# The Complexity of Derivations of Matrix Identities

by

Michael Soltys-Kulinicz

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Mathematics
University of Toronto

# Abstract

The Complexity of Derivations of Matrix Identities

Michael Soltys-Kulinicz
Doctor of Philosophy
Graduate Department of Mathematics
University of Toronto
2001

In this thesis we are concerned with building logical foundations for Linear Algebra, from the perspective of proof complexity. As the cornerstone of our logical theories, we use Berkowitz's parallel algorithm for computing the coefficients of the characteristic polynomial of a matrix.

Standard Linear Algebra textbooks use Gaussian Elimination as the main algorithm, but they invariably use the (very infeasible) Lagrange expansion to prove properties of this algorithm.

The main contribution of this thesis is a (first) feasible proof of the Cayley-Hamilton Theorem, and related principles of Linear Algebra (namely, the axiomatic definition of the determinant, the cofactor expansion formula, and multiplicativity of the determinant). Furthermore, we show that these principles are equivalent, and the equivalence can be proven feasibly.

We also show that a large class of matrix identities, such as:

$$AB = I \rightarrow BA = I$$

proposed by S.A. Cook as a candidate for separating Frege and Extended Frege propositional proof systems, all have feasible proofs, and hence polynomially-bounded Extended Frege proofs. We introduce the notion of completeness for these matrix identities.

As the main tool to prove our results, we design three logical theories:

$$\mathrm{LA} \subset \mathrm{LAP} \subset \forall\mathrm{LAP}$$

LA is a three-sorted quantifier-free theory of Linear Algebra. The three sorts are indices, field elements and matrices. This is a simple theory that allows us to formalize and prove all the basic properties of matrices (roughly the properties that state that the set of matrices is a ring). The theorems of LA have polynomially-bounded Frege proofs.

We extend LA to LAP by adding a new function, P, which is intended to denote matrix powering, i.e., $\mathtt{P}(n, A)$ means $A^n$. LAP is well suited for formalizing Berkowitz's algorithm, and it is strong enough to prove the equivalence of some fundamental principles of Linear Algebra. The theorems of LAP translate into quasi-polynomially-bounded Frege proofs.

We finally extend LAP to ∀LAP by allowing induction on formulas with ∀ matrix quantifiers. This new theory is strong enough to prove the Cayley-Hamilton Theorem, and hence (by our equivalence) all the major principles of Linear Algebra. The theorems of ∀LAP translate into polynomially-bounded Extended Frege proofs.

# Acknowledgements

First and foremost, I would like to thank my supervisor, Stephen Cook. It is thanks to his dedication, his patience, and our weekly meetings that this thesis has been written. It is not possible to have a better advisor.

I want to thank the other members of my PhD committee: Toniann Pitassi, Charles Rackoff, and especially Alasdair Urquhart, for their help and encouragement. Samuel Buss, the external examiner, gave me a lot of very good comments and suggestions.

I am grateful to Zbigniew Stachniak who first told me about Complexity Theory and Logic.

I was lucky to be part of a fantastic group of graduate students: Joshua Buresh-Oppenheim, Valentine Kabanets, Antonina Kolokolova, Tsuyoshi Morioka, Steve Myers, François Pitt, Eric Ruppert, Alan Skelley, and others. I found our theory students seminars extremely helpful.

Finally, I want to thank my parents for their support over the years.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Proof Theory is the area of mathematics which studies the concepts of mathematical proof and mathematical provability ([Bus98]). Proof Complexity is an area of mathematics and theoretical computer science that studies the length of proofs in propositional logic. It is an area of study that is fundamentally connected both to major open questions of computational complexity theory and practical properties of automated theorem provers ([BP98]).

A propositional formula $\phi$ is a *tautology* if $\phi$ is true under all truth value assignments. For example, $\phi$ given by:

$$p \vee \neg p$$

is a tautology. Let TAUT be the set of all tautologies. A *propositional proof system* is a polytime predicate $P \subseteq \Sigma^* \times \text{TAUT}$ such that:

$$\phi \in \text{TAUT} \iff \exists x P(x, \phi)$$

$P$ is *poly-bounded* (i.e., *polynomially bounded*) if there exists a polynomial $p$ such that:

$$\phi \in \text{TAUT} \iff \exists x(|x| \leq p(|\phi|) \wedge P(x, \phi))$$

The existence of a poly-bounded proof system is related to the fundamental question:

$$\text{P} \stackrel{?}{=} \text{NP}$$

In 1979 Cook and Reckhow ([CR79]) proved that NP = co-NP iff there is a poly-bounded proof system for tautologies. On the other hand, if P = NP then NP = co-NP. Thus, if there is no poly-bounded proof system, then NP $\neq$ co-NP, and that in turn would imply that P $\neq$ NP.

There is a one million $ cash prize offered by the Clay Mathematical Institute for settling the P $\stackrel{?}{=}$ NP problem; see the Millennium Prize Problems on the web site of the CMI at `www.claymath.org/index.htm`. Also see [Coo00b], a manuscript prepared by Cook for the CLI for the Millennium Prize Problems, available at `www.cs.toronto.edu/~sacook`.

Thus, considerable effort goes into proving lower bounds (and separations) for propositional proof systems. The program is to show lower bounds for standard proof systems of increasing complexity.

But the P $\stackrel{?}{=}$ NP problem is not the only motivation for finding lower bounds for Propositional Proof Systems (PPS):

- PPS are (mathematically) interesting in their own right.
- Applications to Automated Reasoning (Artificial Intelligence).
- We can use lower bounds for PPS, to prove lower bounds for decision procedures (for SAT). A good example of this is the exponential lower bound for resolution, which gives us an exponential lower bound for the Davis-Putnam procedure for satisfiability. The idea behind the correspondence is very simple: each instance of the Davis-Putnam procedure on a particular set of clauses can be viewed ("upside down") as a resolution refutation. Thus, if all resolution refutations on a family of clauses must be of a certain size, so must be all instances of the Davis-Putnam procedure on that family of clauses. (See [BP96] for the resolution lower bound).

See Figure 1.1 for a table of the principal propositional proof systems. Exponential lower bounds exist for the proof systems below the line. The strongest propositional proof system (Quantified Frege) is shown in the top, and the weakest (Truth Tables) is shown in the bottom. Each system can simulate the one below. The systems Frege and PK are equivalent in the sense that they $p$-simulate each other (see below for $p$-simulation).

In this thesis we are concerned with all four types of Frege proof systems. There is a separation between Bounded Depth Frege and Frege, and there exist lower bounds for Bounded Depth Frege, but no such results exist for the remaining Frege systems. By a *separation* we mean that there exists a family of tautologies $\tau_n$, such that Frege proves $\tau_n$ efficiently (i.e., in polysize), but Bounded Depth Frege does not (i.e., there is no polynomial $p(n)$ such that Bounded Depth Frege can prove $\tau_n$ with derivations of length at most $p(n)$). The Pigeonhole Principle (PHP) is the standard tautology for separating Bounded Depth Frege and Frege (see [Pit92], [BIP93] and [BIK$^+$92]).

Quantified Frege

Extended Frege, Substitution Frege, Renaming Frege

Permutation Frege

Frege, PK

---

Bounded Depth (BD) Frege

Resolution

Truth Tables

Table 1.1: Propositional proof systems

Note that even though we mention Frege, in practice in this thesis we use the sequent calculus proof system PK. Thus we have Bounded Depth PK, PK, Extended PK, and Quantified PK. It is easy to show that Frege and PK $p$-simulate each other, and hence they can be used interchangeably.

The (alleged) separation between Frege and Extended Frege is a *fundamental open problem.* The matrix identity $AB = I \rightarrow BA = I$ was originally proposed by Cook in the context of separating Frege and Extended Frege (private communication; in [BBP94] the authors give examples of tautology families, such as the "Odd Town Theorem", that seem to depend on linear algebra for their proofs, and it was this paper that inspired Cook to think of $AB = I \rightarrow BA = I$). The separation between Extended Frege and Quantified Frege (again, if there is one), seems to be completely out of reach at the moment.

A fundamental notion that appears throughout this thesis is that of a *feasible proof* (and *feasible computation*, or *polytime computation*). Feasible proofs were introduced by Cook in [Coo75], and they formalize the idea of tractable reasoning; a theorem can be proven feasibly, if all the computations involved in the proof are polytime computations, and the induction can be unwound feasibly.

Cook's system PV is the original system for polytime reasoning (see [CU93]). Samuel R. Buss formalized polytime reasoning with the system $S^1_2$ in [Bus86]. The importance of the Extended Frege propositional proof system stems from the fact that first order theorems which have feasible proofs correspond to propositional tautologies which have uniform polysize Extended Frege proofs.

Another fundamental notion throughout this thesis is that of a *$p$-simulation.* We say that a proof system $P$ $p$-simulates a proof system $P'$ if there exists a function $f$ and a polynomial $p$ such that every proof $x$ in $P'$ corresponds to a proof $f(x)$ in $P$, and

$|f(x)| \leq p(|x|)$. In other words, all the proofs of $P'$ can be "reproduced" in $P$ with a small increase in size. Thus, coming back to the separations discussed above, for example, Frege $p$-simulates Bounded Depth Frege, but Bounded Depth Frege does *not* $p$-simulate Frege. It is not known if Frege can $p$-simulate Extended Frege.

## 1.1 Motivation

The motivation for the research presented in this thesis is establishing the complexity of the concepts involved in proving standard theorems in Linear Algebra. We want to understand where do standard theorems of Linear Algebra stand with respect to the Frege proof systems (Bounded Depth Frege, Frege, Extended Frege, and Quantified Frege). In particular, we are interested in the complexity of the proofs of the following principles:

- Standard theorems of Linear Algebra, such as the Cayley-Hamilton Theorem, the axiomatic definition of the determinant, the cofactor expansion formula, and the multiplicativity of the determinant.

