evaluation in $\mathcal{A}$ by

$$\begin{aligned} \operatorname{ev} : R[X] \times \mathcal{A} &\to \mathcal{A} \\ (p,a) &\mapsto p(a) := \sum_k p_k a^k . \end{aligned}$$

The polynomial functions over the $\mathbb{R}$ are obtained by evaluation of the polynomials $\mathbb{R}[X]$ on the real numbers. In this way, the map ev can be interpreted as an algebra homomorphism $\mathbb{R}[X] \to \operatorname{Map}(\mathbb{R};\mathbb{R})$.

We obtain the ring of commutative polynomials by iterative application of the definition of the polynomial ring in one variable. For example, we define the polynomials in two variables as polynomial with polynomial coefficients:

$$p = \sum_k p_k Y^k \in F[X,Y], \quad p_k \in F[X].$$

This iterative construction yields the following definition.

**Definition 21** (Commutative polynomials)**.** *We define the* ***ring of commutative polynomials*** $R[X_1,\dots,X_d]$ *in d indeterminates iteratively by*

$$R[X_1,\dots,X_d] := R[X_1,\dots,X_{d-1}][X_d],$$

*i.e. the commutative polynomial ring over R in d variables is defined as the polynomial ring with coefficients in* $R[X_1,\dots,X_{d-1}]$*.*

The non-commutative polynomials on the other hand are defined similarily to the single variable case. But instead of the index set $\mathbb{N}$, which can be thought of as the set of words in a single symbol alphabet, we use the set of words in an alphabet of $d$ symbols.

**Definition 22** (Non-commutative polynomials). *We define the* ***ring of non-commutative polynomials*** $R\langle X_1, \dots, X_d\rangle$ *to be the set*

$$R\langle X_1, \dots, X_d\rangle := \{p : [d]^* \to R \,|\, \mathrm{supp}(p) \textit{ finite}\}$$

*where $[d]^*$ is the set of all words in $[d]$. The algebra multiplication is defined such that*

$$p * q : \omega \mapsto \sum_{\omega = \omega_1 \omega_2} p_{\omega_1} q_{\omega_2}$$

*for $p, q \in R\langle X_1, \dots, X_d\rangle$. The elements*

$$X_j : \omega \mapsto \begin{cases} 1_R \textit{ if } \omega = j \\ 0_R \textit{ elsewise} \end{cases}$$

*with $j \in [d]$ are called the* ***indeterminates****. We can write each polynomial $p \in R\langle X_1, \dots, X_d\rangle$ in the* ***monomial representation****, i.e. as a sum*

$$p = \sum_{\omega} p_\omega X_{\omega_1} X_{\omega_2} \dots X_{\omega_n}$$

*over the non-vanishing entries.*

Those polynomial algebras carry a natural grading which is defined similarly to the one of polynomial functions. Specifically, the undeterminates are defined to be of degree $\deg(X_i) = 1$ and a monomial, a product of $n$ indeterminates, is of degree $n$. For a general polynomial, both commutative or non-commutative, the **degree** is then the highest degree of a monomial with non-vanishing coefficient. The zero polynomial, i.e. the zero sequence, gets a special treatment and takes the degree $\deg(0) = -\infty$ by convention. We call a polynomial **homogeneous** if it is a combination of monomials of the same degree.

### *-Structures

To equip a polynomial algebra with a *-structure, we have to decide about the corresponding dependency of the indeterminates, i.e. how the involution maps the monomials. Since each polynomial algebra is generated by the unit element together with the degree one monomials, it is sufficient to fix their behavior under the involution. The two usual choices are the following:

1. The trivial involution $\bullet^* : X_i \mapsto X_i$. We can think of the variables as real or rather self-adjoint quantities. This structure can be defined on polynomial algebras with any number of indeterminates.

2. For polynomials in $2d$ variables the involution defined by $X_i^* := X_{i+d}$ for $1 \leq i \leq d$. In this case, we may think of the variables as complex quantities and we shall write $X_i^*$ instead of $X_{i+d}$ in general. Those algebras are typically denoted by $R[\underline{X}, \underline{X}^*]$ or $R\langle \underline{X}, \underline{X}^* \rangle$ respectively.

In both cases, the *-structure extends to general elements due to the antilinearity and the identity $(a_1 a_2)^* = a_2^* a_1^*$. The latter causes a reversion of the monomial order, i.e. of the word in the indeterminates. For a general element of a polynomial algebra we find

$$\left( \sum_{\omega} p_\omega X_{\omega_1} \dots X_{\omega_n} \right)^* = \sum_{\omega} \overline{p_\omega} X_{\omega_n}^* \dots X_{\omega_1}^*.$$

**Positivity cones**

There are several natural notions of positivity in these polynomial algebras. As both the commutative as well as the non-commutative can be considered as a $*$-algebra, it is natural to define the cone of **sum of squares**:

$$\Sigma^2(\mathcal{A}) := \left\{ s \in \mathcal{A} \,\middle|\, \exists k \in \mathbb{N}, \exists a_1, \dots, a_k \in \mathcal{A} : s = \sum_{i=1}^{k} a_i^* a_i \right\}. \tag{1.3}$$

This sum of squares cone, in short SOS cone, can be constructed for all polynomial algebras but depends on the choice of $*$-structure. On the other hand, when we think of the polynomials not as an abstract object but more as a polynomial function, there is another natural notion of positivity. However, this time the definitions differ for the various cases. For commutative polynomials, we define **positivity on** $\mathbb{A}$ by

$$p \in \mathbb{A}[x_1, \dots, x_d] \text{ is } \mathbb{A}\text{-positive} :\Leftrightarrow \forall x \in \mathbb{A}^d : p(x) \geq 0. \tag{1.4}$$

It is immediate that sum of squares inplies positivity on $\mathbb{A}$. However, the opposite direction is not true in general. A typical example of this issue is the Motzkin polynomial [25]

$$p(x, y) = x^2 y^4 + x^4 y^2 - 3x^2 y^2 + 1$$

which is positive on $\mathbb{A}$ but not sum of squares. For non-commutative polynomials, blunt evaluation on $\mathbb{A}$ would disregard the fact of non-commutativity. Instead, we will respect it by evaluating on matrices $\mathrm{Mat}_k(\mathbb{A})$. This is, we say a (non-commutative) polynomial $p \in \mathbb{A}\langle \underline{X} \rangle$ is **matrix-positive** iff

$$\forall k \in \mathbb{N}\, \forall M_1, \dots, M_d \in \mathrm{Her}_k(\mathbb{A}) : p(M_1, \dots, M_d) \geqslant 0. \tag{1.5}$$

For polynomials with a non-trivial $*$-structure, i.e. $p \in \mathbb{A}\langle \underline{X}, \underline{X}^* \rangle$, we replace $\mathrm{Her}_k(\mathbb{A})$ by the full set of matrices $\mathrm{Mat}_k(\mathbb{A})$. In contrast to the commutative case, it was shown by Helton that matrix positivity equals sum of squares.

**Theorem 23** (Helton's theorem [15]). *A non-commutative polynomial is sum of squares if and only if it is matrix-positive.*

A third sense of positivity is less commonly used one and it naturally enters the stage when seeing a polynomial as a collection of coefficients. From this point of view, we might consider polynomials, commutative or non-commutative, to be positive iff they have exclusively **non-negative coefficients**. We denote this cone by $\mathbb{A}_{\geq 0}[\underline{X}]$ and $\mathbb{A}_{\geq 0}\langle \underline{X} \rangle$ in the commutative and non-comutative case respectively. This might be seen as the most primitive notion of positivity since it ignores most of the structure, for example the algebra operation.

# Chapter 2

# Decision Problems for Algebra-valued Tensor Networks

In this chapter, we will introduce a generalization of the decision problem (3) to translationally invariant tensor networks valued in general algebras. This will be followed by a discussion of various special cases concerning polynomial algebras, begining with the non-commutative case due to its closer connection to the original problem. Regarding this link, we will focus our efforts on the construction of a reduction from the mentioned problem.

On the other hand, for the commutative case we will choose another way due to subleties connected to the non-commutativity of $\mathcal{L}(\mathcal{H})^{\otimes}$. Instead, we will use the close relation of the translationally invariant tensor network to the moments of the local tensor and establish a recursion formula for those.

## 2.1 The moments-membership problem

The translationally invariant tensor network that we have introduced in the last section takes its values in the tensor product space $\mathcal{L}(\mathcal{H})^{\otimes n}$ where $n$ is the length of the network. Networks of different length therefore provide elements of different spaces. However, we may think of these as subspaces of different grading in a $\mathbb{N}$-graded algebra, namely the tensor power algebra

$$\mathcal{L}(\mathcal{H})^{\otimes} := \bigoplus_{n=1}^{\infty} \mathcal{L}(\mathcal{H}^k).$$

An element of $A \in \mathcal{L}(\mathcal{H})^{\otimes}$ is an infinite collection of elements $(A_n)_{n\in\mathbb{N}}$ such that $A_n \in \mathcal{L}(\mathcal{H}^{\otimes n})$. Moreover, only a finite amount of the entries is supposed to be non-vanishing. An element with a single non-vanishing entry $A_n \in \mathcal{L}(\mathcal{H}^{\otimes n})$ shall be called a homogeneous element of degree $n$. We can identify a homogeneous element of degree $n$ with an element in $\mathcal{L}(\mathcal{H}^{\otimes n})$, i.e. with its only non-vanishing entry. The algebra product of two homogeneous elements $A_n \in \mathcal{L}(\mathcal{H}^{\otimes n})$ and $B_m \in \mathcal{L}(\mathcal{H}^{\otimes m})$ is defined to be the homogeneous element $A_n \otimes B_m$ of degree $n + m$. This definition provides an algebra product by linearity, which we will also denote by $\otimes$. With this construction, the translationally invariant tensor network takes its value in the algebra $\mathcal{L}(\mathcal{H})^{\otimes}$ independently of its size.

In view of decision problems, we may do the same construction for matrix spaces. For a ring $R$ - think of $\mathbb{Z}$ or $\mathbb{A}$ - we define the tensor power space

$$\mathrm{Mat}_n(R)^{\otimes} := \bigoplus_{k=1}^{\infty} \mathrm{Mat}_{n^k}(R)$$

and endow it with an algebra structure in the very same way as before, but this time $\otimes$ denotes the Kronecker product of matrices. For $R = \mathbb{C}$ the algebras $\mathcal{L}(\mathcal{H})^{\otimes}$ and $\mathrm{Mat}_d(\mathbb{C})^{\otimes}$ with $d = \dim \mathcal{H}$ can be identified by choosing a basis of $\mathcal{H}$. From this point of view, it seems closest to generalize the decision problem in the following way.

**Problem 4** (Membership in $\mathcal{C}$)**.** *Let $\mathcal{A}$ be an algebra over $\mathbb{A}$, $\mathcal{C}$ a convex cone in $\mathcal{A}$ and $D \in \mathbb{N}$. Given a matrix $A \in \mathrm{Mat}_D(\mathcal{A})$, decide whether the tensor network*

$$\chi_n(A) := \sum_{\alpha \in [D]^n} A_{\alpha_1\alpha_2} * A_{\alpha_2\alpha_3} * \cdots * A_{\alpha_n\alpha_1}$$

*is an element of the cone $\mathcal{C}$ for all $n \in \mathbb{N}$.*

Here, some adjustments were made with respect to the definition of the original problem (3). First, we replaced the matrices representing linear operators by general algebra elements. This clearly brings along the necessity of a new positivity measure that is met with the convex cone $\mathcal{C}$. The definition of the tensor network structure, i.e. the underlying graph, is similar to the quantum case. The tensor product in $\tau_n$ has been replaced by the new algebra product. We thereby lose the notion of translational invariant elements in $\mathcal{A}$. More precisely, the reordering action of $\mathbb{Z}_n$ on the product of elements $a_1, \ldots, a_n \in \mathcal{A}$ by

$$T : a_1 * a_2 * \cdots * a_n \mapsto a_n * a_1 * \cdots * a_{n-1}$$

does not induce an action on the algebra $\mathcal{A}$. That is, we can have a translationally invariant construcion of an element in $\mathcal{A}$, for example by a tensor network, but without the additional information of this structure, there is no notion of translationally invariance in the algebra. In a graded algebra however, we can introduce such a structure on the homogeneous subspaces of $\mathcal{A}$. More specifically, for each $k \in \mathbb{N}$ there is a $\mathbb{Z}_k$ translational action on the subspace of $\mathcal{A}_k$ which is generated by the elements of $\mathcal{A}_1$, i.e. the space

$$\langle \mathcal{A}_1 \rangle_k := \left\{ \sum_{i=1}^{n} a_i^{[1]} * \cdots * a_i^{[k]} \,\middle|\, n \in \mathbb{N},\, a_i^{[j]} \in \mathcal{A}_1 \right\}.$$

In the case of a tensor power algebra such as $\mathcal{L}(\mathcal{H})^{\otimes}$ the whole algebra is generated by degree one elements. The notion of translationally invariance is therefore defined on the full subspace $\mathcal{A}_k$ and coincides with the original definition. The same is true for polynomial algebras.