- Universal matrix identities such as $AB = I \rightarrow BA = I$.

Thus, we are concerned with building logical foundations for Matrix Algebra, from the perspective of the complexity of the computations involved in the proofs. We use Berkowitz's parallel algorithm as the main tool for computations, and most results are related to proving properties of this algorithm. Berkowitz's algorithm computes the coefficients of the characteristic polynomial of a matrix, by computing iterated matrix products.

Standard Linear Algebra textbooks use Gaussian Elimination as the main algorithm, but they invariably use the (very infeasible) Lagrange expansion to prove properties of the determinant. Berkowitz's algorithm is a fast parallel algorithm, Gaussian Elimination is poly-time, and the Lagrange expansion is $n!$ (where the parameter for all three is the size of the matrix).

We have chosen Berkowitz's algorithm as the cornerstone of our theory of Linear Algebra because it is the fastest known algorithm for computing inverses of matrices, and it has the property of being field independent (and hence all the results of this thesis are field independent). Furthermore, we show that we can feasibly prove properties of the determinant using Berkowitz's algorithm, while we do not know how to prove them

feasibly using Gaussian Elimination, or any other algorithm (granted that for this thesis, we did concentrate our research on Berkowitz's algorithm).

In order to carry out our proofs based on Berkowitz's algorithm, we developed a *new* approach to proving matrix identities by induction on the size of matrices.

## 1.2 Contributions

In **Section 8.2** we present the main contribution of this thesis: a feasible proof of the Cayley-Hamilton Theorem. It seems that we give the first such proof[1]; in fact we present three feasible proofs. The first is based on interpreting the $\forall$LAP proof of the C-H Theorem in the polytime theory in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$, **Section 8.2.3**. This proof relies on results spread throughout the thesis, so we summarize it in **Section 8.2.4**. The second proof is based on interpreting the $\forall$LAP proof of the C-H Theorem in poly-bounded uniform Permutation Frege (a propositional proof system), **Section 8.2.5**. The $\forall$LAP proof itself is given in **Section 8.2.1**. The third proof is based on Quantified Frege, **Section 8.2.6**.

Note that many of the proofs given in this thesis are substantially more difficult than the corresponding proofs in an average Linear Algebra text book. An extreme example of this is the proof of multiplicativity of the determinant. In [DF91, page 364] the proof of the multiplicativity of the determinant takes one line; this proof relies on the Lagrange Expansion of the determinant. Our proof of multiplicativity of the determinant from the Cayley-Hamilton Theorem takes over six pages (see **Section 6.4**). The proof of the Cayley-Hamilton Theorem takes several sections spread throughout the thesis. However, our proofs are *feasible*; we can prove the propositional tautologies asserting the multiplicativity of the determinant, with Extended Frege, for matrices which have $10^6 \times 10^6$ entries. With the Lagrange Expansion which has $n!$ terms ($n$ is the size of the matrices involved), it is impossible to prove multiplicativity for matrices of size $20 \times 20$ (using Extended Frege).

In **Chapter 6** we show that the C-H Theorem is equivalent to the axiomatic definition of the determinant, and to the cofactor expansion, and that these equivalences can be shown in the theory LAP. The theory LAP formalizes reasoning in POW (the class of problems "easily" reducible to powers of matrices). In **Section 6.4** we show that the multiplicativity of determinant implies (also in LAP) the C-H Theorem, and we show

[1] In **Section 8.1** we present, briefly, two typical infeasible proofs of the C-H Theorem.

that the C-H implies (feasibly, but we do not know if in LAP) the multiplicativity of the determinant. The conclusion is that all these major principles of Linear Algebra have feasible proofs.

In **Section 5.3** we show that $AB = I \rightarrow BA = I$, and hence (by the results in **Section 3.2**) many matrix identities, follows, in LAP, from the Cayley-Hamilton Theorem. Since we give a feasible proof of the C-H Theorem, it follows that these identities also have feasible proofs.

We compute the determinant of a matrix with Berkowitz's algorithm. Since the Cayley-Hamilton Theorem states that the characteristic polynomial of a matrix is an annihilating polynomial (i.e. $p_A(A) = 0$), the Cayley-Hamilton Theorem implies the following:

$$\det(A) \neq 0 \implies A \text{ is invertible}$$

On the other hand, we also give a feasible proof (based on Gaussian Elimination, but still for the determinant as defined by Berkowitz's algorithm) that:

$$\det(A) = 0 \implies A \text{ is } \textit{not} \text{ invertible}$$

Therefore, we give a feasible proof of the fact that a matrix is invertible iff its determinant is not zero.

We define the correctness of Berkowitz's algorithm to be the following property: it computes an annihilating polynomial of the given matrix. Thus, we can look at the central result of this thesis as being a feasible proof of the correctness of Berkowitz's algorithm; the feasible proof of Berkowitz's algorithm is the mechanism that makes a feasible proof of the Cayley-Hamilton Theorem possible.

In **Chapter 2** we design a three-sorted quantifier-free theory of Linear Algebra, and we call it LA. The three sorts are indices, field elements and matrices. LA is field independent, and matrix identities can be expressed very naturally in its language.

LA is a fairly weak theory, which nevertheless allows us to prove all the basic properties of matrices (roughly the properties that state that the set of matrices is a ring). We show this in **Chapter 3**, where we prove, in LA, properties such as the associativity of matrix multiplication, $A(BC) = (AB)C$, or the commutativity of matrix addition, $A + B = B + A$, i.e., the ring properties of the set of matrices.

In **Chapter 7** we show that all the theorems of LA can be translated into poly-bounded families of propositional tautologies, with poly-bounded Frege proofs. Thus, LA is strong enough to prove basic properties of matrices, but at the same time the

truth of any theorem of LA can be verified with poly-bounded Frege. In fact, we prove a tighter result since we show that bounded-depth Frege proofs with MOD $p$ gates suffice, when the underlying field is $\mathbb{Z}_p$.

We identify two classes of matrix identities: *basic* and *hard*. The basic matrix identities are those which can be proven in LA, and, as was mentioned above, they roughly correspond to the ring properties of the set of matrices. Hard matrix identities, introduced in **Section 3.2**, are those which seem to require computing matrix inverses in their derivations; the prototypical example of a hard matrix identity is $AB = I \rightarrow BA = I$, suggested by Cook in the context of separating Frege and Extended Frege. Hard matrix identities are more difficult to define, and their definition is related to the definition of completeness of matrix identities. Roughly, we can say that hard matrix identities are those which can be proven from $AB = I \rightarrow BA = I$ using basic reasoning, i.e., LA.

One of the nicer results of this thesis is identifying equivalent matrix identities, where the equivalence can be proven in LA, hence with basic matrix properties, while the identities themselves are believed to be independent of LA. We refer to:

$$AB = I, AC = I \rightarrow B = C \qquad \text{I}$$

$$AB = I \rightarrow AC \neq 0, C = 0 \qquad \text{II}$$

$$AB = I \rightarrow BA = I \qquad \text{III}$$

$$AB = I \rightarrow A^t B^t = I \qquad \text{IV}$$

presented in **Section 3.2**. This suggests a notion of completeness for matrix identities, which we *try* to make precise. We discuss the notion of completeness for matrix identities in **Section 9.2**, but we do not yet have a satisfactory definition.

In **Chapter 4** we design an extension of LA, called LAP. This new theory is just LA with a new function symbol: P. The intended meaning of $\mathtt{P}(n, A)$ is $A^n$. The addition of P increases considerably the expressive power of LA (however, we have no separation result between LA and LAP—for all we know LAP might be conservative over LA, but we conjecture otherwise). Having added matrix powering, we can now compute products of sequences of matrices, so LAP is ideally suited for formalizing Berkowitz's algorithm; we express the characteristic polynomial, computed by Berkowitz's algorithm, as a term of LAP in **Section 4.2.3**.

Berkowitz's algorithm is a fast parallel algorithm for computing the characteristic polynomial of a matrix. It has the great advantage of being *field independent*, and

therefore all our results (e.g. the Cayley-Hamilton Theorem) hold irrespectively of the underlying field; fields are never an issue in our proofs. We discuss Berkowitz's algorithm in depth in **Section 4.2**.

In **Section 7.4** we show that all the theorems of LAP translate into quasi-poly-bounded Frege proofs.

In **Chapter 6** we use Berkowitz's algorithm to show that LAP proves the equivalence of several important principles of Linear Algebra:

- the Cayley-Hamilton Theorem
- the axiomatic definition of the determinant
- the cofactor expansion formula

Furthermore, we show that LAP proves that all these principles follow from the multiplicativity of the determinant. Thus, by giving a feasible proof of the Cayley-Hamilton Theorem, we are able to give feasible proofs of the axiomatic definition of determinant and the cofactor expansion.

To prove the Cayley-Hamilton Theorem we needed induction over formulas with universal quantifiers for variables of type matrix; thus we designed $\forall$LAP in **Section 8.2.1**. It seems that LAP by itself cannot prove the C-H Theorem, although we have no good evidence for this conjecture. However, we show in **Section 5.2** that LAP is capable of proving the C-H Theorem, and all the other major principles, for triangular matrices.

In **Section 6.4** we show that the Cayley-Hamilton Theorem, together with the identity $\det(A) = 0 \rightarrow AB \neq I$, imply (in LAP) the multiplicativity of the determinant. Since in **Section 8.3.2** we present a feasible proof of $\det(A) = 0 \rightarrow AB \neq I$ (based on Gaussian Elimination), it follows that there is a feasible proof of the multiplicativity of determinant from the C-H Theorem. Therefore, the multiplicativity of determinant also has a feasible proof.