The careful reader might already noticed that problem (4) does not exactly reproduce the initial problem (3) in the case of $\mathcal{A} = \mathrm{Mat}_d(\mathbb{Z})^{\otimes}$ with the cone

$$\mathrm{PSD}_d(\mathbb{Z})^{\otimes} := \bigoplus_{k=1}^{\infty} \left(\mathrm{PSD}_{d^k}(\mathbb{C}) \cap \mathrm{Mat}_{d^k}(\mathbb{Z})\right).$$

Instead, we obtain an enlarged domain $\mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{Z})^{\otimes})$ for the decision problem. We can still state the undecidability of this problem since inclusion in an enumerable set is a reduction. Thus, enlargment of the domain does not change the nature of undecidability. On the other hand, if we reverse the situation, we cannot conclude the undecidability on the smaller domain just from undecidability on the bigger one. However, the results we present in this chapter are also valid for specific smaller subspaces due to the precise form of the reduction. In the case of naturally graded algebras for example, this is the case for the homogeneous subspace $\mathcal{A}_1$ of degree 1 as the input space. We will mostly work with the full algebra as input algebra but keep this issue in the back of our mind.

Before we actually start investigating the behavior of translationally invariant tensor networks in the background of polynomials, we draw an important connection to another mathematical structure: the moments of a matrix. For a matrix ring $\mathrm{Mat}_k(\mathcal{A})$ for any algebra $\mathcal{A}$, we define the ***n*-th moment** of $A \in \mathrm{Mat}_k(\mathcal{A})$ by

$$\mu_n(A) := \mathrm{Tr}(A^n).$$

The multiplication of the matrices is defined similar to matrices over a number field but with the algebra multiplication respectively. It is an interesting observation that the tensor network $\chi_n(A)$ can be associated with the moments $\mu_n$ of its local tensor.

**Proposition 24.** *The value of the translationally invariant tensor network $\chi_n$ for a local tensor $A \in \mathrm{Mat}_D(\mathcal{A})$ is equal to the n-th moment $\mu_n(A)$.*

*Proof.* For any local tensor $A \in \mathrm{Mat}_D(\mathcal{A})$, it is a straight foreward calculation that

$$\begin{aligned}\chi_n(A) &= \sum_{\alpha\in[D]^n} A_{\alpha_1\alpha_2} * A_{\alpha_2\alpha_3} * \cdots * A_{\alpha_n\alpha_1} \\ &= \mathrm{Tr}\left(\left\{\sum_{\alpha\in[D]^{n-1}} A_{\lambda\alpha_1} * A_{\alpha_1\alpha_2} * \cdots * A_{\alpha_{n-1}\sigma}\right\}_{\lambda,\sigma\in[D]}\right) = \mathrm{Tr}(A^n).\end{aligned}$$

which proves the statement. □

Thus, the algebra-valued translationally invariant tensor network is not just an artificial construct motivated by its apperance in quantum physics but rather a structure of broader interest [7]. With this new perspective, we shall call the generalized tensor-network problem (4) the **moments-membership problem**. This connection will be of special interest in the vecinity of commutative algebras.

## 2.2 Non-commutative algebras

We will begin our discussion with non-commutative algebras since they are closely related to the original problem. An example of special interest for us will be the algebra of non-commutative polynomials, where we pose the moments-membership problem for the cone of sum of squares polynomials. In this section, we will discuss an ansatz using a vector space homomorphism together with Helton's theorem to construct a potential reduction from the undecidable problem (3) of membership in PSD. However, it turns out that this ansatz is not suitable for a reduction for various reasons. One the one hand, the $*$-structure on the polynomial algebra is of a fundamentally different nature than the adjunction of linear operators. This leads to conflicting cone structures due to non-closedness of the sum of square cone under algebra multiplication. On the other hand, the reduction by a vector space homomorphism yields a disordering of the indices, also between in- and output indices of the linear operator. Positive semidefiniteness is not conserved by this reordering, which turns to be a major issue.

In subsection 2.2.2, we shall see that the original problem is more closely related to algebras with a cone of positive coefficients. The first of this kind that we consider is the algebra of non-commutative polynomials together with the cone of positive coefficients. We will find in section 2.2.3 that whether a homomorphism of vector spaces is a reduction is closely related to its unique extension to an algebra homomorphism. For a special breed of reductions the extension is supposed to be injective which rules out commutative algebras. This is one of the reasons why we have to consider these cases separately.

In the end of the section, we shall return to sum of square polynomials to discuss them in more detail. We will specifically show a bijective correspondence between non-commutative homogeneous sum of squares polynomials and positive semidefinite matrices.

### 2.2.1 A first look on SoS polynomials

In the following, let $d \in \mathbb{N}$ be the number of indeterminates and $D \in \mathbb{N}$ the bond dimension of the tensor network. In this section, we will discuss an ansatz for a proof of undecidability of the moments-membership problem for non-commutative sum of squares polynomials. The basic idea is to use a map

$$\phi : \mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{A})) \to \mathrm{Mat}_D(\mathbb{A}\langle z_1, \ldots, z_d \rangle)$$

that is acting componentwise on a matrix in $\mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{A}))$ according to the homomorphism $\hat{\phi} : \mathrm{Mat}_d(\mathbb{A}) \to \mathbb{A}\langle z_1, \ldots, z_d\rangle$ defined by

$$\hat{\phi}(\{A^{ij}\}_{i,j\in[d]}) := \sum_{i,j\in[d]} A_{ij} z_i z_j.$$

To check whether this map is a reduction, we have to evaluate the tensor network $\chi_n$ on the instances of the form $\phi(A)$ with $A \in \mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{A}))$. However, we will see that this ansatz is not suitable for the SOS cone structure.

To begin with, we apply the cyclic tensor network to an instance $\phi(A)$, $A \in \mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{A}))$, which yields

$$\begin{aligned} \chi_n(\phi(A)) &= \sum_{\alpha\in[D]^n} \hat{\phi}(A_{\alpha_1\alpha_2})\hat{\phi}(A_{\alpha_2\alpha_3})\ldots\hat{\phi}(A_{\alpha_n\alpha_1}) && (2.1)\\ &= \sum_{i,j\in[d]^n} \underbrace{\sum_{\alpha\in[D]^n} A^{i_1j_1}_{\alpha_1\alpha_2}A^{i_2j_2}_{\alpha_2\alpha_3}\ldots A^{i_nj_n}_{\alpha_n\alpha_1}}_{=\langle i_1\ldots i_n|\tau_n(A)|j_1\ldots j_n\rangle} z_{i_1}z_{j_1}\ldots z_{i_n}z_{j_n}. && (2.2) \end{aligned}$$

The representation in the last line draws a connection of the network $\chi_n$ evaluated for $\phi(A)$ to the matrix elements of the tensor network $\tau_n(A)$. We are supposed to show that $\tau_n(A)$ being positive semidefinite is equivalent to $\chi_n(\phi(A))$ being sum of squares. This would result in the undecidability of moments-membership problem for sum of square polynomials due to the undecidability of the tensor network problem (3). As we have seen, Helton's theorem provides a connection of sum of squares polynomials and positive semidefinite matrices. The central idea is to use this theorem to show that $\chi_n(\phi(A))$ is sum of squares if and only if $\tau_n(A) \in \mathrm{PSD}$. For the first direction, assume $\chi_n(\phi(A))$ to be sum of squares. According to Heltons's theorem, for any $k \in \mathbb{N}$ and any tuple of matrices $M_1, \ldots, M_d \in \mathrm{Her}_k(\mathbb{A})$ we find that

$$\chi_n(A)(M_1, \ldots, M_d) = \sum_{i,j\in[d]^n} \langle i_1 \ldots i_n|\tau_n(A)|j_1 \ldots j_n\rangle M_{i_1}M_{j_1}\ldots M_{i_n}M_{j_n}$$

is positive semidefinite. To prove that $\tau_n(A)$ is PSD, it would be sufficient to show that one can choose $M_1, \ldots, M_d \in \mathrm{Her}_{d^n}$ such that the set of products of the form $\{M_{i_1}M_{i_2}\ldots M_{i_{2n}}\}_{i\in[d]^{2n}}$ is an orthonormal basis of $\mathrm{Mat}_{d^n}(\mathbb{A})$ which is unitary equivalent to $\{|i_1 \ldots i_n\rangle\langle j_1 \ldots j_n|\}_{i,j\in[d]^n}$. This would then yield

$$\begin{aligned} \chi_n(A)(M_1, \ldots, M_d) &= \sum_{i,j\in[d]^n} \langle i_1 \ldots i_n|\tau_n(A)|j_1 \ldots j_n\rangle M_{i_1}M_{j_1}\ldots M_{i_n}M_{j_n}\\ &= \sum_{i,j\in[d]^n} \langle i_1 \ldots i_n|\tau_n(A)|j_1 \ldots j_n\rangle U^\dagger|i_1 \ldots i_n\rangle\langle j_1 \ldots j_n|U\\ &= U^\dagger\tau_n(A)U, \end{aligned}$$

where we used the linearity of the adjoint map in the last step. This would prove the positivity of $\tau_n(A)$, since the spectrum is invariant under conjugation by unitaries. However, so is the rank of a matrix. Therefore, a matrix unitarily equivalent to $|i_1 \dots i_n\rangle\langle j_1 \dots j_n|$ has to be of unit rank, which is in contradiciton with the following result:

**Lemma 25.** *Let $B_1, \dots, B_{d^{2n}} \in \mathrm{Mat}_{d^n}(\mathbb{A})$ an orthonormal basis with respect to the scalar product $\langle A, B\rangle = \mathrm{tr}(A^\dagger B)$. Moreover, assume $\mathrm{rank}(B_i) = 1$ for all $i \in [d^{2n}]$. Then, there are no matrices $M_1, \dots, M_d \in \mathrm{Her}_{d^n}(\mathbb{A})$ together with a bijection $\pi : [d]^{2n} \to [d^{2n}]$ such that*

$$B_{\pi(i_1,\dots,i_{2n})} = M_{i_1} \dots M_{i_{2n}}.$$

*Proof.* We will prove the statement by contradiction. Assume there are matrices $M_1, \dots, M_d \in \mathrm{Her}_{d^n}(\mathbb{A})$ and a bijection $\pi$ as defined above. For $i \in [d]$ we find

$$B_{\pi(i,\dots,i)} = M_i^{2n} \in \mathrm{Her}_{d^n}(\mathbb{A}).$$

We can diagonalize $M_i = U^\dagger D U$ since it is hermitian and further find that $B_{\pi(i,\dots,i)} = U^\dagger D^{2n} U$. Since $B_{\pi(i,\dots,i)}$ is of rank 1, so is $D^{2n}$, $D$ and therefore $M_i$. As a consequence, there are $v_1, \dots, v_d \in \mathbb{A}^{d^n}$ such that $M_i = \mathbb{P}_{v_i} = |v_i\rangle\langle v_i|$. Since the matrices $(B_j)_{j\in[d^{2n}]}$ are orthonormal and $B_{\pi(i,\dots,i)} = M_i^{2n} = \mathbb{P}_{v_i}$, we find that

$$\delta_{ij} = \mathrm{tr}(B_{\pi(i,\dots,i)} B_{\pi(j,\dots,j)}) = \mathrm{tr}(\mathbb{P}_{v_i}\mathbb{P}_{v_j}) = |\langle v_i|v_j\rangle|^2$$

for any $i, j \in [d]$. But this implies $\mathbb{P}_{v_i}\mathbb{P}_{v_j} = 0$ for $i \neq j$ and thus $B_{\pi(i_1,\dots,i_{2n})} = 0$ if $i_k \neq i_l$ for some $k, l \in [2n]$, which is a contradiction completing the proof. □

Hence, we conclude that there are no matrices $M_1, \dots, M_d$ and unitary $U$ such that $\tau_n(A) = U^\dagger \chi_n(\phi(A))(M_1, \dots, M_n)U$. In fact, things get even worse, as the following stronger result states that we cannot form any orthonormal basis by products $M_{i_1} \dots M_{i_{2n}}$:

**Lemma 26.** *For $d > 1$, there are no $M_1, \dots, M_d \in \mathrm{Her}_{d^n}(\mathbb{A})$ such that the products $B_{i_1,\dots,i_{2n}} = M_{i_1} \dots M_{i_{2n}}$ form an orthonormal basis of $\mathrm{Mat}_{d^n}(\mathbb{A})$ with respect to the Frobenius product $\langle A, B\rangle := \mathrm{tr}(A^\dagger B)$.*

*Proof.* We will again prove it by contradiction. Consider the $B_{i_1,\dots,i_{2n}}$ to be an orthonormal basis and concider the element $B_{1,\dots,1,2} = M_1^{2n-1} M_2$, then the cyclicity of the trace yields

$$\begin{aligned} 1 = \mathrm{tr}(B^\dagger_{1,\dots,1,2} B_{1,\dots,1,2}) &= \mathrm{tr}\left(\left[M_2 M_1^{2n-1}\right] \cdot \left[M_1^{2n-1} M_2\right]\right) \\ &= \mathrm{tr}\left(\left[M_1^{2n}\right] \cdot \left[M_1^{2n-2} M_2 M_2\right]\right) = \mathrm{tr}(B^\dagger_{1,\dots,1} B_{1,\dots,1,2,2}) = 0, \end{aligned}$$

which is a contradiction. □

This lemma implies that one cannot reconstruct the matrix $\tau_n(A)$ with the data obtained by using a single set of matrices $M_1, \dots, M_d \in \mathrm{Her}_{d^n}(\mathbb{A})$. The products of the matrices cannot be a basis which results in an undecodable mixing of the matrix elements of $\tau_n(A)$. To resolve this problem one has to use either higher dimensional matrices or multiple sets of matrices. However, we will instead later establish another connection between sum of square polynomials and positive semidefinite matrices, namely the Gram representation. We will use this and some other arguments to show that the mapping $\phi$ as defined above is not a reduction of decision problems.