To show that LAP proves the equivalences mentioned above, we developed a new approach to proving identities that involve the determinant and the adjoint. Since LAP is a theory that relies mainly on powers of matrices and on induction on terms of type index, we need a new method for proving properties of the determinant and the adjoint. The main idea in this new method is to consider the following submatrices:

$$A = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

where $a_{11}$ is the $(1,1)$ entry of $A$, and $R, S$ are $1 \times (n-1)$, $(n-1) \times 1$ submatrices, respectively, and $M$ is the *principal submatrix* of $A$. So consider for example the property of multiplicativity of the determinant, $\det(AB) = \det(A)\det(B)$. To prove this property, we assume inductively that it holds for the principal submatrices of $A$ and $B$, and show that it holds for $A$ and $B$. To accomplish this, we have developed many (perhaps new) matrix identities, such as for example:

$$\det(SR + M) = \det(M) + R\text{adj}(M)S$$

which is identity (6.22) in **Chapter 6**. Another interesting identity is identity (6.23). Both identities have feasible proofs (in LAP from the Cayley-Hamilton Theorem) given at the end of **Section 6.4**.

To illustrate out method, suppose that we want to prove that $\det(A) = \det(A^t)$. We show that:

$$\det(M) = \det(M^t) \rightarrow \det(A) = \det(A^t)$$

(this is the induction step), and we show that since $\det((a)) = a$, the claim also holds in the basis case. Using induction on the size of matrices we conclude that the claim holds for all matrices. Basically, we use induction to prove a given claim for bigger and bigger submatrices, as the picture in Figure 1.1 shows. We can define and parameterize these submatrices using our constructed terms (i.e., $\lambda ij\langle m, n, t\rangle$).

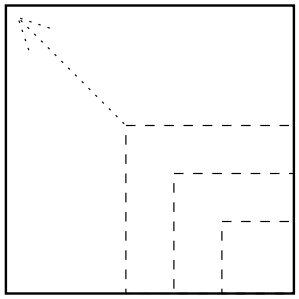


Figure 1.1: Proving claims by induction on submatrices

In **Section 8.3.1** we present a feasible proof of correctness of Gaussian Elimination. This is an interesting result because it was very difficult to give a proof of correctness of Berkowitz's algorithm, so potentially, the correctness of Gaussian Elimination might have been very problematic as well. Furthermore, we give a proof of correctness of Gaussian Elimination using poly-time concepts, that is, concepts in the same complexity class as

the Gaussian Elimination algorithm. We did *not* manage to give a proof of correctness of Berkowitz's algorithm in its own complexity class; while Berkowitz's algorithm is an $NC^2$ algorithm, its proof of correctness uses poly-time concepts.

In **Section 8.2.1** we extend LAP to $\forall$LAP by allowing $\Pi_1^M$ Induction in our proofs (that is, induction on formulas with $\forall$ matrix quantifiers, with bounds on the size of the matrix). This new theory is strong enough to prove the Cayley-Hamilton Theorem, and hence all the major principles of Linear Algebra.

Finally, we list open problems in **Chapter 9**. We discuss each of the seven open problems presented in some detail.

## 1.3 Summary of results

In this section we summarize the main results of this thesis in table format. In Table 1.2 we give brief descriptions of our logical theories LA, LAP, and ∀LAP, summarizing what are the important properties that can be proved in them. In Table 1.3 we show the propositional proof systems (and the related complexity classes), that correspond to the theories LA, LAP, and ∀LAP. In Table 1.4 we conjecture what we expect to be true.

| **Theory** | **Summary of properties provable in the theory** |
|---|---|
| LA | Ring properties of matrices (with the usual matrix addition and multiplication); for example, associativity of matrix products: $A(BC) = (AB)C$, or commutativity of matrix addition $A + B = B + A$.<br>It can also prove equivalences of hard matrix identities. |
| LAP | It extends LA by adding a new function symbol, P, for computing powers of matrices.<br>Berkowitz's algorithm can be defined in this theory (as a term in the language of LAP), and it is strong enough to prove equivalences of the Cayley-Hamilton Theorem, the axiomatic definition of the determinant, and the cofactor expansion formula.<br>It can also prove that the multiplicativity of the determinant implies the Cayley-Hamilton Theorem. |
| ∀LAP | It extends LAP by allowing universal quantifiers over variables of type matrix; in particular, it allows induction over formulas of this type.<br>It is strong enough to prove the Cayley-Hamilton Theorem and related principles, while it is still feasible. |

Table 1.2: Summary of theories

| **Theory** | **Propositional Proof System; corresponding complexity class** |
|---|---|
| LA | fields $\mathbb{Z}_p$: polybounded Bounded Depth Frege with MOD $p$ gates; $\mathrm{AC}^0[p]$ |
| | field $\mathbb{Q}$: polybounded Frege; $\mathrm{NC}^1$ |
| LAP | quasi-polybounded Frege; DET $\subseteq \mathrm{NC}^2$ |
| $\forall$LAP | polybounded Extended Frege; P/poly |

Table 1.3: Summary of translations

| **Theory** | **Related conjecture** |
|---|---|
| LA | LA $\nvdash AB = I \rightarrow BA = I$; LA does *not* prove any of the hard matrix identities.<br>In fact, we conjecture something stronger: $AB = I \rightarrow BA = I$ does *not* have polybounded Frege proofs, but it has quasi-polybounded Frege proofs. |
| LAP | LAP $\vdash AB = I \rightarrow BA = I$, that is, LAP proves hard matrix identities; we are also going to make the following bold conjecture: LAP proves the Cayley-Hamilton Theorem. We make this conjecture because we think that it is reasonable to assume that we can prove properties of the characteristic polynomial, as computed by Berkowitz's algorithm, within the complexity class of Berkowitz's algorithm. |
| $\forall$LAP | Captures polytime reasoning |

Table 1.4: Summary of conjectures

# Chapter 2

# The Theory LA

In this chapter we define a quantifier-free theory of Linear Algebra (of Matrix Algebra), and call it LA. Our theory is strong enough to prove basic properties of matrices, but weak enough so that all the theorems of LA translate into propositional tautologies with short Frege proofs.

We want LA to be just strong enough to prove all the ring properties of the set of matrices; for example, the associativity of matrix multiplication: $A(BC) = (AB)C$, or the commutativity of matrix addition: $A + B = B + A$.

We have three sorts of object: *indices*, *field elements*, and *matrices*. We define the theory LA to be a set of sequents. We use sequents, rather than formulas, for two reasons: (i) sequents are convenient for expressing matrix identities (see, for example, the four hard matrix identities in Section 3.2, page 40), and (ii) we use the sequent calculus proof system to formalize propositional derivations.

We define LA as the set of sequents which have derivations from the axioms A1–33, given below, using: rules for propositional consequence, the induction (on indices) rule, and a rule for concluding equality of matrices. Of course, all the details will be given below.

Note that LA is a quantifier-free theory, but all the sequents are implicitly universally quantified.

## 2.1 Language

We use $i, j, k, l$ as metasymbols for indices, $a, b, c$ as metasymbols for field elements, and $A, B, C$ as metasymbols for matrices. We use $x, y, z$ as meta-metasymbols; this is useful,

for example, in axiom A2 given below where $x$ can be a variable of any sort. We use primes or subscripts when we run out of letters.

**Definition 2.1.1** The language of LA, denoted $\mathcal{L}_{\mathrm{LA}}$, has the function and predicate symbols given in Table 2.1 below. The indices are intended to range over natural numbers. We have 0 and 1 indices, we also have the usual addition and multiplication of indices, but subtraction ("$-$") is intended to be "cut-off subtraction"; that is, if $i > j$, then $j - i$ is intended to be 0. The functions `div` and `rem` are intended to be the standard quotient and reminder functions. Then we also have field elements, with 0 and 1, and addition and multiplication, and multiplicative inverses (where we define $0^{-1}$ to be 0). Finally, we have $\leq$ and $=$ for indices, and $=$ for field elements and matrices. Below we give the details more formally.

$$0_{\mathrm{index}}, 1_{\mathrm{index}}, +_{\mathrm{index}}, *_{\mathrm{index}}, -_{\mathrm{index}}, \mathtt{div}, \mathtt{rem}, \mathrm{cond}_{\mathrm{index}}$$
$$0_{\mathrm{field}}, 1_{\mathrm{field}}, +_{\mathrm{field}}, *_{\mathrm{field}}, -_{\mathrm{field}}, {}^{-1}, \mathrm{cond}_{\mathrm{field}}$$
$$\mathtt{r}, \mathtt{c}, \mathtt{e}, \Sigma$$
$$\leq_{\mathrm{index}}, =_{\mathrm{index}}, =_{\mathrm{field}}, =_{\mathrm{matrix}}$$

Table 2.1: Function and predicate symbols in $\mathcal{L}_{\mathrm{LA}}$

**Intended meaning of the symbols:**

- $0_{\mathrm{index}}, 1_{\mathrm{index}}$ and $0_{\mathrm{field}}, 1_{\mathrm{field}}$ are constants (i.e. 0-ary function symbols), of type index and field, respectively.

- $+_{\mathrm{index}}, *_{\mathrm{index}}$ are 2-ary function symbols for addition and multiplication of indices, and $+_{\mathrm{field}}, *_{\mathrm{field}}$ are 2-ary function symbols for addition and multiplication of field elements.

- $-_{\mathrm{index}}$ is a 2-ary function symbol that denotes cut-off subtraction of index elements. $-_{\mathrm{field}}$ and ${}^{-1}$ are 1-ary function symbols denoting the additive and multiplicative inverse, respectively, of field elements. Again, we intend $0^{-1}$ to be 0.