### 2.2.2 Polynomials with non-negative coefficients

Due to the modest success of the efforts in the foregoing section concerning sum of squares, let us try to find a more suitable cone for the reduction. Besides the sum of square polynomials as a natural notion of positivity, we have also introduced the cone of non-negative coefficients (NNC). Indeed, we shall see that the construction of the former section will suffice as a reduction when considering the NNC cone. This yields the following instance of the moments-membership problem:

**Problem 5** (Non-negative coefficients (NNC)). *Let $d$ and $D$ be natural numbers. Given a matrix $A \in \mathrm{Mat}_D(\mathbb{Z}\langle X_1, \ldots, X_d\rangle)$, decide whether $\chi_n(A)$ has non-negative coefficients for all $n \in \mathbb{N}$.*

As mentioned, we use the very same construction as the one discussed in the former section. However, this time this will yield our first result of undecidability for an algebra-valued tensor network. Again, this is formulated for the integers $\mathbb{Z}$ instead of the algebraic numbers but it will also be valid for the latter field since the inclusion of $\mathrm{Mat}_D(\mathbb{Z}\langle \underline{X}\rangle)$ in $\mathrm{Mat}_D(\mathbb{A}\langle \underline{X}\rangle)$ is a reduction according to lemma (9).

**Theorem 27.** *The NNC problem (5) is undecidable for $d, D \geq 7$. This is true even if we limit ourselves to inputs of the form $A = \left(\sum_{i=1}^{d} c^i_{\alpha\beta} X_i^2\right)_{\alpha,\beta\in[D]}$ with $c^i_{\alpha\beta} \in \mathbb{Z}$.*

*Proof.* The central element of this proof is the vector space homomorphism

$$\begin{aligned}\phi : \mathrm{Mat}_d(\mathbb{Z}) &\to \mathbb{Z}\langle X_1, \ldots, X_d\rangle \\ a &\mapsto \sum_{i,j\in[d]} a_{ij} X_i X_j\end{aligned}$$

which can be extended to matrices component-wise, i.e. $\Phi : \mathrm{Mat}_D(\mathrm{Mat}_d(\mathbb{Z})) \to \mathrm{Mat}_D(\mathbb{Z}\langle \underline{X}\rangle)$. We will show that $\Phi$ is a reduction from the membership-in-PSD problem (3) by using the identity (2.2),

$$\chi_n(\Phi(A)) = \sum_{i,j\in[d]^n} \langle i|\tau_n(A)|j\rangle X_{i_1} X_{j_1} \ldots X_{i_n} X_{j_n}$$

for any $n \in \mathbb{N}$, as derived in the former section 2.2.1. Note that, as stated in theorem (12), the undecidability of problem (3) still holds if we restrict to diagonal matrices, i.e. $A \in \mathrm{Mat}_D(\mathrm{Mat}_d^{\mathrm{Diag}}(\mathbb{Z}))$. We shall make use of this fact and base the reduction on this sub-problem, i.e. we restrict to the diagonal instances. The respective tensor network polynomial of such an element then reads

$$\chi_n(\Phi(A)) = \sum_{i\in[d]^n} \langle i|\tau_n(A)|i\rangle (X_{i_1})^2 \dots (X_{i_n})^2.$$

From this equation, we can clearly deduce that the coefficients of $\chi_n(A)$ are precisely the diagonal entries of the network $\tau_n(A)$. Combined with the fact that those are the only non-vanishing entries of $\tau_n(A)$, this correspondence yields that

$$\tau_n(A) \geqslant 0 \Leftrightarrow \chi_n(\Phi(A)) \text{ is NNC},$$

at least for the diagonal case. Since this is true for any $n \in \mathbb{N}$ and the computability of $\Phi$ is immediate, this map is indeed a reduction of decision problems.

Since the reduction used in this proof maps exclusively onto polynomials of the form $\sum_{i=1}^d c^i X_i^2$, we conclude that the undecidability is valid even if we restrict the algebra to such polynomials. □

We have found that these structures - the positive semidefinite matrices on the one hand and polynomials with non-negative coefficients on the other - fit perfectly in the diagonal case. This was not the case for sum of square polynomials. Driven by this result, we shall do a thorough investigation of the structural issues in the next section. But before doing so, we want to put the NNC problem in a slightly different form.

**Definition 28** (Component-wise matrix-positivity (CMP)). *We call a polynomial $p \in \mathbb{A}\langle X_1, \dots, X_d\rangle$ component-wise matrix-positive iff for any $k \in \mathbb{N}$ and any non-negative matrices $M_1, \dots, M_d \in \mathrm{Mat}_k(\mathbb{A}_{\geq 0})$ the evaluation $p(M_1, \dots, M_d)$ is again a non-negative matrix.*

**Problem 6** (CMP). *Let $d, D \in \mathbb{N}$. Given a matrix $A \in \mathrm{Mat}_D(\mathbb{Z}\langle X_1, \dots, X_d\rangle)$, decide whether $\chi_n(A)$ is component-wise matrix-positive for all $n \in \mathbb{N}$.*

This problem is indeed equivalent to the NNC problem since the following theorem shows that component-wise matrix-positivity and non-negative coefficients are one and the same property. As the name suggests, it is in the style of Polya's theorem but for non-commutative polynomials instead of commutative ones.

**Theorem 29** (Polya's theorem; Non-commutative version). *Let $p \in \mathbb{A}\langle X_1, \dots, X_d\rangle$, then the following statements are equivalent:*

1. $p \in \mathbb{A}_{\geq 0}\langle \underline{X}\rangle$
2. *$p$ is component-wise matrix positive.*

*Proof.* The direction (1) $\Rightarrow$ (2) is clear since non-negative matrices are closed under the product and non-negative superpositions. To prove that (2) implies (1), we construct matrices that allow us to isolate a single coefficient. Consider a component-wise matrix positive polynomial $p$ with monomial representation

$$p = \sum_{n=0}^{\deg p} \sum_{\alpha \in [d]^n} p_\alpha X_{\alpha_1} \ldots X_{\alpha_n}.$$

For any $n \in \mathbb{N}$ and $\beta \in [d]^n$, we define the non-negative matrices $M_1, \ldots, M_d \in \mathrm{Mat}_{(n+1)}(\mathbb{A}_{\geq 0})$ by

$$M_j = \sum_{i=1}^{n} \delta_{j,\beta_i} |i\rangle\langle i+1|,$$

We obtain a compact expression for the product of $m$ such matrices by using the orthogonality of the basis vectors $|i\rangle$:

$$M_{j_1} \ldots M_{j_m} = \sum_{l=1}^{n-m+1} \left( \prod_{k=1}^{m} \delta_{j_k, \beta_{(l+k-1)}} \right) |l\rangle\langle l+m|$$

if $m \leq n$ and vanishing otherwise. As a consequence, the only product with a non-vanishing coefficient for $|1\rangle\langle n+1|$ is given by $j \equiv \beta$. Thus, we find that $p_\beta = \langle 1|p(M_1, \ldots, M_n)|n+1\rangle$ which is non-negative by the assumption that $p$ is component-wise matrix-positive. This completes the proof since $n$ and $\beta$ where choosen arbitrarily. $\square$

### 2.2.3 Reduction by vector space homomorphisms

We want to devote this section to a general study of the ansatz used so far. Our goal is to shed light on the underlying structure and to understand under which circumstances the ansatz allows for a reduction. This will allow us to establish a result of undecidability for certain graded algebras that posses a suitable free subalgebra.

In the following, let $V$ be a vector space over the algebraic numbers $\mathbb{A}$. To begin this discussion, we note that the map we used to assign a polynomial to each linear operator is a vector space homomorphism. We would like to construct an extension of this homomorphism that is defined on the full tensor power algebra of $\mathcal{L}(V)$. By restricting ourselves to extensions that are algebra homomorphisms, the following theorem ensures a unique and well-defined extension.

**Theorem 30.** *Let $V$ be a finite dimensional vector space and $\mathcal{A}$ an algebra, both over $\mathbb{A}$. Given a vector space homomorphism*

$$\varphi : \mathcal{L}(V) \to \mathcal{A},$$

*there is a unique extention $\Phi : \mathcal{L}(V)^{\otimes} \to \mathcal{A}$ of $\varphi$ to an algebra homomorphism.*

*Proof.* Let $d = \dim V$ and $\{E_i\}_{i\in[d^2]}$ be a basis of $\mathcal{L}(V)$. Then each $C \in \mathcal{L}(V)^{\otimes}$ can be uniquely expressed as a linear combination

$$C = \bigoplus_{n\in\mathbb{N}} \sum_{\omega\in[d^2]^n} c_\omega E_{\omega_1} \otimes \cdots \otimes E_{\omega_n}$$

with coefficients $c_\omega \in \mathbb{A}$. We define $\Phi : \mathcal{L}(V)^{\otimes} \to \mathcal{A}$ as the mapping

$$C \mapsto \sum_{n\in\mathbb{N}} \sum_{\omega\in[d^2]^n} c_\omega a_{\omega_1} * \cdots * a_{\omega_n}$$

where we introduced $a_i := \varphi(E_i)$. We will show in the following that this map has the desired properties. It is clearly an extension of $\varphi$ and well defined due to the uniqueness of the coefficients $\{c_\omega\}$. That $\Phi$ is indeed a homomorphism of algebras is also easy to check.

To show the uniqueness of this extension, suppose $\Phi' : \mathcal{L}(V)^{\otimes} \to \mathcal{A}$ is another extension of $\varphi$ to an algebra homomorphism. For any $C \in \mathcal{L}(V)^{\otimes}$ we find

$$\Phi'(C) = \sum_{n\in\mathbb{N}} \sum_{\omega\in[d^2]^n} c_\omega \Phi'(E_{\omega_1}) * \cdots * \Phi'(E_{\omega_n})$$

due to the properties of an algebra homomorphism. Since $\Phi'$ is an extension of $\phi$ we obtain

$$\Phi'(E_i) = \varphi(E_i) = a_i$$

and thus $\Phi' = \Phi$. □

The uniqueness of a structure is often important when we want to apply it in the vicinity of a reduction. This is mostly because we have to achieve an equivalence of the instances. Non-uniqueness may lead to inconveniencies in one direction. We will find a similar situation for representations of sum of square polynomials in the upcoming section. In the present case, the existence of such an extension allows for the following statement.

**Lemma 31.** *A homomorphism of vector spaces $\phi : \mathcal{L}(V) \to \mathcal{A}$ yields a reduction from problem (3) to the moments-membership problem for the algebra $\mathcal{A}$ equipped with the cone $\mathcal{C}$ if and only if its unique extension $\Phi : \mathcal{L}(V)^{\otimes} \to \mathcal{A}$ satisfies*

$$(\forall n \in \mathbb{N} : \tau_n(A) \geq 0) \Leftrightarrow (\forall n \in \mathbb{N} : \Phi(\tau_n(A)) \in \mathcal{C})$$

*for all instances $A \in \mathrm{Mat}_D(\mathcal{L}(V))$.*

*Proof.* We recall that a reduction has to be computable as well as faithful in the mapping of the property at hand. The map we are interested in is the component-wise action of $\phi$ on matrices over $\mathcal{L}(\mathcal{H})$, i.e.

$$\begin{aligned} \phi_D : \mathrm{Mat}_D(\mathcal{L}(\mathcal{H})) &\to \mathrm{Mat}_D(\mathcal{A}) \\ A &\mapsto \left(\phi(A_{\alpha\beta})\right)_{\alpha,\beta\in[D]} . \end{aligned}$$

The computability of this map follows from the computability of $\phi$ as a linear map with a finite dimensional domain space. The latter requirement yields the condition

$$(\forall n \in \mathbb{N} : \tau_n(A) \geq 0) \Leftrightarrow (\forall n \in \mathbb{N} : \chi_n(\phi_M(A)) \in \mathcal{C})$$

for this specific case. To obtain the claimed result, we use the unique tensor-algebra extension of $\phi$ which yields $(\phi_D(A))_{\alpha\beta} = \phi(A_{\alpha\beta}) = \Phi(A_{\alpha\beta})$ and thus

$$\begin{aligned} \chi_n(\phi_D(A)) &= \sum_{\alpha\in[D]^n} \phi_D(A)_{\alpha_1\alpha_2} * \cdots * \phi_D(A)_{\alpha_n\alpha_1} \\ &= \sum_{\alpha\in[D]^n} \Phi(A_{\alpha_1\alpha_2}) * \cdots * \Phi(A_{\alpha_n\alpha_1}) \\ &= \Phi\left(\sum_{\alpha\in[D]^n} A_{\alpha_1\alpha_2} \otimes \cdots \otimes A_{\alpha_n\alpha_1}\right) \end{aligned}$$

where we used the properties of an algebra homomorphism in the last step. This finishes the proof since the argument in the last line is precisely the tensor network $\tau_n(A)$. □

This lemma is essentially a reformulation of the reduction condition to fit better in the situation at hand. We can strengthen the condition in the lemma (31) and require for each $n \in \mathbb{N}$ separately that the extension of $\phi$ satisfies

$$\tau_n(A) \geq 0 \Leftrightarrow \Phi(\tau_n(A)) \in \mathcal{C}.$$

Though the *only if* gets lost in this step, we shall restrict to this smaller set of reductions since it is easier to deal with. The easiest case for which these conditions are satisfied is if $\Phi$ is an injective homomorphism and the cone is chosen to be the image cone $\Phi(\mathrm{PSD})$.