- $\mathtt{div}, \mathtt{rem}$ are the quotient and reminder 2-ary functions, respectively. That is, for any numbers $m, n$, we have:

$$m = n \cdot \mathtt{div}(m,n) + \mathtt{rem}(m,n) \qquad \text{where } 0 \leq \mathtt{rem}(m,n) < m \tag{2.1}$$

  We want $m, n \geq 0$, and when we incorporate equation (2.1) as axioms of LA, we make sure that $n \neq 0$ to avoid division by zero.

  These two functions are not really used in LA, but become very important in Chapter 4, where they are used to compute products of sequences of matrices with the powering function $\mathtt{P}$.

- $\text{cond}_{\text{index}}$ and $\text{cond}_{\text{field}}$ are 3-ary function symbols, whose first argument is a formula, and the two other arguments are indices (in $\text{cond}_{\text{index}}$) or field elements (in $\text{cond}_{\text{field}}$). The intended meaning is the following:

$$\text{cond}(\alpha, \mathsf{term}_1, \mathsf{term}_2) = \begin{cases} \mathsf{term}_1 & \text{if } \alpha \text{ is true} \\ \mathsf{term}_2 & \text{otherwise} \end{cases}$$

- $\mathtt{r}$ and $\mathtt{c}$ are 1-ary function symbols whose argument is of type matrix, and whose output is of type index. $\mathtt{r}(A)$ and $\mathtt{c}(A)$ are intended to denote the number of rows and columns of the matrix $A$, respectively.

- $\mathtt{e}$ is a 3-ary function symbol, where the first argument is of type matrix, and the other two are of type index, and $\mathtt{e}(A,i,j)$ is intended to denote $A_{ij}$, i.e. the $(i,j)$-th entry of $A$. Sometimes we will use $A_{ij}$ instead of $\mathtt{e}(A,i,j)$ to shorten formulas. It is important to realize one technical point which is going to play a role later on; a matrix is a finite array, and therefore, we are going to encounter the following problem: what if we access an entry out of bounds? That is, suppose that $A$ is a $3 \times 3$ matrix. What is $\mathtt{e}(A,4,3)$? We make the convention of defining all out of bounds entries to be zero. Thus, we can view matrices as infinite arrays, with only a finite upper-left portion being non-zero.

- $\Sigma$ is a 1-ary function whose argument is of type matrix, and the intended meaning is that $\Sigma$ adds up all the entries of its argument.

We will usually omit the type subscripts $_{\text{index}}$, $_{\text{field}}$ and $_{\text{matrix}}$, for the sake of readability. This is not a problem as the type will be clear from the context and the names of the metavariables involved.

## 2.2 Terms, formulas and sequents

### 2.2.1 Inductive definition of terms and formulas

We define inductively the terms and formulas over the language $\mathcal{L}_{\mathrm{LA}}$. It is customary to define terms and formulas separately, but we define them together as the terms $\mathrm{cond}_{\mathrm{index}}$ and $\mathrm{cond}_{\mathrm{field}}$ take a formula as an argument.

We use the letters $n, m$ for terms of type index, $t, u$ for terms of type field, and $T, U$ for terms of type matrix.

**Base Case:** $0_{\mathrm{index}}, 1_{\mathrm{index}}, 0_{\mathrm{field}}, 1_{\mathrm{field}}$ and variables of all three types, are all terms.

**Induction Step:**

1. If $m$ and $n$ are of type index, then $(m +_{\mathrm{index}} n)$,$(m -_{\mathrm{index}} n)$, $(m *_{\mathrm{index}} n)$, $\mathtt{div}(m, n)$, and $\mathtt{rem}(m, n)$ are all of type index.

2. If $t$ and $u$ are of type field, then $(t +_{\mathrm{field}} u)$ and $(t *_{\mathrm{field}} u)$ are of type field.

3. If $t$ is a term of type field, then $-t$ and $t^{-1}$ are terms of type field.

4. If $T$ is of type matrix, then $\mathtt{r}(T)$ and $\mathtt{c}(T)$ are of type index, and $\Sigma(T)$ is of type field.

5. If $m$ and $n$ are of type index, and $T$ is of type matrix, then $\mathtt{e}(T, m, n)$ is of type field.

6. If $m$ and $n$ are of type index, and $t$ is of type field, then $\lambda ij\langle m, n, t\rangle$ is a *constructed* term of type matrix. There is one restriction:

$$i, j \textit{ do not occur free in } m \textit{ and } n \tag{2.2}$$

   The idea behind constructed terms is to avoid having to define a whole spectrum of matrix functions (matrix addition, multiplication, subtraction, transpose, inverse, etc.). Instead, since matrices can be defined in terms of their entries (for example, matrix addition is just addition entry by entry), we use functions of type field to define matrix functions; the $\lambda$ operator allows us to do this. For example, suppose that $A$ and $B$ are $3 \times 3$ matrices. Then, $A + B$ can be defined as follows: $\lambda ij\langle 3, 3, \mathtt{e}(A, i, j) + \mathtt{e}(B, i, j)\rangle$. Incidentally, note that there is nothing that prevents us from constructing matrices with zero rows or zero columns, i.e., empty matrices.

7. If $m, n, t, u, T, U$ are terms, then:

$$m \leq_{\text{index}} n$$
$$m =_{\text{index}} n$$
$$t =_{\text{field}} u$$
$$T =_{\text{matrix}} U$$

   are formulas (called *atomic* formulas).

8. If $\alpha$ is a formula, so is $\neg\alpha$, and if $\alpha$ and $\beta$ are formulas so are $(\alpha \wedge \beta)$ and $(\alpha \vee \beta)$.

9. Suppose $\alpha$ is a formula where all atomic subformulas have the form $m \leq_{\text{index}} n$ or $m =_{\text{index}} n$, where $m$ and $n$ are terms of type index. Then, if $m', n'$ are terms of type index, then $\text{cond}_{\text{index}}(\alpha, m', n')$ is a term of type index, and if $t$ and $u$ are terms of type field, then $\text{cond}_{\text{field}}(\alpha, t, u)$ is a term of type field.

This finishes the inductive definition of terms and formulas.

The $\lambda$ is the $\lambda$-operator, and in our case it just indicates that the variables $i, j$ are *bound*. From now on, we say that an occurrence of a variable is *free* if it is not an index variable $i$ or $j$ in a subterm of $\lambda ij\langle \ldots \rangle$ (so in particular all field and matrix variables are always free), and it is bound otherwise. Note that the same index variable might occur in the same term both as a free and a bound variable.

We let $\alpha \supset \beta$ abbreviate $\neg\alpha \vee \beta$, and $\alpha \equiv \beta$ abbreviate $\alpha \supset \beta \wedge \beta \supset \alpha$.

### 2.2.2 Definition of sequents

We follow the presentation of Samuel R. Buss in [Bus98, Chapter 1]. As we mentioned in the introduction, LA is a theory of sequents, rather than a theory of formulas, because sequents are more appropriate for expressing matrix identities.

A *sequent* is written in the form:

$$\alpha_1, \ldots, \alpha_k \to \beta_1, \ldots, \beta_l \tag{2.3}$$

where the symbol $\to$ is a new symbol called the sequent arrow, and where each $\alpha_i$ and $\beta_j$ is a formula. The intuitive meaning of the sequent is that the conjunction of the $\alpha_i$'s implies the disjunction of the $\beta_j$'s. Thus, a sequent is equivalent in meaning to the formula:

$$\bigwedge_{i=1}^{k} \alpha_i \supset \bigvee_{j=1}^{l} \beta_j \tag{2.4}$$

We adopt the convention that an empty conjunction ($k = 0$ above) has value $\mathsf{True}$, and that an empty disjunction ($l = 0$ above) has value $\mathsf{False}$. Thus the sequent $\rightarrow \alpha$ has the same meaning as the formula $\alpha$, and the *empty sequent* $\rightarrow$ is false. A sequent is defined to be valid or a tautology iff its corresponding formula is.

The sequence of formulas $\alpha_1, \ldots, \alpha_k$ is called the *antecedent* of the sequent displayed above; $\beta_1, \ldots, \beta_l$ is called its *succedent*. They are both referred to as *cedents*.

The semantic equivalence between (2.3) and (2.4) holds regardless of whether the $\alpha$'s and the $\beta$'s are propositional formulas, or formulas over the language $\mathcal{L}_{\mathrm{LA}}$. However, as was mentioned in the introduction, all sequents are implicitly universally quantified, hence (2.3) is really equivalent in meaning to the formula:

$$\forall x_1 \ldots x_n \left( \bigwedge_{i=1}^{k} \alpha_i \supset \bigvee_{j=1}^{l} \beta_j \right)$$

where $x_1, \ldots, x_n$ is the list of all the free variables that appear in the sequent (2.3).

### 2.2.3 Defined terms, formulas and cedents

We use ":=" to define new objects. For example:

$$\mathtt{max}\{i, j\} := \mathrm{cond}(i \leq j, j, i)$$

denotes that $\mathtt{max}\{i, j\}$ stands for $\mathrm{cond}(i \leq j, j, i)$. This way we can simplify formulas over $\mathcal{L}_{\mathrm{LA}}$ by providing meaningful abbreviations for complicated terms. Of course, these abbreviations are there only to make derivations more human-readable, and they are not part of the language $\mathcal{L}_{\mathrm{LA}}$ (for example, $\mathtt{max}$ is not a function symbol in $\mathcal{L}_{\mathrm{LA}}$).

Since we can construct new matrix terms with $\lambda ij\langle m, n, t\rangle$, we can avoid including many operations (such as matrix addition) as primitive operations by defining them instead. For example, we can define the addition of two matrices $A$ and $B$ as follows:

$$A + B := \lambda ij\langle \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(B)\}, \mathtt{max}\{\mathtt{c}(A), \mathtt{c}(B)\}, A_{ij} + B_{ij}\rangle \tag{2.5}$$

In the above definition of addition of matrices, we used "+" instead of "$+_{\mathrm{field}}$" on the right-hand side, and on the left hand side "+" should be "$+_{\mathrm{matrix}}$", but all this is clear from the context.