**Theorem 32.** *Let $\mathcal{A}$ be an algebra and $\phi : \mathcal{L}(\mathbb{A}^d) \hookrightarrow \mathcal{A}$. Problem (4) concerning the algebra $\mathcal{A}$ is undecidable for $d, D \geq 7$ and $\mathcal{C} = \Phi(\mathrm{PSD}_d^{\otimes})$ if the extension $\Phi$ is injective. This is true even when restricting to the image of diagonal operators.*

*Proof.* The statement is a direct consequence of lemma (31). The injectivity of $\Phi$ implies the equivalence of the statements

1. $\tau_n(A) \geq 0$ and
2. $\Phi(\tau_n(A)) \in \Phi(\mathrm{PSD}_d^{\otimes})$

for $n \in \mathbb{N}$ and $A \in \mathrm{Mat}_D(\mathcal{L}(\mathbb{A}^d))$. That this is also true for diagonal operators follows from the fact that the membership-in-PSD problem is undecidable for diagonal matrices. $\square$

Due to the structure of the matrix power algebra, we find that the extension $\Phi$ is injective if and only if its image is a free subalgebra in $\mathcal{A}$. That means, there is a generating set such that all their products are lineary independent. If such a free subalgebra exists, the vector space homomorphism $\varphi$ can be choosen appropriately to map onto its generators yielding an injective extension. Together with the stronger statement of theorem (32) for diagonal operators, this results in the following corollary.

**Corollary 33.** *Let $\mathcal{A}$ be a free algebra generated by $d$ elements $g_1, \ldots, g_d$ and $\mathcal{C}$ the cone of non-negative combinations of their products. The corresponding moments-membership problem is undecidable for $D, d \geq 7$.*

We have already encountered an instance with applicability of this corollary for polynomial algebras in theorem (27). However, if we are interested in sum of squares polynomials, this theorem won't apply that easily since this cone is not closed under algebra multiplication, i.e. a product of sum of squares polynomials is not necessarily sum of squares. To be able to apply this result, we have to find a sub-cone in the sum of squares which is closed under multiplication and generated by at least 7 lineary independent elements.

**Lemma 34.** *Let $\mathcal{C}$ be a non-empty convex subcone of the sum of squares closed under algebra multiplication. If $\mathcal{C}$ is, as a convex cone, finitely generated by homogeneous elements of equal degree, then it is minimally generated by a single element.*

We will give a proof of this statement in the next subsection after establishing a duality between sum of squares and positive semidefinite matrices. For now, we are more concerned with its implication. Since the precise map of linear operators to polynomials assigns homogeneous elements of degree 2, the considered cone has to consist of elements generated by those. Due to lemma (34), which requires such a cone to be generated by a single element, this map does not fulfill the considerations of corollary (33).

However, there are a lot of examples that satisfy the conditions of this corollary. To use that argument, we have to extract a free sub-algebra with a set of independent generators such that the problem in this subspace is about non-negative coefficients. As already mentioned, one example is the algebra generated by the monomials $\{X_i X_i\}_{i\in[7]}$ with the question about non-negative coefficients. Other examples can easily be constructed by means of tensor power algebras. Let $V$ be a $\mathbb{A}$-vector space of dimension $d \geq 7$ and $V^{\otimes} = \bigoplus_{k=1}^{\infty} V^{\otimes k}$ its tensor power algebra. Then for any basis $\{v_i\}_{i\in[d]}$, the problem of determining whether the $V^{\otimes}$-valued tensor network $\chi_n$ has non-negative coefficients with respect to the induced basis is undecidable. Consider for example the tensor algebra $\mathrm{Mat}_3(\mathbb{A})^{\otimes}$ generated by $3 \times 3$-matrices. In this specific case, the basis of standard matrices yields the associated cone of non-negative matrices. This is because the tensor product basis corresponding to the standard basis is again a standard basis in a higher dimension. Corollary (33) applies to this case since the vector space of $3 \times 3$-matrices is 9-dimensional.

### 2.2.4 Representing SOS polynomials by PSD matrices

In this section, we shall construct a representation of sum of square polynomials by positive semidefinite matrices. We will show that this representation is unique and well-defined for homogeneous polynomials, i.e. there is a bijection between the homogeneous sum of squares of degree $2n$ and positive semidefinite matrices of a certain dimension.

From now on consider the non-commutative polynomials in complex indeterminates $\mathcal{A} = \mathbb{A}\langle \underline{X}, \underline{X}^* \rangle$. This is in contrast to the foregoing discussion, where we dalt with the trivial involution. For the sake of better readability, we will denote the conjugate indeterminates by $\overline{X}_i$ - similar to complex conjugation - rather than $X_i^*$. We denote the family of monomials of degree less or equal to $n$ and degree equal $n$ by the column vectors

$$
\begin{aligned}
M^n &:= (1, X_1, \ldots, \overline{X}_d, X_1 X_1, \ldots, \overline{X}_d^n) \\
H^n &:= (X_1^n, X_1^{n-1} X_2, \ldots, \overline{X}_d^{n-1} \overline{X}_{d-1}, \overline{X}_d^n)
\end{aligned}
$$

respectively.

**Lemma 35.** *Let $p = \sum_{i=1}^r \overline{p_i} p_i$ be a sum of squares polynomial, then*

$$\deg p = 2 \cdot \max \{\deg p_i \mid i \in [r]\}.$$

*Proof.* We already know that the degree of a sum of polynomials is at most the maximal degree of the summands. We will show that this identity is saturated for sums of squares. Consider the polynomial $p$ as described above and let $q$ be one of the polynomials $p_i$ in the respective SOS decomposition. We denote the highest-degree part of $q$ by $q_m = \sum_\alpha q_m^\alpha H_\alpha^m$ with $m = \deg(q)$. The contribution of highest degree to $\overline{q}q$, i.e. the part with degree $2m$, is fully specified by $q_m$:

$$\sum_{\alpha\beta} \overline{H}_\alpha^m \underbrace{\overline{q_m^\alpha} q_m^\beta}_{:=K^{\alpha\beta}} H_\beta^m.$$

It is immediate that the matrix $(K^{\alpha\beta})_{\alpha,\beta}$ is positive semidefinite and non-vanishing. Due to our interest in the contribution of degree

$$2\mathfrak{m} := 2 \max \{\deg p_i \mid i \in [r]\}$$

, we discard the polynomials $p_i$ with $\deg(p_i) \neq \mathfrak{m}$. The degree $2\mathfrak{m}$ contribution to $p$ then is represented by the sum of the non-vanishing and positive semidefinite matrices $(K^{\alpha\beta})_{\alpha,\beta}$. Such a sum is again non-vanishing and we thus conclude $\deg(p) = 2\mathfrak{m}$ as claimed. □

We will use this lemma to prove the following theorem which establishes a correspondence between sum of square polynomials and positive semidefinite matrices. The rest of the section will then be devoted to the question of uniqueness.

**Theorem 36.** *Let $p \in \mathbb{A}\langle \underline{X}, \overline{\underline{X}}\rangle$ of degree $2n$. Then $p$ is sum of squares if and only if there is a matrix $G \geqslant 0$ such that*

$$p = \sum_{\alpha\beta} \overline{M}^n_\alpha G^{\alpha\beta} M^n_\beta.$$

*Proof.* We will show both directions separately:

"$\Leftarrow$" Assume there is a matrix $G \geqslant 0$ such that the above is true. By the spectral theorem $G$ can be decomposed as

$$G = \sum_\lambda \lambda P_\lambda$$

with $P_\lambda$ the $\lambda$-eigenspace projection. We thus find that

$$p = \sum_{\alpha\beta} \sum_\lambda \lambda \overline{M}^n_\alpha P^{\alpha\beta}_\lambda M^n_\beta.$$

Using that the eigenvalues of a positive semidefinite matrix are nonnegative as well as the properties $P^2_\lambda = P_\lambda$ and $P^\dagger_\lambda = P_\lambda$ of orthonormal projectors yields the representation

$$p = \sum_\lambda \sum_{\alpha\beta\gamma} \lambda \overline{M}^n_\alpha P^{\alpha\gamma}_\lambda P^{\gamma\beta}_\lambda M^n_\beta = \sum_\lambda \sum_\gamma \overline{\left( \sum_\alpha \sqrt{\lambda} P^{\gamma\alpha}_\lambda M^n_\alpha \right)} \underbrace{\left( \sum_\beta \sqrt{\lambda} P^{\gamma\beta}_\lambda M^n_\beta \right)}_{:= p^\gamma_\lambda}.$$

We conclude that $p = \sum_{\lambda\gamma} \overline{p}^\gamma_\lambda p^\gamma_\lambda$ is sum of squares.

"$\Rightarrow$" On the other hand, consider a sum of squares polynomial $p = \sum_i \overline{p_i} p_i$ of degree $2n$. We want to show that it can be represented by a matrix $G \geq 0$. Due to $\deg p_i \leq n$ we can use the representation $p_i = \sum_\alpha p^\alpha_i M^n_\alpha$. The polynomial $p$ then takes the form

$$\begin{aligned} p &= \sum_i \sum_{\alpha\beta} \overline{M}^n_\alpha \overline{p^\alpha_i} p^\beta_i M^n_\beta \\ &= \sum_{\alpha\beta} \overline{M}^n_\alpha \underbrace{\left( \sum_i \overline{p^\alpha_i} p^\beta_i \right)}_{:= G^{\alpha\beta}} M^n_\beta. \end{aligned}$$

It is immediate to see that $G$ is positive semidefinite.

□

Such a positive semidefinite representation matrix is called a **Gram matrix** of the polynomial. Unfortunately, the theorem only states the existence of such a Gram representation, but it is not necessarily unique. Moreover, not each representation by a matrix has to be a positive semidefinite one. However, we are mainly interested in homogeneous polynomials due to their appearence in the reduction. In the following, we will prove the uniqueness of the Gram representation for this specific case. The representations of the zero-polynomial, not necessarily positive semidefinite, will take an important part in this proof.

**Definition 37** (Zero-mode)**.** *We call a Hermitian matrix $\sigma_n$ that represents the zero-polynomial, i.e.*

$$0 = \sum_{\alpha\beta} \overline{M}^n_\alpha \sigma_n^{\alpha\beta} M^n_\beta,$$

*a **zero-mode** (of n-th order).*

These zero-modes arise due to the linear dependencies in the collection of monomials $\{\overline{M}^n_\alpha M^n_\beta\}_{\alpha\beta}$. The reason why they are important to us is that they introduce an ambiguity to the representation of a polynomial. Adding a zero-mode to a representation corresponds to adding the zero polynomial. However, since the zero-modes are Hermitian but not necessarily positive semidefinite, the resulting matrix might not be PSD. In case of a single complex indeterminate, we find the first non-trivial representation of the zero-polynomial for $n = 1$ with

$$\sigma_1 = \begin{pmatrix} 0 & a & -b \\ b & 0 & 0 \\ -a & 0 & 0 \end{pmatrix}$$

for arbitrary $a, b \in \mathbb{A}$. It is immediate that this is Hermitian, thus a zero-mode, if and only if $a = \bar{b}$. The number of zero-representations depends on the number of monomials in $\{\overline{M}^n_\alpha M^n_\beta\}_{\alpha\beta}$ and the actual number of independent monomials up to degree $2n$. However, due to the structure of these representations, there is a unique positive semidefinite zero-mode for each order.

**Lemma 38.** *For any order, the only positive semidefinite zero-mode is the zero-matrix.*

*Proof.* Let $\sigma_n$ be a positive semidefinite $n$-th order zero-mode. The assumption $\sigma_n \neq 0$ implies the existence of a positive eigenvalue $\lambda > 0$. Following the construction of theorem (36), there is a representation of the zero-polynomial as a sum of squares of non-vanishing polynomials. This, however, is in contradiction with lemma (35). □

This result is important for our goal of proving the uniqueness of a representation. The existence of a non-trivial positive semidefinite zero-mode would be an obstruction for unique representations. This is because the sum of a

Gram representation and this zero-mode would always result in a new distinct Gram representation. Thus, there would be infinite family of such valid representations.

**Lemma 39.** *Let $p \in \Sigma^2$ be a sum of squares polynomial and $G_1, G_2$ Gram representations of $p$. The difference $G_1 - G_2$ is a zero-mode.*

*Proof.* To begin with, note that the difference $\sigma = G_1 - G_2$ is Hermitian since both $G_1$ and $G_2$ are. Moreover, $\sigma$ yields a representation of the zero polynomial due to

$$\sum_{\alpha\beta} \overline{M}^n_\alpha \sigma^{\alpha\beta} M^n_\beta = \sum_{\alpha\beta} \overline{M}^n_\alpha \left( G_1^{\alpha\beta} - G_2^{\alpha\beta} \right) M^n_\beta = p - p = 0.$$

Thus, $\sigma$ is a zero mode. □

In the following, we focus on representations of homogeneous polynomials due to their appearance in the tensor network after the reduction. First, we establish a useful characterization of the affine sections corresponding to homogeneous sum of squares polynomials.