We now define standard matrix functions. Let $A$ by a variable of type matrix. Then, *scalar multiplication* is defined by:

$$aA := \lambda ij\langle \mathtt{r}(A), \mathtt{c}(A), a * A_{ij}\rangle \tag{2.6}$$

and the *transpose* by:

$$A^t := \lambda ij\langle \mathtt{c}(A), \mathtt{r}(A), A_{ji}\rangle \tag{2.7}$$

The only requirement is that if $A$ is replaced by a constructed matrix term $T$, then $i$ and $j$ are *new* index variables which do not occur free in $T$.

The *zero matrix* and the *identity matrix* are defined by:

$$0_{kl} := \lambda ij\langle k, l, 0\rangle \quad \text{and} \quad I_k := \lambda ij\langle k, k, \mathrm{cond}(i = j, 1, 0)\rangle \tag{2.8}$$

respectively, where $\mathrm{cond}(i = j, 1, 0)$ expresses that $I_k$ is 1 on the diagonal and it is zero everywhere else. Sometimes we will just write 0 and $I$ when the sizes are clear from the context.

We define the *trace* function by:

$$\mathtt{tr}(A) := \Sigma\lambda ij\langle \mathtt{r}(A), 1, A_{ii}\rangle \tag{2.9}$$

Note that $\lambda ij\langle \mathtt{r}(A), 1, A_{ii}\rangle$ is a column vector consisting of the diagonal entries of $A$, and that $i, j$ are new index variables which do not occur free in $T$, if $T$ replaces $A$.

We let the *dot product* of two matrices, $A, B$, be $A \cdot B$, and we want it to be the sum of the products of corresponding entries of $A$ and $B$. Formally, we define the dot product by:

$$A \cdot B := \Sigma\lambda ij\langle \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(B)\}, \mathtt{max}\{\mathtt{c}(A), \mathtt{c}(B)\}, A_{ij} * B_{ij}\rangle \tag{2.10}$$

where $i, j$ do not occur free in $T, U$, if $T, U$ replace $A, B$.

With the dot product we can define *matrix multiplication* by letting the $(i, j)$-th entry of $A * B$ be the dot product of the $i$-th row of $A$ and the $j$-th column of $B$. Formally:

$$A * B := \lambda ij\langle \mathtt{r}(A), \mathtt{c}(B), \lambda kl\langle \mathtt{c}(A), 1, \mathtt{e}(A, i, k)\rangle \cdot \lambda kl\langle \mathtt{r}(B), 1, \mathtt{e}(B, k, j)\rangle\rangle \tag{2.11}$$

where $i, j$ do not occur freely in $T, U$, if $T, U$ replace $A, B$.

Finally, as was mentioned in the introduction, the following decomposition of an $n \times n$ matrix $A$ is going to play a prominent role in this thesis:

$$A = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

where $a_{11}$ is the $(1, 1)$ entry of $A$, and $R, S$ are $1 \times (n-1)$, $(n-1) \times 1$ submatrices, respectively, and $M$ is the principal submatrix of $A$, i.e., $M = A[1|1]$. In general, $A[i|j]$

indicates that row $i$ and column $j$ have been deleted from $A$. Therefore, we make the following precise definitions:

$$\begin{aligned} \mathtt{R}(A) &:= \lambda ij\langle 1, \mathsf{c}(A) - 1, \mathsf{e}(A, 1, i+1)\rangle \\ \mathtt{S}(A) &:= \lambda ij\langle \mathtt{r}(A) - 1, 1, \mathsf{e}(A, i+1, 1)\rangle \\ \mathtt{M}(A) &:= \lambda ij\langle \mathtt{r}(A) - 1, \mathsf{c}(A) - 1, \mathsf{e}(A, i+1, j+1)\end{aligned} \tag{2.12}$$

### 2.2.4 Substitution

Suppose that term is a term. We can indicate that a variable $x$ occurs in term by writing $\mathsf{term}(x)$. If $\mathsf{term}'$ is also a term, of the same type as the variable $x$, then $\mathsf{term}(\mathsf{term}'/x)$ denotes that the free occurrences of the variable $x$ have been replaced throughout term by $\mathsf{term}'$, and we say that $\mathsf{term}(\mathsf{term}'/x)$ is a *substitution instance* of term. If $\alpha$ is a formula, then $\alpha(\mathsf{term}'/x)$ is defined analogously.

However, the existence of bound variables complicates things, and substitution is not always as straightforward as the above paragraph would suggest. Thus, to avoid confusion, we give a precise definition of substitution, by structural induction on term:

**Basis Case:** term is just a variable $x$; in this case $x(\mathsf{term}'/x) =_{\text{synt}} \mathsf{term}'$. Note that $\mathsf{term}'$ must be of the same type as the variable $x$.

**Induction Step:** We examine items 1–9. For example, if term is of the form $(m + n)$, then $(m+n)(\mathsf{term}'/x)$ is simply $(m(\mathsf{term}'/x) + n(\mathsf{term}'/x))$. All cases, except item 6 and item 9, are just as straightforward, so we only present item 6 and item 9:

Suppose that term is of the form $\lambda ij\langle m, n, t\rangle$. If $x$ is $i$ or $j$, then the substitution has no effect, as we cannot replace bound variables. So we assume that $x$ is neither $i$ nor $j$. If $\mathsf{term}'$ does not contain $i$ or $j$, then $\lambda ij\langle m, n, t\rangle(\mathsf{term}'/x)$ is just:

$$\lambda ij\langle m(\mathsf{term}'/x), n(\mathsf{term}'/x), t(\mathsf{term}'/x)\rangle \tag{2.13}$$

If, on the other hand, $\mathsf{term}'$ contains $i$ or $j$, then, if we substituted carelessly as in (2.13), the danger arises that $x$ might occur in $m$ or $n$, and we would violate restriction (2.2). Furthermore, if $x$ also occurs in $t$, then the $i$ and $j$ from $\mathsf{term}'$ would "get caught" in the scope of the $\lambda$-operator, and change the semantics of $t$ in an unwanted way.

Thus, if $\mathsf{term}'$ contains $i$ or $j$, then, to avoid the problems listed in the above paragraph, we rename $i, j$ in $\lambda ij\langle m, n, t\rangle$ to new index variables $i', j'$, and carry on as in (2.13).

A detailed exposition of substitution and $\lambda$ calculus can be found, for example, in [HS86].

Suppose that $\mathsf{term}$ is of the form $\mathrm{cond}_{\mathrm{index}}(\alpha, m, n)$. Then the result of replacing $x$ by $\mathsf{term}'$ is simply:

$$\mathrm{cond}_{\mathrm{index}}(\alpha(x/\mathsf{term}'), m(\mathsf{term}'/x), n(\mathsf{term}'/x))$$

Note that the only worry is whether $\alpha(\mathsf{term}'/x)$ continues to be a boolean combination of atomic formulas with terms of type index (see item 9 above). But this is not a problem as we require $\mathsf{term}'$ to be of the same type as the variable $x$, so, and this can be proven by induction, substitution does not change the type of the term.

**Lemma 2.2.1** Every substitution instance of a term is a term (of the same type). Similarly, every substitution instance of a formula is a formula, and every substitution instance of a sequent is a sequent.

*Proof.* Immediate from the above inductive definition of substitution. □

We end this section with some more terminology: if $\mathsf{term}, \mathsf{term}_1, \ldots, \mathsf{term}_k$ are terms, and $x_1, \ldots, x_k$ are variables, where $x_i$ is of the same type as $\mathsf{term}_i$, then:

$$\mathsf{term}(\mathsf{term}_1/x_1, \ldots, \mathsf{term}_k/x_k)$$

denotes the *simultaneous* substitution of $\mathsf{term}_i$ for $x_i$. On the other hand,

$$\mathsf{term}(\mathsf{term}_1/x_1) \ldots (\mathsf{term}_k/x_k)$$

denotes a *sequential* substitution, where we first replace all instances of $x_1$ by $\mathsf{term}_1$, then we replace all instances of $x_2$ in $\mathsf{term}(\mathsf{term}_1/x_1)$ by $\mathsf{term}_2$, and so on. We have analogous conventions for formulas and sequents.

### 2.2.5 Standard models

In this section we define standard models for formulas over $\mathcal{L}_{\mathrm{LA}}$; we follow the terminology and style of [Bus98, chapter 2.1.2.]. We do not define general models as we do not need them. A *standard model* is a structure where the universe for terms of type index is $\mathbb{N}$, the universe for terms of type field is $\mathbb{F}$, for some fixed field $\mathbb{F}$, and the universe for terms of type matrix is $M(\mathbb{F}) = \bigcup_{m,n\in\mathbb{N}} M_{m\times n}(\mathbb{F})$ and $M_{m\times n}(\mathbb{F})$ is the set of $m \times n$ matrices over the field $\mathbb{F}$. The standard model is denoted by $\mathcal{S}_{\mathbb{F}}$. All operations are given the

standard meaning by $\mathcal{S}_{\mathbb{F}}$ ("$-_{\text{index}}$" is cut-off subtraction). We define $0^{-1}$ to be 0, and $\mathtt{div}(i,j)$ and $\mathtt{rem}(i,j)$ are undefined when $j = 0$.

If $\alpha$ is a formula over $\mathcal{L}_{\text{LA}}$ without free variables, i.e. $\alpha$ is a *sentence*, then we write $\mathcal{S}_{\mathbb{F}} \models \alpha$ to denote that $\alpha$ is true in the structure $\mathcal{S}_{\mathbb{F}}$. However, formulas over $\mathcal{L}_{\text{LA}}$ may have free variables in them. Thus, to give meaning to a general formula $\alpha$ we not only need a structure $\mathcal{S}_{\mathbb{F}}$, but also an *object assignment*, which is a mapping $\tau$ from the set of variables (at least the ones free in $\alpha$) to the universe of $\mathcal{S}_{\mathbb{F}}$. That is, $\tau$ assigns values from $\mathbb{N}$ to all the free index variables, values from $\mathbb{F}$ to all the field variables, and matrices over $\mathbb{F}$ to all the matrix variables.

We write $\mathcal{S}_{\mathbb{F}} \models \alpha[\tau]$ to denote that $\alpha$ is true in the structure $\mathcal{S}_{\mathbb{F}}$ with the given object assignment $\tau$. To give a formal definition of $\mathcal{S}_{\mathbb{F}} \models \alpha[\tau]$, we first need to define the interpretation of terms, i.e. we need to formally define the manner in which arbitrary terms represent objects in the universe of $\mathcal{S}_{\mathbb{F}}$. To this end, we define $\mathsf{term}^{\mathcal{S}}[\tau]$, for a given $\mathcal{S} = \mathcal{S}_{\mathbb{F}}$, by structural induction:

**Basis Case:** $\mathsf{term}$ is a variable of one of the three sorts, or a constant. For example, if $\mathsf{term}$ is $i$, then $i^{\mathcal{S}}[\tau]$ is just $\tau(i) \in \mathbb{N}$.

**Induction Step:** Suppose that $\mathsf{term}$ is of the form $(m +_{\text{index}} n)$. Then, $(m +_{\text{index}} n)^{\mathcal{S}}[\tau] = m^{\mathcal{S}}[\tau] + n^{\mathcal{S}}[\tau]$, where "+" denotes the usual addition of natural numbers. Similarly we can deal with multiplication, and the basic operations of field elements.

Suppose that $\mathsf{term}$ is of the form $\mathtt{r}(T)$. Then $(\mathtt{r}(T))^{\mathcal{S}}[\tau]$ is the number of rows of $T^{\mathcal{S}}[\tau]$, which is the number of rows of $\tau(A)$ if $T$ is the matrix variable $A$, and it is $m^{\mathcal{S}}[\tau]$, if $T$ is of the form $\lambda ij\langle m, n, t\rangle$.

Suppose that $\mathsf{term}$ is of the form $\mathtt{e}(T, m, n)$. Then $(\mathtt{e}(T, m, n))^{\mathcal{S}}[\tau]$ is the entry $(m^{\mathcal{S}}[\tau], n^{\mathcal{S}}[\tau])$ of the matrix $T^{\mathcal{S}}[\tau]$ (and it is zero if one of the parameters is out of bounds).

All other cases can be dealt with similarly.

Since all free variables in a formula $\alpha$ are implicitly universally quantified, we say that $\alpha$ is *true* in the standard model, denoted $\mathcal{S}_{\mathbb{F}} \models \alpha$, if $\mathcal{S}_{\mathbb{F}} \models \alpha[\tau]$ for all object assignments $\tau$.

## 2.3 Axioms

In this section we present all the axioms of the theory LA. The axioms are divided into four groups: equality axioms (section 2.3.1), the axioms for indices which are the axioms of Peano's Arithmetic *without* induction (section 2.3.2), the axioms for field elements (section 2.3.3), and the axioms for matrices (section 2.3.4). We have the following axiom convention:

$$\textit{All substitution instances of axioms are also axioms.} \tag{2.14}$$

Thus, our axioms are really axiom schemas.

### 2.3.1 Equality Axioms

We have the usual equality axioms. The symbol "=" is a metasymbol for one of the three equality symbols, and the variables $x, y$ are meta-metavariables, that is, they stand for one of the three types of standard metavariables. The function symbol $f$ in A4 is one of the function symbols of $\mathcal{L}_{\mathrm{LA}}$, given in Table 2.1, and $n$ is the corresponding arity.

**A1** $\to x = x$

**A2** $x = y \to y = x$

**A3** $(x = y \wedge y = z) \to x = z$

**A4** $x_1 = y_1, \ldots, x_n = y_n \to f x_1 \ldots x_n = f y_1 \ldots y_n$

**A5** $i_1 = j_1, i_2 = j_2, i_1 \leq i_2 \to j_1 \leq j_2$

Table 2.2: Equality axioms

**Example 2.3.1** A particular instance of A4 would be:

$$i_1 = i_2, j_1 = j_2, A = B \to \mathsf{e}(A, i_1, j_1) = \mathsf{e}(B, i_2, j_2)$$

Here $f = \mathsf{e}$, and since $\mathsf{e}$ has arity 3, $n = 3$.

### 2.3.2 Axioms for indices

The axioms for indices are the usual axioms of Peano's Arithmetic *without* induction[1], with A15 for cut-off subtraction definitions, (note that $i \not\leq j$ abbreviates $\neg(i \leq j)$), A16 for the quotient and reminder function definitions, and A17 for the conditional function definitions (recall that $\alpha$ has to satisfy the restriction of item 9 given in Section 2.2.1).

| | |
|---|---|
| **A6** $\rightarrow i+1 \neq 0$ | **A11** $\rightarrow i \leq j, j \leq i$ |
| **A7** $\rightarrow i*(j+1) = (i*j)+i$ | **A12** $\rightarrow i+(j+1) = (i+j)+1$ |
| **A8** $i+1 = j+1 \rightarrow i = j$ | **A13** $i \leq j, j \leq i \rightarrow i = j$ |
| **A9** $\rightarrow i \leq i+j$ | **A14** $\rightarrow i*0 = 0$ |
| **A10** $\rightarrow i+0 = i$ | |

**A15** $i \leq j, i+k = j \rightarrow j-i = k$ **and** $i \not\leq j \rightarrow j-i = 0$

**A16** $j \neq 0 \rightarrow \mathtt{rem}(i,j) < j$ **and** $j \neq 0 \rightarrow i = j*\mathtt{div}(i,j)+\mathtt{rem}(i,j)$

**A17** $\alpha \rightarrow \mathrm{cond}(\alpha,i,j) = i$ **and** $\neg\alpha \rightarrow \mathrm{cond}(\alpha,i,j) = j$

Table 2.3: Axioms for indices

[1] Thus, the index fragment of LA *does not* correspond to Peano Arithmetic, since LA has no quantifiers, and the induction (introduced later in this chapter as a rule) is on quantifier-free formulas.

### 2.3.3 Axioms for field elements

The axioms for the field elements are the usual field axioms, plus A27 for the conditional function definition (recall that $\alpha$ has to satisfy the restriction of item 9 given in Section 2.2.1).

| | |
|---|---|
| **A18** $\rightarrow 0 + a = a$ | **A23** $\rightarrow a * b = b * a$ |
| **A19** $\rightarrow a + (-a) = 0$ | **A24** $\rightarrow a + (b + c) = (a + b) + c$ |
| **A20** $\rightarrow 1 * a = a$ | **A25** $\rightarrow a * (b * c) = (a * b) * c$ |
| **A21** $a \neq 0 \rightarrow a * (a^{-1}) = 1$ | **A26** $\rightarrow a * (b + c) = a * b + a * c$ |
| **A22** $\rightarrow a + b = b + a$ | |

**A27** $\alpha \rightarrow \text{cond}(\alpha, a, b) = a$ **and** $\neg\alpha \rightarrow \text{cond}(\alpha, a, b) = b$

Table 2.4: Axioms for field elements

### 2.3.4 Axioms for matrices

In this section we define the last six axioms which govern the behavior of matrices. Axiom A28 states that $\mathtt{e}(A, i, j)$ is zero when $i, j$ are outside the size of $A$. Axiom A29 defines the behavior of constructed matrices. Axioms A30–A33 define the function $\Sigma$ recursively as follows:

- First, A30 and A31, we define $\Sigma$ for *row vectors*, that is for matrices of the form:

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}$$

  If $n = \mathtt{c}(A) = 1$, so $A = (a)$, then $\Sigma((a)) = a$. Suppose $\mathtt{r}(A) = 1 \wedge \mathtt{c}(A) > 1$. In that case we define $\Sigma$ as follows:

$$\Sigma(A) = \Sigma \begin{pmatrix} a_1 & \dots & a_n \end{pmatrix} = \Sigma \begin{pmatrix} a_1 & \dots & a_{n-1} \end{pmatrix} + a_n$$

- If $A$ is a *column vector*, A32, then $A^t$ is a row vector, and so $\Sigma(A) = \Sigma(A^t)$ which is already defined.

- In A33, we extend $\Sigma$ to all matrices. Suppose that $\mathtt{r}(A) > 1$ and $\mathtt{c}(A) > 1$, that is:

$$A = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

Then, $\Sigma$ is defined recursively as follows:

$$\Sigma(A) = a_{11} + \Sigma(R) + \Sigma(S) + \Sigma(M) \tag{2.15}$$

Note that throughout $m < n$ is an abbreviation for $(m \leq n \wedge m \neq n)$, and, of course, $m \neq n$ is an abbreviation for $\neg(m = n)$. Finally, see (2.7) for the precise definition of $A^t$ in A32, and see (2.12), page 20, for definitions of the terms $\mathtt{R}(A), \mathtt{S}(A), \mathtt{M}(A)$ in A33.