**Lemma 40.** *Let $p$ be a homogeneous sum of squares polynomial of degree $2n$. Each $n$-th order representation $G$ of $p$ is of the form $G = \mathbf{0} \oplus K + \sigma_n$ with a hermitian matrix $K$ of dimension $(2d)^n$ and $\sigma_n$ a zero-mode.*

*Proof.* Since $p$ is a homogeneous polynomial and the set of monomials $\{\overline{H}^n_\alpha H^n_\beta\}_{\alpha\beta}$ is a basis of those, there is a unique representation $p = \sum_{\alpha\beta} \overline{H}^n_\alpha K^{\alpha\beta} H^n_\beta$. Moreover, a sum of squares polynomial is symmetric which yields that the matrix $K$ is hermitian. As stated by theorem (39) all other representations are obtained by adding a zero-mode. □

We are now able to prove the main theorem of this section, namely the uniqueness of the Gram representation of homogeneous sum of squares polynomials.

**Theorem 41.** *Homogeneous sum of squares polynomials are sum of squares of homogeneous polynomials. Moreover, the homogeneous sum of squares of degree $2n$ are in bijective relation to $PSD_{(2d)^n}$.*

*Proof.* Let $p$ be a homogeneous sum of squares polynomial and $G = \mathbf{0} \oplus K + \sigma_n$ a Gram representation of $p$. We will show that $\sigma_n$ is the trivial zero-mode.

Since the Gram representation $G$ is Hermitian, it can be decomposed in a block matrix form

$$G = \begin{pmatrix} A & B \\ B^\dagger & Q \end{pmatrix}$$

where $A$ and $Q$ are again Hermitian. Here, $Q$ represents the coefficients of the heighest degree $2n$, i.e. $Q$ has the same size as $K$. Moreover, $A$ can itself be shown to be a zero-mode of degree $n-1$ by counting the number of dependent coefficients in the Gram representation. Since the monomials $\{\overline{H}^n_\alpha H^n_\beta\}_{\alpha\beta}$ are each contained once in the family $\{\overline{M}^n_\alpha M^n_\beta\}_{\alpha\beta}$, their coefficients in the zero-mode $\sigma_n$ have to vanish. This precisely affects the block part $Q$ and we find that $Q = K$.

In the next step, we use that $G$ is positive semidefinite which yields

$$\langle v|G|v\rangle \geq 0$$

for all vectors $v$. Applying the direct sum decomposition on the vector $v = v_1 \oplus v_2$ leads to the inequality

$$\langle v_1|A|v_1\rangle + 2\,\mathrm{Re}\langle v_1|B|v_2\rangle + \langle v_2|K|v_2\rangle \geq 0$$

for arbitrary $v_1$ and $v_2$. Since $A$ is a zero-mode, it cannot be positive semidefinite and non-vanishing at the same time. Assume $A \neq 0$, then there is a vector $n$ such that $\langle n|A|n\rangle < 0$. We insert the vector $v_1 := \mu n$ with $\mu > 0$ in the inequality above to obtain

$$\mu^2\langle n|A|n\rangle + 2\mu\,\mathrm{Re}\langle n|B|v_2\rangle + \langle v_2|K|v_2\rangle \geq 0.$$

This inequality is supposed to hold for all $\mu$. However, due to the leading coefficient $\langle n|A|n\rangle$ being negative the inequality will be violated for $\mu$ large enough. We conclude that $A$ has to vanish for a valid Gram representation.

Concerning the off-diagonal terms $B$, we use a similar argument. With $A$ set to the zero matrix, the inequality reads

$$2\,\mathrm{Re}\langle v_1|B|v_2\rangle + \langle v_2|K|v_2\rangle \geq 0$$

for arbitrary vectors $v_1$ and $v_2$. We focus on the first term and consider the possibilities:

(i) $\mathrm{Re}\langle v_1|B|v_2\rangle < 0$: The inequality is violated for $\tilde{v}_1 := \mu v_1$ with $\mu > 0$ large enough.

(ii) $\mathrm{Re}\langle v_1|B|v_2\rangle > 0$: Reduces to case (i) for $v_i \to -v_i$.

(iii) $\mathrm{Re}\langle v_1|B|v_2\rangle = 0$: No violation of the inequality.

We conclude that $\mathrm{Re}\langle v_1|B|v_2\rangle = 0$ for all $v_1$ and $v_2$. Finally, $B = 0$ follows directly from this statement. Thus, $G = \mathbf{0} \oplus K$ is the unique positive semidefinite representation of the homogeneous sum of squares polynomial $p$. □

This is a powerful result because it sets homogeneous sum of squares in exact correspondence with positive semidefinite matrices. We will devote the following section to its implications to the moments-membership problem. But before that, we shall give a proof for lemma (34). As a reminder, this lemma states that if a convex subcone of the SOS polynomials is closed under algebra multiplication and generated by lineary independent homogeneous elements of the same degree, then it is generated by a single element.

*Proof.* Let $q, p \in \Sigma_k^2$ be homogeneous generators of a convex subcone of $\Sigma^2$. According to the assumption of closedness under algebra multiplication, the product $q \cdot p$ is an element of this subcone and thus a sum of squares. If we represent the generators by $q = \sum_\alpha q^\alpha \overline{H_\alpha^k}$ and $p = \sum_\beta p^\beta H_\beta^k$, their product yields

$$qp = \sum_{\alpha\beta} \overline{H_\alpha^k} q^\alpha p^\beta H_\beta^k.$$

Due to the assumptions, $qp$ is a homogeneous sum of square polynomial and thus, according to theorem (41), $(G^{\alpha\beta})_{\alpha,\beta} = (q^\alpha p^\beta)_{\alpha,\beta}$ is its unique Gram representation. However, $G$ is positive semidefinite if and only if $q^\alpha = c \cdot \overline{p}^\beta$ with $c > 0$. Since sum of squares are self-adjoint, this is a contradiction to the assumption of linear independence. $\square$

This proves the statement for sum of squares in $\mathbb{A}\langle \underline{X}, \underline{X}^* \rangle$. A similar result is obtained for $\mathbb{A}\langle \underline{X} \rangle$ by adopting the treatment of this section. This is essentially achieved by substitution of the monomial vectors

$$\begin{aligned} M^n &= (1, X_1, \ldots, X_d, X_1 X_1, \ldots, X_d^n) \\ H^n &= (X_1^n, X_1^{n-1} X_2, \ldots, X_d^n) \end{aligned}$$

and repeat the same proof strategies. The main deviations are the dimensions of the object. As a consequece, the homogeneous sum of square polynomials of degree $n$ will then correspond to positive semidefinite matrices of dimension $d^n$.

### 2.2.5 Membership in $\Sigma^2$ revisited

In the following, we shall apply the results of the former section to the moments-membership problem for sum of squares. The Gram representation will allow us to see more clearly the precise problem, i.e. the incompatibility. We will see that this is due to a heavy intertwining of the indices in the matrix representation.

For better applicability of the results on Gram representations in the former section, we shall consider the polynomials $\mathbb{A}\langle \underline{X}, \overline{X} \rangle$ with a non-trivial $*$-structure. Thus, $X_i$ and its conjugate $\overline{X}_i$ are distinct elements. This is respected in the homomorphism ansatz by conjugation of the former monomial:

$$\varphi : A \mapsto \left\{ \sum_i A^i_{\alpha\beta} \overline{X}_i X_i \right\}_{\alpha,\beta \in [D]}.$$

Note that we directly work in the diagonal case $A \in \mathrm{Mat}_D(\mathrm{Mat}_d^{\mathrm{Diag}}(\mathbb{A}))$. Recall that the tensor network $\chi_n$ of such a local tensor reads

$$\begin{aligned}\chi_n(\varphi(A)) &= \sum_{\alpha\in[D]^n} \varphi(A)_{\alpha_1\alpha_2}\dots\varphi(A)_{\alpha_n\alpha_1} \\ &= \sum_{i\in[d]^n} \mathrm{tr}(A^{i_1}\dots A^{i_n})\overline{X}_{i_1}X_{i_1}\dots\overline{X}_{i_n}X_{i_n}.\end{aligned}$$

To work out the Gram representation, we have the rearrange the monomials slightly. This will yield different results depending on whether $n$ is even or odd. In the following, we will focus on the even case and set $n = 2m$. We put the first half of the monomial in a conjugate form to obtain

$$\chi_n(\varphi(A)) = \sum_{i\in[d]^n} \langle i|\tau_n(A)|i\rangle \overline{\overline{X}_{i_m}X_{i_m}\dots\overline{X}_{i_1}X_{i_1}}\,\overline{X}_{i_{m+1}}X_{i_{m+1}}\dots\overline{X}_{i_{2m}}X_{i_{2m}},$$

where $\langle i|\tau_n(A)|i\rangle = \mathrm{tr}(A^{i_1}\dots A^{i_n})$ was used. This is essentially the unique Gram representation, but with strongly intertwined indices. As a result, already self-adjointness of the network imposes relations on the matrix-entries of $\langle i|\tau_n(A)|i\rangle$. Conjugation of the network polynomial corresponds to an index swapping of $i_1 \leftrightarrow i_{2m}$, $i_2 \leftrightarrow i_{2m-1}$ and so on. Thus, positive semidefiniteness of $\langle i|\tau_n(A)|i\rangle$ doesn't even suffice for self-adjointness of $\chi_n(\phi(A))$. With this last insight, we shall leave the problem of membership in the sum of squares unsolved.

## 2.3 The moments-membership problem for commutative algebras

In this section, we will investigate the moments-membership problem in the vicinity of commutative algebras. In the first part, we will derive identities that are related to the Newton identities [32] but also valid for algebraically non-closed algebras. These in turn will help us establish a recursion formula for the moments, namely

$$\mu_n = -\sum_{k=1}^{D} \frac{1}{k!}\mu_{n-k}B_k(-0!\mu_1, -1!\mu_2, \dots, -(k-1)!\mu_k)$$

for $n > D$, where $B_k(x_1, \dots, x_k)$ is the $k$-th complete Bell polynomial.

### 2.3.1 The moments of matrices over a commutative ring

In the following, we will consider commutative rings instead of algebras because this is the crucial structure for this investigation. The results will then be applied to commutative, associative, unital algebras, since those carry a ring structure with addition and algebra multiplication. In the following, we consider $R$ to be a commutative ring and $\mathrm{Mat}_n(R)$ the associated ring of $n \times n$ matrices over $R$. Since there are multiple different conventions for the precise structure of a ring, we recall the definition used in this work.

**Definition 42** (Ring)**.** *A **ring** $(R, +, *)$ is a group $(R, +)$ together with an operation $* : R \times R \to R$ such that:*

*(R1) there is a unit element $1 \in R$, i.e. $\forall a \in R : 1 * a = a = a * 1$,*

*(R2) the product $*$ is associative, i.e. $\forall a, b, c \in R : a * (b * c) = (a * b) * c$ and*

*(R3) the product $*$ is distributive, i.e. $\forall a, b, c : a * (b + c) = a * b + a * c$ as well as $(a + b) * c = a * c + b * c$.*

A central position in this section will be taken by the characteristic polynomial of matrices over the ring $R$. It is defined the same way as it is for the number rings $\mathbb{Q}, \mathbb{R}$ or $\mathbb{C}$, but for completeness we will begin by recalling the main definitions. For a commutative ring we begin by introduction of the **determinant** mapping $\det : \mathrm{Mat}_n(R) \to R$. In the same way as for the regular number rings, this is defined by the combinatorial sum

$$\det(A) := \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) A_{1,\sigma(1)} A_{2,\sigma(2)} \dots A_{n,\sigma(n)}. \tag{2.3}$$

The **characteristic polynomial** is then defined by the following determinant:

$$\begin{aligned} \mathfrak{ch} : \mathrm{Mat}_n(R) &\to R[t] \\ A &\mapsto \det(t \cdot \mathrm{Id} - A). \end{aligned}$$

To each matrix $A \in \mathrm{Mat}_n(R)$, this map assigns a polynomial $\mathfrak{ch}_A \equiv \mathfrak{ch}(A)$ in one variable which we call the characteristic polynomial of $A$. In the case of the complex numbers, this polynomial is well known to encode the eigenvalues of the matrix by means of its roots. However, the roots are connected to the moments of the matrix only if the characteristic polynomial splits in linear factors. If the considered ring $R$ is not algebraically closed, meaning that polynomials over $R$ do not split in linear factors in general, we have to consider the algebraic closure of $R$ to connect the roots to the moments. Consider for example the ring $R = \mathbb{Z}$ of integers and the matrix

$$A = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}.$$

We find its characteristic polynomial $\mathfrak{ch}_A = t^2 - 2$, which has no root in $\mathbb{Z}$. If we consider the same matrix over the real numbers $R = \mathbb{R}$ we find that $\pm\sqrt{2}$ are roots of $\mathfrak{ch}_A$ and they yield a spliting in linear factors $\mathfrak{ch}_A = (t - \sqrt{2})(t + \sqrt{2})$. However, $\mathbb{R}$ is still not algebraically closed. An example of a polynomial without any roots in $\mathbb{R}$ is the polynomial $x^2 + 1$. The algebraic closure of $\mathbb{Z}$ and $\mathbb{Q}$ is a subring of the complex numbers $\mathbb{C}$ that we have introduced above - the algebraic numbers $\mathbb{A}$.

Instead of working with the algebraic closure of the ring $R$, we will make use of another property of the characteristic polynomial. As we will see in the following, the polynomial $\mathfrak{ch}_A$ does not only specify the eigenvalues of $A$ if $R$ is a field, but it also contains all information needed to reconstruct all the moments of the matrix $A$, even if $\mathfrak{ch}_A$ does not split in linear factors. Before that, however, we shall prove the reverse statement in form of the following theorem.