**A28** $(i = 0 \vee \mathtt{r}(A) < i \vee j = 0 \vee \mathtt{c}(A) < j) \to \mathtt{e}(A, i, j) = 0$

**A29** $\to \mathtt{r}(\lambda ij\langle m, n, t\rangle) = m$ **and** $\to \mathtt{c}(\lambda ij\langle m, n, t\rangle) = n$ **and**
$1 \leq i, i \leq m, 1 \leq j, j \leq n \to \mathtt{e}(\lambda ij\langle m, n, t\rangle, i, j) = t$

**E**[a] $\mathtt{r}(A) = 0 \vee \mathtt{c}(A) = 0 \to \Sigma A = 0$

**A30** $\mathtt{r}(A) = 1, \mathtt{c}(A) = 1 \to \Sigma(A) = \mathtt{e}(A, 1, 1)$

**A31** $\mathtt{r}(A) = 1, 1 < \mathtt{c}(A) \to \Sigma(A) = \Sigma(\lambda ij\langle 1, \mathtt{c}(A) - 1, A_{ij}\rangle) + A_{1\mathtt{c}(A)}$

**A32**[b] $\mathtt{c}(A) = 1 \to \Sigma(A) = \Sigma(A^t)$

**A33**[c] $1 < \mathtt{r}(A), 1 < \mathtt{c}(A) \to \Sigma(A) = \mathtt{e}(A, 1, 1) + \Sigma(\mathtt{R}(A)) + \Sigma(\mathtt{S}(A)) + \Sigma(\mathtt{M}(A))$

[a] The axiom **E**(mpty) is necessary to take care of empty matrices—matrices with zero rows or zero columns. There is nothing that prevents us from construction a matrix $\lambda ij\langle 0, 3, t\rangle$, for example, and we want $\Sigma$ of such a matrix to be $0_{\text{field}}$, regardless of $t$.

[b] See page 19 for the definition of $A^t$.

[c] See page 20 for the definitions of $\mathtt{R}, \mathtt{S}, \mathtt{M}$.

Table 2.5: Axioms for matrices

## 2.4 Rules of inference and proof systems

We start by defining the propositional sequent calculus proof system PK, following loosely the presentation in [Bus98, Chapter 1].

A PK proof consists of an ordered sequence of sequents $\{S_1, \ldots, S_n\}$, where $S_n$ is the *endsequent* and it is the sequent proved by the proof. All sequents in $\{S_1 \ldots, S_n\}$ are either *initial sequents* of the form $\alpha \to \alpha$, for any formula $\alpha$, or follow by one of the rules for propositional consequence (defined below) from previous sequents in the proof.

**Definition 2.4.1** A *rule of inference* is denoted by a figure:

$$\frac{S_1}{S} \qquad \frac{S_1 \quad S_2}{S} \qquad \frac{S_1 \quad S_2 \quad S_3}{S}$$

indicating that the sequent $S$ may be inferred from $S_1$, or from the pair $S_1$ and $S_2$, or from the triple $S_1$ and $S_2$ and $S_3$. The conclusion, $S$, is called the *lower sequent* of the inference; each hypotheses is an *upper sequent* of the inference.

**Definition 2.4.2** The rules in Tables 2.6, 2.7, and 2.8, are the PK *rules for propositional consequence*. These rules are essentially schematic, in that $\alpha$ and $\beta$ denote arbitrary formulas and $\Gamma, \Delta$ denote arbitrary cedents.

| | |
|---|---|
| exchange-left: $\dfrac{\Gamma_1, \alpha, \beta, \Gamma_2 \to \Delta}{\Gamma_1, \beta, \alpha, \Gamma_2 \to \Delta}$ | exchange-right: $\dfrac{\Gamma \to \Delta_1, \alpha, \beta, \Delta_2}{\Gamma \to \Delta_1, \beta, \alpha, \Delta_2}$ |
| contraction-left: $\dfrac{\alpha, \alpha, \Gamma \to \Delta}{\alpha, \Gamma \to \Delta}$ | contraction-right: $\dfrac{\Gamma \to \Delta, \alpha, \alpha}{\Gamma \to \Delta, \alpha}$ |
| weakening-left: $\dfrac{\Gamma \to \Delta}{\alpha, \Gamma \to \Delta}$ | weakening-right: $\dfrac{\Gamma \to \Delta}{\Gamma \to \Delta, \alpha}$ |

Table 2.6: Weak structural rules

$$\frac{\Gamma \to \Delta, \alpha \quad \alpha, \Gamma \to \Delta}{\Gamma \to \Delta}$$

Table 2.7: Cut rule

The PK system, as a *propositional proof system*, is *sound* and *complete*, that is to say, any PK-provable sequent is a propositional tautology, and every propositionally valid

$$\neg\text{-left: } \frac{\Gamma \to \Delta, \alpha}{\neg\alpha, \Gamma \to \Delta} \qquad \neg\text{-right: } \frac{\alpha, \Gamma \to \Delta}{\Gamma \to \Delta, \neg\alpha}$$

$$\wedge\text{-left: } \frac{\alpha, \beta, \Gamma \to \Delta}{\alpha \wedge \beta, \Gamma \to \Delta} \qquad \wedge\text{-right: } \frac{\Gamma \to \Delta, \alpha \quad \Gamma \to \Delta, \beta}{\Gamma \to \Delta, \alpha \wedge \beta}$$

$$\vee\text{-left: } \frac{\alpha, \Gamma \to \Delta \quad \beta, \Gamma \to \Delta}{\alpha \vee \beta, \Gamma \to \Delta} \qquad \vee\text{-right: } \frac{\Gamma \to \Delta, \alpha, \beta}{\Gamma \to \Delta, \alpha \vee \beta}$$

Table 2.8: Rules for introducing connectives

sequent (tautology) has a PK-proof. For a proof of this, see theorems 1.2.6 and 1.2.8 in [Bus98, Chapter 1].

We now define the sequent calculus proof system PK-LA. Besides the rules for propositional consequence, we need a rule for induction on indices, and a rule for concluding equality of matrices.

**Definition 2.4.3** Recall that $\alpha(\mathsf{term}/x)$ denotes that *every* occurrence of the variable $x$ in $\alpha$ is replaced by the term $\mathsf{term}$ (note that $\mathsf{term}$ must be of the same type as the variable $x$). Thus we define the *induction rule* as in Table 2.9; note that $i$ must be an

$$\frac{\Gamma, \alpha(i) \to \alpha(i+1/i), \Delta}{\Gamma, \alpha(0/i) \to \alpha(n/i), \Delta}$$

Table 2.9: Induction Rule

index variable (as we only allow induction on indices), and $n$ is any term of type index. We have induction on indices because we want to prove matrix identities by induction on the size of the matrices involved.

**Definition 2.4.4** The *matrix equality rules* are defined in Table 2.10; the only restriction

$$\text{left: } \frac{\mathtt{r}(T) = \mathtt{r}(U), \mathtt{c}(T) = \mathtt{c}(U), \mathtt{e}(T, i, j) = \mathtt{e}(U, i, j), \Gamma \to \Delta}{T{=}U, \Gamma \to \Delta}$$

$$\text{right: } \frac{\Gamma \to \Delta, \mathtt{e}(T, i, j) = \mathtt{e}(U, i, j) \quad \Gamma \to \Delta, \mathtt{r}(T) = \mathtt{r}(U) \quad \Gamma \to \Delta, \mathtt{c}(T) = \mathtt{c}(U)}{\Gamma \to \Delta, T{=}U}$$

Table 2.10: Equality Rules

is that $i, j$ do not occur free in the bottom sequent of Equality right. Note that three

types of equalities appear in this rule: equality of indices, field elements, and matrices. (As usual, for the sake of readability, we omit the corresponding subscripts). Note that we have the "reverse" of the equality rule by using axiom A4.

**Definition 2.4.5** We define the proof system PK-LA to be a system of sequent calculus proofs, where all the initial sequents are either of the form $\alpha \to \alpha$ (for any formula $\alpha$ over $\mathcal{L}_{\mathrm{LA}}$), or are given by one of the axiom schemas A1–33, and all the other sequents (if any) follow from previous sequents in the proof by one of the PK rules for propositional consequence, or by Ind, or by Eq.

Thus, a PK-LA proof of a sequent $S$ consists of an ordered sequence of sequents $\{S_1, \ldots, S_n\}$, where each $S_i$ is either of the form $\alpha \to \alpha$, or is given by one of the axiom schemas A1–33, or follows from previous $S_j$'s by a PK rule for propositional consequence, or by Ind, or by Eq. The endsequent, $S_n$ is $S$. The *length* of this derivation is $n$.

**Definition 2.4.6** The theory LA is the set of sequents over $\mathcal{L}_{\mathrm{LA}}$ which have PK-LA derivations.

Note that, in particular, all the sequents given by the axiom schemas A1–33 are in LA.

**Definition 2.4.7** The *substitution rule* is given in Table 2.11; $S$ is any sequent, and

$$\text{Subst:} \quad \frac{S(x_1, \ldots, x_k)}{S(\mathsf{term}_1/x_1, \ldots, \mathsf{term}_k/x_k)}$$

Table 2.11: The *derived* Substitution Rule

$S(x_1, \ldots, x_k)$ indicates that $x_1, \ldots, x_k$ are variables in $S$. Recall that the expression $S(\mathsf{term}_1/x_1, \ldots, \mathsf{term}_k/x_k)$ indicates that the terms $\mathsf{term}_1, \ldots, \mathsf{term}_k$ replace all *free* occurrences of the variables $x_1, \ldots, x_k$ in $S$, simultaneously. Here, $x_i$ has any of the three types, and the term $\mathsf{term}_i$ has the same type as $x_i$.

**Lemma 2.4.1** LA is closed under the substitution rule.

*Proof.* We prove the lemma by induction on the length of a derivation of the sequent $S$. Basis Case: If $S$ is an axiom of LA, then by the axiom convention (2.14) in section 2.3, all the substitution instances of $S$ are also axioms of LA.