**Theorem 43.** *Let $R$ be a ring containing the rational numbers. The characteristic polynomial of $A \in \mathrm{Mat}_D(R)$ is fully specified by its first moments $\mu_1^A, \dots, \mu_D^A$:*

$$\mathfrak{ch}_A = \sum_{k=0}^{D} \frac{1}{k!} B_k(-0!\mu_1^A, -1!\mu_2^A, \dots, -(k-1)!\mu_k^A) t^{D-k}, \tag{2.4}$$

*where $B_k$ are the complete exponential Bell polynomials.*

*Proof.* To prove the statement we will use the product form of the determinant and separate it in different classes of permutations. More precisely, we will expand in powers of $t$ in the first step to then rearrange the permutations according to their decomposition in cycles:

$$\begin{aligned}
\mathfrak{ch}_A &= \det(t \cdot I - A) \\
&= \sum_{\sigma \in S_D} \mathrm{sgn}(\sigma) \prod_{i=1}^{D} \left( t\delta_{i,\sigma(i)} - A_{i,\sigma(i)} \right) \\
&= \sum_{n=0}^{D} \frac{1}{n!} \sum_{\alpha \in [D]^n} t^{D-n} \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} \left( -A_{\alpha_i, \alpha_{\sigma(i)}} \right).
\end{aligned}$$

In the last expression, the image of $\alpha$ is the set of indices that did not contribute to the power of $t$ and the factor of $(n!)^{-1}$ is due to redundancy in this sum. In the next step we will use the fact that every permutation can be uniquely decomposed in disjoint cycles. We will denote these cycles by $\zeta = (\zeta_1 \dots \zeta_k)$, meaning that $\zeta_1$ is mapped to $\zeta_2$ and so on. If $\zeta$ is a cycle in the decomposition of a permutation $\sigma$, we denote this by $\zeta \lhd \sigma$. By making use of this decomposition, we obtain the representation

$$\mathfrak{ch}_A = \sum_{n=0}^{D} \frac{(-1)^n}{n!} \sum_{\alpha \in [D]^n} t^{D-n} \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{(\zeta_1 \dots \zeta_k) \lhd \sigma} \prod_{i=1}^{k} A_{\alpha_{\zeta(i)} \alpha_{\zeta(i+1)}}.$$

To proceed, we pull the sum over the indices $\alpha$ into the product over cyclic components of the permutation. This yields a representation of the characteristic polynomial solely by the moments of the matrix:

$$\mathfrak{ch}_A = \sum_{n=0}^{D} \frac{(-1)^n}{n!} t^{D-n} \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{(\zeta_1 \dots \zeta_k) \lhd \sigma} \underbrace{\sum_{\alpha \in [D]^k} A_{\alpha_1,\alpha_2} A_{\alpha_2,\alpha_3} \cdots A_{\alpha_k,\alpha_1}}_{\mu_k}.$$

In the last part of the proof we want to determine the coefficients of this expansion in terms of combinatiorics. There is some redundancy since the summand in the sum over permutations is dependent only on the length of its cyclic components. We can replace this sum by a double sum, the first covering the number of disjoint cycles followed by one specifying the length of those. Here we also use that the sign of a permutation is the product of the signs of its cyclic components. Those are determined by $(-1)^k$ where $k$ is the length of the cycle. Thus, we obtain the expression

$$\mathfrak{ch}_A = \sum_{n=0}^{D} \frac{(-1)^n}{n!} t^{D-n} \sum_{k=1}^{n} \sum_{i} \mathfrak{f}(k,i) \prod_{l=1}^{n-k+1} \left((-1)^{l+1}\mu_l\right)^{i_l}.$$

with a family of combinatorical factors $\mathfrak{f}(k,i)$ yet to be determined. Here, the index $i_l$ with $l \in [n-k+1]$ specifies how many cycles of length $l$ appear in the permutation and thus the number of $l$-th moments we obtain. The sum of these indices $i_1, \dots, i_{n-k+1}$ has to obey

$$i_1 + i_2 + \cdots + i_{n-k+1} = k,$$

since the total number of cycles is $k$, as well as

$$1i_1 + 2i_2 + \cdots + (n-k+1)i_{n-k+1} = n.$$

To proceed, we have to identify the combinatorical factor $\mathfrak{f}$. It counts the number of disjoint permutations that consist of $k$ disjoint cycles with length and multiplicity determined by $i_1, \dots, i_{n-k+1}$. The factor arises since these permutations lead to the same combination of moments. The factor $\mathfrak{f}$ can be split in three contributions:

1. The number of ways a permutation of length $n$ can be seperated in partitions $n-k+1$ of lengths $i_l \cdot l$, $l \in [n-k+1]$. This is given by the multinomial coefficient

   $$\mathfrak{f}_1 = \binom{n}{m_1, \dots, m_{n-k+1}} := \frac{n!}{m_1! \dots m_{n-k+1}!}$$

   with $m_l$ set to $i_l \cdot l$.

2. The number of ways this partitions of length $i_l \cdot l$ can be distributed in $i_l$ partitions of length $l$:

   $$\mathfrak{f}_2 = \prod_l \left[\frac{1}{i_l!}\binom{i_l \cdot l}{l, \dots, l}\right].$$

3. The number of different cyclic permutations of a set of length $l$ (for each cycle contributing to the permutation):

   $$\mathfrak{f}_3 = \prod_l [(l-1)!]^{i_l}.$$

By putting these together and doing some cancelations, we find the combinatorical factor to be

$$\mathfrak{f} = \frac{n!}{i_1! \cdots i_{n-k+1}!} \prod_l \left[ \frac{(l-1)!}{l!} \right]^{i_l}.$$

The characteristic polynomial therefore reads

$$\mathfrak{ch}_A = \sum_{n=0}^{D} \frac{1}{n!} t^{D-n} \sum_{k=1}^{n} \sum_i \frac{n!}{i_1! \cdots i_{n-k+1}!} \prod_{l=1}^{n-k+1} \left( \frac{-(l-1)!\mu_l}{l!} \right)^{i_l},$$

where we also rearranged the powers of $(-1)$. At this point we can identify the coefficients with the complete exponential Bell polynomials. These are defined by

$$B_n(x_1, \ldots, x_n) := \sum_{k=1}^{n} \sum_i \frac{n!}{i_1! \cdots i_{n-k+1}!} \prod_{l=1}^{n-k+1} \left( \frac{x_l}{l!} \right)^{i_l}$$

with the same index set for $i$ as before; not to forget the implicit dependence on $k$. By using these together with the convention $B_0 \equiv 1$, the characteristic polynomial can be rewritten in the form

$$\mathfrak{ch}_A = \sum_{n=0}^{D} \frac{1}{n!} t^{D-n} B_n(-0!\mu_1, \ldots, -(n-1)!\mu_n)$$

as claimed. □

For a better overall view, we will avoid the messy arguments $-(k-1)!\mu_k$ of the Bell polynomials and introduce the coefficient functions

$$C_k(\mu) \equiv C_k(\mu_1, \ldots, \mu_k) := \frac{1}{k!} B_k(-0!\mu_1, -1!\mu_2, \ldots, -(k-1)!\mu_k). \tag{2.5}$$

The Bell polynomials have a visual interpretation in combinatorics. The coefficient of a monomial $x_1^{i_1} \ldots x_k^{i_k}$ is precisely the number of disjoint partitions of

$$n = i_1 \cdot 1 + \cdots + i_k \cdot k$$

objekts in $i_l$ sets of size $l$. In a concrete example this means for $B_4$ that the coefficient of the monomial $x_1 x_3$ is 4 since there are 4 possible partitions of $\{1,2,3,4\}$ in two sets of size 1 and 3:

$$\{1\} \cup \{2,3,4\}, \{2\} \cup \{1,3,4\}, \{3\} \cup \{1,2,4\} \text{ and } \{4\} \cup \{1,2,3\}.$$

The calculation of the coefficient using the definition given above yields the same result:

$$\frac{4!}{1! \cdot 0! \cdot 1!} \cdot \left( \frac{1}{1!} \right)^1 \cdot \left( \frac{1}{3!} \right)^1 = \frac{4!}{3!} = 4.$$

This interpretation is connected to the way we proved the former theorem. We partitioned the product $A_{1,\sigma(1)} \dots A_{n,\sigma(n)}$ in disjoint subsets - more specifically into the disjoint cycles. The factor of $(l-1)!$ in the arguments arises due to the fact that each partition of length $l$ represents $(l-1)!$ different cycles.

TABLE 2.1: List of the first complete Bell polynomials

| $n$ | $B_n(x_1, \dots, x_n)$ |
|---|---|
| 0 | 1 |
| 1 | $x_1$ |
| 2 | $x_1^2 + x_2$ |
| 3 | $x_1^3 + 3x_1x_2 + x_3$ |
| 4 | $x_1^4 + 6x_1^2x_2 + 4x_1x_3 + 3x_2^2 + x_4$ |
| 5 | $x_1^5 + 10x_1^3x_2 + 15x_1x_2^2 + 10x_1^2x_3 + 10x_2x_3 + 5x_1x_4 + x_5$ |
| ... | ... |

From table (2.1) it is immediate that $\mathfrak{ch}_A$ is a monic polynomial, i.e. the highest degree coefficient is the unit of $R$, due to the zeroth Bell polynomial $B_0 = 1$ being the highest order coefficient. It is the representation

$$\mathfrak{ch}_A = \sum_{k=0}^{D} t^{D-k} C_k(\mu)$$

together with the following theorem of Cayley-Hamilton which let us establish the recursion formula for the moments as stated in the introduction of this section.

**Theorem 44** (Cayley-Hamilton [21])**.** *Each matrix $A \in \mathrm{Mat}_D(R)$ satisfies its own characteristic polynomial, i.e. $\mathfrak{ch}_A(A)$ is the zero matrix.*

It is immediate that taking the trace of a polynomial in the matrix $A$ yields a linear expression in the moments of $A$:

$$\mathrm{Tr}[p(A)] = \sum_{k=0}^{n} p_k \, \mathrm{Tr}[A^k] = \sum_{k=0}^{n} p_k \mu_k.$$

Since the coefficients of the characteristic polynomial are themselves polynomials in the moments $\mu_1, \dots \mu_D$, the trace maps the equation $\mathfrak{ch}_A(A) = 0$ onto a polynomial equation in the moments. However, this equation turns out to be trivial. We resolve this issue by multipling $\mathfrak{ch}_A(A)$ with a power of $A$ before taking the trace, which yields a non-trivial algebraic equation. Since $\mathfrak{ch}_A$ is monic and the coefficients $B_k(-0!\mu_1, \dots, -(k-1)!\mu_k)$ do depend on the $D$-th moment at most, we obtain a algebraic equation that can be resolved for the highest present moment, as done in the following theorem.

**Theorem 45.** *The moments of $A \in \mathrm{Mat}_D(R)$ are fully specified by the first $D$ moments $\mu_1, \ldots, \mu_D$. They satisfy the algebraic recursion formula*

$$\mu_n = -\sum_{k=1}^{D} \mu_{n-k} C_k(\mu_1, \ldots, \mu_k) \tag{2.6}$$

*for $n > D$.*

*Proof.* The theorem of Cayley-Hamilton applied on the representation (2.4) of the characteristic polynomial yields

$$\mathfrak{ch}_A(A) = \sum_{k=0}^{D} A^{D-k} C_k(\mu_1, \ldots, \mu_k) = 0$$

for any $A \in \mathrm{Mat}_D(R)$. Multiplication by $A^{n-D}$ with $n > D$ results in

$$\mathfrak{ch}_A(A) A^{n-D} = \sum_{k=0}^{D} A^{n-k} C_k(\mu_1, \ldots, \mu_k) = 0.$$

As last step, we apply the trace to this equation and use its linearity to obtain the algebraic equation

$$\sum_{k=0}^{D} \mu_{n-k} C_k(\mu_1, \ldots, \mu_k) = 0$$

which can be solved for $\mu_n$, since $C_0 = 1$, to get the claimed result. $\square$

This recursion relation reduces all the moments of an algebra-valued matrix onto the first few ones. Decidability of the problem would be in correspondence to the existence of a label, i.e. computable quantity $\mathcal{L}(\mu_1, \ldots, \mu_D)$, that quantifies whether the moments are all positive (with respect to the convex cone) or not. But before we dive deeper into the question of decidability, we shall prove some more statements. We will begin with the following corollary wich yields an alternative form of the moments recursion formula.

**Corollary 46.** *Let $A \in \mathrm{Mat}_D(R)$ and $(\mu_i)_{i \in \mathbb{N}}$ its moments. The Bell polynomials evaluated at the moments*

$$B_n(-0!\mu_1, \ldots, -(n-1)!\mu_n) \tag{2.7}$$

*vanish for $n > D$.*

*Proof.* We will prove the statement by induction over $k$ and exploit a well known recursion formula for the Bell polynomials [20]:

$$B_{n+1}(x_1, \ldots, x_{n+1}) = \sum_{k=0}^{n} \binom{n}{k} B_k(x_1, \ldots, x_k) x_{n-k+1}.$$

Concerning the case $k = D + 1$, this recursion formula yields

$$\begin{aligned} B_{D+1}(-0!\mu_1, \dots, -D!\mu_{D+1}) \\ &= \sum_{k=0}^{D} \binom{D}{k} B_k(-0!\mu_1, \dots, -(k-1)!\mu_k) \left[-(D-k)!\mu_{D+1-k}\right] \\ &= -D! \left[ \mu_{D+1} + \sum_{k=1}^{D} \frac{1}{k!} B_k(-0!\mu_1, \dots, -(k-1)!\mu_k)\mu_{D+1-k} \right], \end{aligned}$$

where we split the first summand off the sum. The term in the square brackets vanishes due to the moments recursion formula (2.6), which finishes the first step of induction.

Now, assume $B_k(-0!\mu_1, \dots, -(n-1)!\mu_n)$ to vanish for $k = D+1, \dots, n$ for an $n > D$. Doing the very same modifications to the Bell recursion formula as in the first step yields

$$\begin{aligned} B_{n+1}(-0!\mu_1, \dots, -n!\mu_{n+1}) \\ &= -n! \left[ \mu_{n+1} + \sum_{k=1}^{n} \frac{1}{k!} \mu_{n+1-k} B_k(-0!\mu_1, \dots, -(k-1)!\mu_k) \right]. \end{aligned}$$

By using the induction assumption together with the moments recursion formula, we again find that the square brackets vanish. □

That all the higher Bell polynomials vanish for the moments of a matrix allows us to put the recursion relation in another, more compact form. To see this, we note that, concerning the $k$-th Bell polynomial, the only non-vanishing monomial containing $x_k$ is the monomoial $x_k$ itself. Moreover, its coefficient is the unit and, thus, we can write

$$B_k(x_1, \dots, x_k) = B_k(x_1, \dots, x_{k-1}, 0) + x_k.$$

That in turn yields

$$\begin{aligned} \mu_k &= \frac{1}{(k-1)!} B_k(-0!\mu_1, -1!\mu_2, \dots, -(k-2)!\mu_{k-1}, 0) \\ &= -kC_k(\mu_1, \dots, \mu_{k-1}, 0) \end{aligned}$$

after inserting $x_i \mapsto -(i-1)!\mu_i$ and applying corollary (46).

As a completion of this section, we want to draw a connection to the Newton identities. Those are algebraic equations - see for example [32] - that establish a connection between power sum polynomials,

$$p_k := x_1^k + x_2^k + \cdots + x_n^k,$$

and elementary symmetric polynomials

$$e_k := \sum_{1 \le i_1 < \cdots < i_k \le n} x_{i_1} \ldots x_{i_k}.$$

Most importantly, those equations allow us to specify the power sums for $k > n$ just by the first $n$ power sums. Obviously, this is a relation similar to what we just showed for the moments of a matrix. This is beacause, as stated in the begining of this section, we can write the moments of a matrix $A \in \mathrm{Mat}_n(R)$ as power sums if the characteristic polynomial splits in linear factors,

$$\mathfrak{ch}_A = \prod_{i=1}^{n} (t - \lambda_i).$$

In that case, the moments read $\mu_k = p_k(\lambda_1, \ldots, \lambda_n)$ and, by the Newton identities, the elementary symmetric polynomials are given by the Bell polynomials,

$$\frac{(-1)^k}{k!} B_k(-0!\mu_1, -1!\mu_2, \ldots, -(k-1)!\mu_k) = e_k(\lambda_1, \ldots, \lambda_n).$$

Last but not least, the Newton identities then yield the recursion formula (2.6). However, as soon as the characteristic polynomial does not split, the moments are not related to power sums anymore. This is why we established the identities above that apply to non-algebraically closed rings.

### 2.3.2 Implications for the moments-membership problem

To begin, we summarize the results of the former section. If $\mathcal{A}$ is a commutative, associative and unital algebra, we used the characteristic polynomial of the matrix $A \in \mathrm{Mat}_D(\mathcal{A})$ to establish a recursion formula

$$\mu_n = \sum_{k=1}^{D} \mu_{n-k} C_k(\mu_1, \ldots, \mu_k)$$

for the moments $\mu_n$, $n > D$, of the matrix $A$. The first implication of this result is a significant redundancy in the input set. Instead of handing over the whole matrix $A \in \mathrm{Mat}_D(\mathcal{A})$ to an algorithm, it is indeed sufficient to just submit the first $D$ moments as input data since they already represent the full set of moments.

**Problem 7** (Moments-membership - revisited)**.** *Let $\mathcal{A}$ be a commutative, associative, unital algebra and $\mathcal{C} \subset \mathcal{A}$ a convex cone. Given $\mu_1, \ldots, \mu_D \in \mathcal{A}$, decide whether all moments specified by the recursion (2.6) are elements of $\mathcal{C}$.*

To see that this formulation of the problem for special commutative algebras is indeed equivalent to the moments-membership problem (4), we reconstruct a representative matrix from the first few moments. Specifically, let

$\mu = (\mu_1, \dots, \mu_D) \in \mathcal{A}^D$, then the matrix

$$T(\mu) := \begin{pmatrix} 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \\ -C_D(\mu) & -C_{d-1}(\mu) & \cdots & -C_1(\mu) \end{pmatrix} \tag{2.8}$$

can be shown to have the first moments $\mu_1, \dots, \mu_D$. This is done by calculating its characteristic polynomial. This yields

$$\mathfrak{ch}_T = t^d + \sum_{k \in [d]} C_k(\mu) t^{d-k}$$

which is the characteristic polynomial of a matrix with the first moments $\mu_1, \dots, \mu_D$. Note that the definition of the matrix $T(\mu)$ yields a computable mapping $T : \mathcal{A}^d \to \mathrm{Mat}_d(\mathcal{A})$. Together with the map

$$\begin{aligned} \mu : \mathrm{Mat}_d(\mathcal{A}) &\to \mathcal{A}^d \\ A &\mapsto (\mathrm{tr}\, A^k)_{k \in [d]}, \end{aligned}$$

this establishes an equivalence of decision problems. This equivalence builds on the fact that multiple matrices can have the same set of moments. Thus, we have a computable equivalence relation $\sim_\mu$ on the set of matrices. The relation by $\mu$ reduces the input set to the minimum concerning this redundancy, as it maps a whole equivalence class of matrices onto a single instance. The map $T$ on the other hand can be seen as a section of the quotient space $\mathrm{Mat}_d(\mathcal{A})/\sim_\mu$, i.e. a map assigning a representative $A \in \mathrm{Mat}_d(\mathcal{A})$ to each equivalence class in $[A] \in \mathrm{Mat}_d(\mathcal{A})/\sim_\mu$.

We have established that it is sufficient to take the generating moments as the input set of the decision problem. In the next step, we want to put the recursion formula in another form that reflects the determination of any higher moment by the first $D$ ones in a more explicit manner. More precisely, we note that the matrix $T(\mu)$ just defined can be used to raise the moments order in the following way:

$$\begin{pmatrix} \mu_{n-d+1} \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \\ -C_d(\mu) & -C_{d-1}(\mu) & \cdots & -C_1(\mu) \end{pmatrix} \begin{pmatrix} \mu_{n-d} \\ \vdots \\ \mu_{n-1} \end{pmatrix}. \tag{2.9}$$

Iterative application of this equation yields yet another form of the recursion relation:

**Corollary 47.** *Let $d \in \mathbb{N}$ and $\mu_1, \ldots, \mu_d \in \mathcal{A}$ the generating moments. Then the moments satisfy*

$$\begin{pmatrix} \mu_{n+1} \\ \vdots \\ \mu_{n+d} \end{pmatrix} = T(\mu)^n \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_d \end{pmatrix}$$

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

This form has a central advantage when we want to calculate the higher moments algorithmically. Note that the previous identities let us compute the moment $\mu_{n+1}$ in its dependence of $\mu_n, \ldots, \mu_{n-D}$. This time however, we directly obtain $\mu_{n+1}$ in its dependence solely of the generating moments which spares us an extensive substitution procedure. We may state this formula in the more compact manner

$$\mu_{n+1} = \langle 1|T(\mu)^n|\mu\rangle = \langle d|T(\mu)^{n-d+1}|\mu\rangle \tag{2.10}$$

with $|\mu\rangle \equiv (\mu_1, \ldots, \mu_D)^T$ refering to the column of generating moments and $\langle 1| \equiv (1, 0, \ldots, 0)$ as usual.

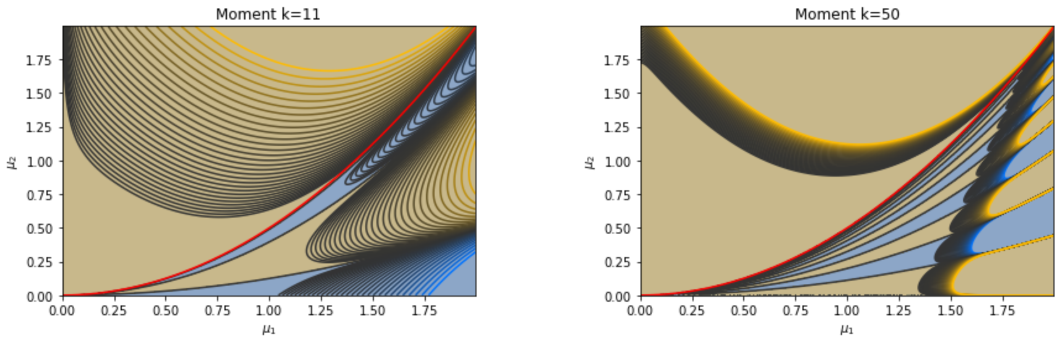
### 2.3.3 The case $\mathcal{A} = \mathbb{A}$

We shall start this discussion with the easiest non-trivial case, namely the field $\mathbb{A}$ itself as a one-dimensional algebra. The algebra product thereby coincides with the scalar multiplication. We will begin this discussion by showing the decidability of the moments-membership problem for this simple case for $D = 2$. However, the solving algorithm will be based on an argumentation concerning the eigenvalues of the input matrix. Due to our interest in more general commutative algebras over $\mathbb{A}$, we shall finish this section with a discussion of some ansatzes building around the moments instead of the eigenvalues.

In the scenario described above, our task is to decide whether all moments of a given matrix $A \in \mathrm{Mat}_D(\mathbb{A})$ are non-negative. Considering the case $D = 2$, we have two generating moments. Thus, we want to investigate the limit $n \to \infty$ of the sequence of semi-algebraic sets $S_n$ described by the families of polynomial inequalities

$$P_n = \{\mu_k(\mu_1, \mu_2) \geq 0\}_{2<k\leq n}$$

for $n > 2$ respectively. More precisely, we will show that this limit is a semi-algebraic set, i.e. generated by a finite amount of inequalities. Such a set is computable; the algorithm runs through all the inequalities and halts with output "0" if any is violated and "1" if all are fulfilled. Since we demand for all moments to be non-negative, we can assume that $\mu_1, \mu_2 \geq 0$. In figure (2.1), two moments are illustrated in their dependence of the generating moments. For the calculation of the presented data, the recursion formulas (2.6) as well as (2.10) were implemented in a phyton script. We observe two different domains with a seemingly well-behaving transition. Its limit is indicated by the red line. In the domain underneath this line, we note a strongly oscillating behaviour between positive and negative values. On the other hand, the domain above the red line is well-behaving and positive. This is precisely what we will find in the ongoing discussion of the problem. The red line turns out to be the boundary, where the roots of the characteristic polynomial get imaginary.




(A) The 11th moments dependence on $(\mu_1, \mu_2)$

(B) The 50th moments dependence on $(\mu_1, \mu_2)$

FIGURE 2.1: Illustration of the dependence of two higher moments of a $2 \times 2$-matrix on the generating moments $\mu_1$ and $\mu_2$. The lines are level lines where the color indicates positive (orange), negative (blue) and near zero domains (dark grey). The red line shows the border line between yes- and no-instances of the decision problem.

In the following, we will show that the limit $\bigcap_{n\to\infty} S_n$ is semi-algebraic and determined by two polynomial inequalities. To do so, we will use analytic tools rather than algebraic ones. To begin with, we note that if the first two moments are real, then so are the coefficients of the characteristic polynomial. This in turn implies that its roots are closed under complex conjugation, i.e. if $\lambda$ is a root with non-vanishing imaginary part, then $\bar{\lambda}$ too is a root of the polynomial. In the present case, we thus deal with either two real or a pair of complex conjugate eigenvalues. In the first case, it is immediate to see that the non-negativity of the first moment implies non-negativity of all others due to the monotony of the power map $\bullet^k$ for $k > 0$. Precisely, if $a \geq b$ is fulfilled then so is $a^k \geq b^k$ for $k \in \mathbb{N}$. On the other hand, if we have a complex conjugate pair of roots, the moments take the form

$$\mu_k = \lambda^k + \bar{\lambda}^k = 2|\lambda|^k \cos(k\theta),$$

where $\lambda = |\lambda| \exp(i\theta)$ in polar form. We are interested in the values of $|\lambda|$ and $\theta$ that yield non-negative moments for all $k \in \mathbb{N}$. Clearly, the sign of the moment is determined by the cosine term which is non-negative for all $k$ if and only if

$$\theta = 0 \mod 2\pi.$$

Thus, we conclude that a non-vanishing imaginary part always implies the existence of negative moments. Moreover, for each $k \in \mathbb{N}$ we can find a matrix with non-negative moments up to $\mu_k$ and $\mu_{k+1} < 0$ by choosing the phase $\theta$ appropriately.

We have found that the set of non-negative moments can be specified by $\mu_1 \geq 0$ and real roots, i.e. $\mathrm{Im}(\lambda_1) = \mathrm{Im}(\lambda_2) = 0$. This is already enought to conclude the decidability of the moments problem for $\mathcal{A} = \mathbb{A}$ and $D = 2$.

**Theorem 48.** *The moments membership problem (7) for $\mathcal{A} = \mathbb{A}$ and $D = 2$ is decidable.*

We can further show that the set is semi-algebraic by deriving an expression for the roots of the characteristic polynomial. We already know that

$$\begin{aligned}\mathfrak{ch} &= t^2 + C_1(\mu_1)t + C_2(\mu_1, \mu_2) \\ &= t^2 - \mu_1 t + \frac{1}{2}(\mu_1^2 - \mu_2).\end{aligned}$$

The characteristic polynomial yields the eigenvalues in their dependence on the moments:

$$\lambda_\pm = \frac{\mu_1}{2} \pm \frac{1}{2}\sqrt{2\mu_2 - \mu_1^2}.$$

It is immediate that the roots are real if and only if the inequality $2\mu_2 - \mu_1^2 \geq 0$ is satisfied. Thus, we find that

$$\bigcap_{n\to\infty} S_n = \{(\mu_1, \mu_2) \in \mathbb{A}^2_{\geq 0} \mid 2\mu_2 - \mu_1^2 \geq 0\}. \tag{2.11}$$

The saturated equation $\mu_2 = \frac{1}{2}\mu_1^2$ corresponding to the latter inequality gives rise to the red line in figure (2.1). Concerning these visualizations, one might already notice the eye-catching parabola formed structures. Those can be traced back to a scaling action of $\mathbb{A}_{\geq 0}$ that does not change the sign of the moments. Precisely, given a matrix $A \in \mathrm{Mat}_d(\mathbb{A})$, a scaling by a positive number $\lambda \in \mathbb{A}_{\geq 0}$, i.e. $A \to \lambda A$, yields a polynomial scaling of the moments according to $\mu_n \to \lambda^n \mu_n$. Since this scaling does not effect the moments sign, all instances related by such a transformation yield an equal outcome in view of the moments-membership problem.

### 2.3.4 Outlook

Unfortunately, the investigation of the former section cannot be generalized to other commutative algebras due to the usage of analytic tools, i.e. the polar representation of the eigenvalues. Moreover, if we were not blessed with an algebraically closed algebra, the eigenvalues would not necessarily exist - at least not as elements of the algebra. We are thus in need of a different ansatz that allows us to analyze those cases.

Since we have established the recursion formula (2.6), it seems worth to start a closer investigation from the view of constant-recursive sequences. Those are sequences $\{x_i\}_{i\in\mathbb{N}}$ of the form

$$x_{n+k} = c_1 x_{n+k-1} + \cdots + c_k x_n$$

with recursion coefficients $c_i$ independent of $n$. As this is the case for the sequence of moments, an investigation from this point of view could yield a better understanding for the case of higher dimensional algebras.

Another interesting direction for a further investigation builds a bridge to a more exotic domain of mathematics. The hyperreal numbers - a special field extension of the real numbers - have an interesting behaviour concerning the intersection of an enumerable amount of semi-algebraic sets. Precisely, they are $\aleph_1$-saturated [29]. As a consequence, if such an intersection is semialgebraic, the intersection is already saturated for a finite amount of those. This has clear implications for the moments of a matrix over the field of hyperreal numbers. A result like equation (2.11), i.e. that the set of matrices with non-negative moments is semi-algebraic, would imply that we would only have to check the positivity of the first $k_{\max}$ moments for a certain $k_{\max} \in \mathbb{N}$ to decide for all. However, for standard complex numbers, we have seen that already in the simplest case $D = 2$ there is no such $k_{\max}$. Instead, for any $k$ we can construct a matrix with non-negative moments up to the $k$-th moment and a negative $(k+1)$-th moment. Combining those different results, we can conclude that a proof of (2.11) has to use arguments that cannot be extended to the hyperreals. Otherwise, it would contradict the $\aleph_1$-saturation. It would be interesting to investigate this issue in more detail and for other algebras.

# Chapter 3

# Conclusion and Outlook

With the moments-membership problem (4) for an algebra $\mathcal{A}$ and a convex cone $\mathcal{C}$, we have introduced a decision problem for algebra-valued tensor networks:

Given a local tensor $A \in \mathrm{Mat}_D(\mathcal{A})$, decide whether the moments $\{\mathrm{Tr}(A^k)\}_{k\in\mathbb{N}}$ are members of the cone $\mathcal{C}$.

This can be seen as a generalization to the undecidable problem for positive semidefiniteness of translationally invariant tensor networks as discussed in [8]. By construction of a reduction based on the former problem, we showed the undecidability of the moments-membership problem for non-commutative polynomials equipped with the cone of non-negative coefficients in theorem (27). As a foundation of this reduction, we took a vector space homomorphism into a subspace of the respective algebra. In a further, more general analysis of this precise ansatz, we proved the undecidability for a large class of algebras, namely the tensor power algebras $V^{\otimes}$ for vector spaces $V$ of dimension greater than 7. This was established in theorem (32) and the following corollary. Similar to the first example, this algebra was equipped with the cone of non-negative coefficents for an arbitrary basis of the generating vector space $V$. Despite all efforts to prove the undecidability of the moments-membership problem for the algebra of non-commutative polynomials equipped with the cone of sum of squares, its decidability remains unclear. We conclude that the membership-in-PSD problem (3) is probably not a suitable foundation for the construction of a reduction. This can be traced back to the precise cone structure of the sum of square polynomials which is not closed under multiplication. In a following investigation of commutative algebras, we established a recursion formula for the moments of an algebra-valued matrix in theorem (45). To do so, we derived an identity closely related to the Newton formulas.

There are various points that allow for a follow-up investigation. Firstly, since the moments-membership problem for sum of square polynomials remains open, one may try to resolve it on the foundation of another undecidable problem like the matrix-mortality problem - or one of its relatives. On the other hand, we set a basis for a broader investigation of the moments-membership problem for commutative algebras. Its connection to constant-recursive sequences may provides a good starting point. However, such a procedure should be combined with a closer look on the hyperreal number field. As motivated, their exotic properties, specifically the $\aleph_1$-saturation, might cause some interesting restrictions on the moments-membership problem. Last but not least, it remains to explore the implications of undecidable moments-membership problem towards non-existence results. Though ubiquitous, undecidability can impose strong restrictions on structures related to the decision problem at hand. As done in numerous recent works [2, 6, 8] in different branches of physics, these kind of implications certainly offer various possibilities for future projects.

# Bibliography


[1] Ian Affleck et al. "Rigorous results on valence-bond ground states in antiferromagnets". In: *Physical Review Letters* 59.799 (1987).

[2] J. Bausch et al. *Size-Driven Quantum Phase Transitions*. Tech. rep. 2018. arXiv: 1512.05687v3.

[3] Robert Berger. "The undecidability of the domino problem". In: *American Mathematical Soc.* (1966).

[4] Julien Cassaigne et al. "Tighter Undecidability Bounds for Matrix Mortality, Zero-in-the-Corner Problems, and More". In: (2014). arXiv: 1404.0644. URL: http://arxiv.org/abs/1404.0644.

[5] Gregory M Crosswhite and Dave Bacon. "Finite automata for caching in matrix product algorithms". In: *Physical Review A* (2008). DOI: 10.1103/PhysRevA.78.012356.

[6] T. S. Cubitt, D. Perez-Garcia, and M. M. Wolf. *Undecidability of the Spectral Gap*. Tech. rep. 2020. arXiv: 1502.04573v4.

[7] G. De las Cuevas, T. Fritz, and T. Netzer. "Optimal Bounds on the Positivity of a Matrix from a few Moments". In: *Communications in Mathematical Physics, Springer* (2020).

[8] G. De las Cuevas et al. "Fundamental limitations in the purifications of tensor networks". In: *Journal of Mathematical Physics* 57.7 (2016). ISSN: 00222488. DOI: 10.1063/1.4954983. arXiv: 1512.05709.

[9] Gemma De Las Cuevas, Matt Hoogsteder Riera, and Tim Netzer. *Tensor decompositions on simplicial complexes with invariance*. 2019. arXiv: 1909.01737.

[10] David Deutsch. "Quantum computational networks". In: *Proceedings of the Royal Society of London* 425.1868 (1989).

[11] G Evenbly and G Vidal. "Algorithms for entanglement renormalization". In: *Physical Review B* (2009). DOI: 10.1103/PhysRevB.79.144108.

[12] Richard P. Feynman. "Quantum mechanical computers". In: *Foundations of Physics* 16.6 (1986), pp. 507–531. ISSN: 00159018. DOI: 10.1007/BF01886518. URL: https://link.springer.com/article/10.1007/BF01886518.

[13] Wen Cong Gan, Fu Wen Shu, and Meng He Wu. "Thermal geometry from CFT at finite temperature". In: *Physics Letters, Section B: Nuclear, Elementary Particle and High-Energy Physics* 760 (2016), pp. 796–799. ISSN: 03702693. DOI: 10.1016/j.physletb.2016.07.073. arXiv: 1605.05999.

[14] Kurt Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: *Monatshefte für Mathematik* (1931).

[15] J. William Helton. ""Positive" noncommutative polynomials are sums of squares". In: *Annals of Mathematics* 156.2 (2002), pp. 675–694. ISSN: 0003486X. DOI: 10.2307/3597203.

[16] D. Hilbert and W. Ackermann. *Grundzüge der theoretischen Logik*. 1928.

[17] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Pearson Educated Limited, 2014.

[18] D. Jaschke, S. Montangero, and L. D. Carr. *One-dimensional many-body entangled open quantum systems with tensor network methods*. 2019. DOI: 10.1088/2058-9565/aae724. arXiv: 1804.09796. URL: https://iopscience.iop.org/article/10.1088/2058-9565/aae724https://iopscience.iop.org/article/10.1088/2058-9565/aae724/meta.

[19] Dexter C. Kozen. *Automata and Computability*. Springer, 1997.

[20] M I Krivoruchenko. "Trace Identities for Skew-Symmetric Matrices". In: *Mathematics in Computer Science* 1.2 (2016), p. 28. ISSN: 1661-8270. DOI: 10.11648/J.MCS.20160102.11. URL: http://www.sciencepublishinggroup.com/j/mcs.

[21] Serge Lang. *Algebra*. Graduate Texts in Mathematics. New York, NY: Springer New York, 2002. ISBN: 978-1-4612-6551-1. DOI: 10.1007/978-1-4613-0041-0. URL: http://link.springer.com/10.1007/978-1-4613-0041-0.

[22] F. William Lawvere. "Diagonal arguments and cartesian closed categories". In: 15 (1969), pp. 134–145. DOI: 10.1007/bfb0080769.

[23] Raphael Loewy and David London. "A note on an inverse problem for nonnegative matrices". In: *Linear and Multilinear Algebra* 6.1 (1978), pp. 83–90. URL: https://doi.org/10.1080/03081087808817226.

[24] M. Miyaji et al. "Boundary states as holographic duals of trivial spacetimes". In: *Journal of High Energy Physics* 2015.5 (2015), p. 152. ISSN: 10298479. DOI: 10.1007/JHEP05(2015)152. arXiv: 1412.6226. URL: https://link.springer.com/article/10.1007/JHEP05(2015)152.

[25] Theodore S. Motzkin. "The arithmetic-geometric inequality". In: *Academic Press, New York* (1967), pp. 205–224.

[26] R. Penrose. "Applications of negative-dimensional tensors". In: *Academic Press Inc.* (1971).

[27] V. Pestun and Y. Vlassopoulos. *Tensor Network Language Model*. Tech. rep. 2017. arXiv: 1710.10248v2.

[28] B. Poonen. *Undecidable problems: A sampler*. Tech. rep. 2014. URL: http://www.cambridge.org/gb/academic/subjects/philosophy/.

[29] Alexander Prestel and Charles N. Delzell. *Positive Polynomials*. Springer Monographs in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. ISBN: 978-3-642-07445-5. DOI: 10.1007/978-3-662-04648-7. URL: http://link.springer.com/10.1007/978-3-662-04648-7.

[30] Bertrand Russell. *The Philosophy of Logical Atomism*. 1918.

[31] E. M. Stoudenmire and D. J. Schwab. *Supervised Learning With Quantum-Inspired Tensor Networks*. Tech. rep. 2017. arXiv: 1605.05775v2. URL: https://github.com/emstoudenmire/TNML.

[32] Fernando Szechtman. *Newton's identities*. 2009. DOI: 10.2307/2324242.

[33] F. Verstraete and J. I. Cirac. *Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions*. Tech. rep. 2004.

[34] G Vidal. "Entanglement Renormalization". In: *Physical Review Letters* (2007). DOI: 10.1103/PhysRevLett.99.220405.

[35] Michael M Wolf. *Lecture on undecidability*. 2012, pp. 1–49.

[36] Noson S. Yanofsky. "A Universal Approach to Self-Referential Paradoxes, Incompleteness and Fixed Points". In: *Bulletin of Symbolic Logic* 9.3 (2003), pp. 362–386. ISSN: 1079-8986. DOI: 10.2178/bsl/1058448677. arXiv: 0305282 [math].

[37] Richard Zach. *Hilbert's Program Then and Now*. Tech. rep. 2005.