Induction Step: $S$ is derived by one of the rules (by one of the rules for propositional consequence, by Ind, or by Eq). Suppose $S$ is obtained by Ind. Then $S =_{\text{synt}} \Gamma \to \alpha(n/i), \Delta$, and it is obtained as follows:

$$\frac{\Gamma, \alpha(0/i), \alpha(i) \to \alpha(i+1/i), \Delta}{\Gamma \to \alpha(n/i), \Delta} \tag{2.16}$$

and $x_1, \ldots, x_k$ is a list of variables that occur in $\Gamma \to \alpha(n/i), \Delta$. The first thing we do is replace $i$ in the premiss of (2.16) by a new variable $i'$. Note that this can be done by our induction hypothesis. Now we can present the derivation of $\Gamma, \alpha(n/i) \to \Delta$ as follows:

$$\frac{\Gamma, \alpha(0/i'), \alpha(i') \to \alpha(i'+1/i'), \Delta}{\Gamma \to \alpha(n/i'), \Delta} \tag{2.16$'$}$$

Note that (2.16$'$) is still a valid induction rule. Now we replace $x_1, \ldots, x_k$ in (2.16$'$) by $\mathsf{term}_1, \ldots, \mathsf{term}_k$. Note that since $i'$ is a new variable, it was not replaced by any of the terms $\mathsf{term}_1, \ldots, \mathsf{term}_n$. Thus, we obtained a derivation of:

$$(\Gamma \to \alpha(n/i'), \Delta)(\mathsf{term}_1/x_1, \ldots, \mathsf{term}_k/x_k)$$

which is just $S(\mathsf{term}_1/x_1, \ldots, \mathsf{term}_k/x_k)$.

Suppose $S$ is of the form $\Gamma \to \Delta, T = U$ and it is obtained by the equality rule. We proceed similarly to the induction rule case: we replace $i, j$ by two new variables $i', j'$ which do not occur in $x_1, \ldots, x_k$. Again, we can do this by the induction hypothesis. Then, we replace $x_1, \ldots, x_k$ throughout in the rule by $\mathsf{term}_1, \ldots, \mathsf{term}_k$, and we are done.

Finally, if $S$ is obtained by a rule for propositional consequence, then we just replace $x_1, \ldots, x_k$ throughout the rule by $\mathsf{term}_1, \ldots, \mathsf{term}_k$. $\square$

# Chapter 3

# The Theorems of LA

In this chapter we will show that all the basic properties of matrices can be proven in LA. More precisely, we will show that all the matrix identities which state that the set of $n \times n$ matrices is a ring, and all the matrix identities that state that the set of $m \times n$ matrices is a module over the underlying field, are theorems of LA.

The conclusion is that all the basic matrix manipulations can be proven correct in LA. By "basic" we mean for example the associativity of matrix multiplication. However, LA is apparently not strong enough to prove matrix identities which require arguing about inverses; thus, it seems that LA is not strong enough to prove $AB = I \rightarrow BA = I$.

One approach to show the independence of $AB = I \rightarrow BA = I$ from LA is by constructing a model $\mathcal{M}$ of LA that does not satisfy $AB = I \rightarrow BA = I$. A less promising approach would be to show that $AB = I \rightarrow BA = I$ has no short Frege proofs (whereas all the theorems of LA have short Frege proofs; see Chapter 7). In any case, the independence of $AB = I \rightarrow BA = I$ from LA is stated as open problem 9.1.

In Section 3.2 we show that LA proves the equivalence of several hard matrix identities. This is an interesting result, as LA seems too weak to prove the identities themselves. We also show that LA can prove combinatorial results (The Odd Town Theorem is given here) that rely on "linear-independence results" from hard matrix identities.

## 3.1 LA proofs of basic matrix identities

We will use the following strategy to prove that $T = U$: we first show that $\mathtt{r}(T) = \mathtt{r}(U)$ and $\mathtt{c}(T) = \mathtt{c}(U)$, and then we show $\mathtt{e}(T, i, j) = \mathtt{e}(U, i, j)$, from which we can conclude that $T = U$ invoking the equality rule. Thus, we are showing equality of two matrices

by showing that they have the same size and the same entries. We will omit the proof of $\mathtt{c}(T) = \mathtt{c}(T)$ as in all cases it is analogous to the proof of $\mathtt{r}(T) = \mathtt{r}(U)$.

For the sake of readability we will omit "$*$" (the multiplication symbol), as it will always be clear from the context when does multiplication apply, and what type of multiplication is being used (product of indices, field elements or of matrices).

Recall that the formula $\alpha$ is equivalent in meaning to the sequent $\to \alpha$. Therefore, we can omit the arrow, but formally LA is a theory of sequents, and so the arrow is there. Also, our derivations are informal; recall that a sequent $S$ is in LA iff it has a PK-LA derivation. However, providing complete PK-LA derivations would be tedious and unnecessary, so we derive all theorems below informally, sometimes giving informal justifications in the right margin, but we keep in mind that these informal derivations can be formalized in PK-LA.

### 3.1.1 Ring properties

**T1** $A + 0_{\mathtt{r}(A)\mathtt{c}(A)} = A$

*Proof.* $\mathtt{r}(A + 0_{\mathtt{r}(A)\mathtt{c}(A)}) = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(0_{\mathtt{r}(A)\mathtt{c}(A)})\} = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(A)\} = \mathtt{r}(A)$, and the entries: $\mathtt{e}(A + 0_{\mathtt{r}(A)\mathtt{c}(A)}, i, j) = A_{ij} + 0 = A_{ij}$. □

**T2** $A + (-1)A = 0_{\mathtt{r}(A)\mathtt{c}(A)}$

*Proof.* $\mathtt{r}(A + (-1)A) = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}((-1)A)\} = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(A)\} = \mathtt{r}(A) = \mathtt{r}(0_{\mathtt{r}(A)\mathtt{c}(A)})$, and the entries: $\mathtt{e}(A + (-1)A, i, j) = A_{ij} + (-1)A_{ij} = 0$. □

To prove the commutativity and associativity of matrix addition we need to prove two properties of $\mathtt{max}$; hence T3 and T5.

**T3** $\mathtt{max}\{i, j\} = \mathtt{max}\{j, i\}$

*Proof.* We have to prove that $\mathrm{cond}(i \leq j, j, i) = \mathrm{cond}(j \leq i, i, j)$. We introduced the following abbreviation: $i < j$ stands for $i \leq j \wedge i \neq j$. Then, by A11, we have that

$$i < j \vee i = j \vee j < i$$

To see this just note that $i \leq j$ propositionally implies $(i \leq j \wedge i \neq j) \vee i = j$.

We now consider each of the three cases in $i < j \vee i = j \vee j < i$ separately. If $i = j$, then by A13, $i \leq j$ and $j \leq i$, so $\mathrm{cond}(i \leq j, j, i) = j$ and $\mathrm{cond}(j \leq i, i, j) = i$, where

we used A17, but since $i = j$, using the equality axioms we have that $\text{cond}(i \leq j, j, i) = \text{cond}(j \leq i, i, j)$, and we are done.

Consider the case $i < j$. Then $i \leq j$, so, by A17, $\text{cond}(i \leq j, j, i) = j$. Now, if $i < j$, then $\neg j \leq i$. To see this, suppose that $i < j \wedge j \leq i$. Then, $i \leq j \wedge i \neq j \wedge j \leq i$, so, by A13, $i = j \wedge i \neq j$, contradiction. Thus $\neg j \leq i$. From this we have, by A17, that $\text{cond}(j \leq i, i, j) = j$, and again, by equality axioms we are done.

The case $j < i$ can be done similarly, and we are done. □

Now we can prove the commutativity of matrix addition:

**T4** $A + B = B + A$

*Proof.* $\mathtt{r}(A + B) = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(B)\}$ and by T3, this is equal to $\mathtt{max}\{\mathtt{r}(B), \mathtt{r}(A)\} = \mathtt{r}(B + A)$. Since addition of field elements is commutative (A22), we can conclude that: $\mathtt{e}(A + B, i, j) = A_{ij} + B_{ij} = B_{ij} + A_{ij} = \mathtt{e}(B + A, i, j)$. □

**T5** $\mathtt{max}\{i, \mathtt{max}\{j, k\}\} = \mathtt{max}\{\mathtt{max}\{i, j\}, k\}$

**T6** $A + (B + C) = (A + B) + C$

*Proof.* $\mathtt{r}(A + (B + C)) = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(B + C)\} = \mathtt{max}\{\mathtt{r}(A), \mathtt{max}\{\mathtt{r}(B), \mathtt{r}(C)\}\}$ and by T5, $\mathtt{max}\{\mathtt{r}(A), \mathtt{max}\{\mathtt{r}(B), \mathtt{r}(C)\}\} = \mathtt{max}\{\mathtt{max}\{\mathtt{r}(A), \mathtt{r}(B)\}, \mathtt{r}(C)\}$, which is equal to $\mathtt{r}((A + B) + C)$. Since addition of field elements is associative (A22), we have that: $\mathtt{e}(A + (B + C), i, j) = A_{ij} + (B_{ij} + C_{ij}) = (A_{ij} + B_{ij}) + C_{ij} = \mathtt{e}((A + B) + C, i, j)$ □

Before we prove the next theorem, we outline a strategy for proving claims about matrices by induction on their size. The first thing to note is that it is possible to define empty matrices (matrices with zero rows or zero columns), but we consider such matrices to be special. Our theorems hold for this special case, by axioms A28 and **E** on page 26, so we will always implicitly assume that it holds. Thus, the Basis Case in the inductive proofs that will follow, is when there is one row (or one column). Therefore, instead of doing induction on $i$ (see page 28 for the Induction Rule), we do induction on $j$, where $i = j + 1$.

Also note that the size of a matrix has two parameters: the number of rows, and the number of columns. We deal with this problem as follows: suppose that we want to prove

something for all matrices $A$. We define a new (constructed) matrix $M(i, A)$ as follows: first let $d(A)$ be:

$$d(A) := \text{cond}(\mathtt{r}(A), \mathtt{c}(A), \mathtt{r}(A) \leq \mathtt{c}(A))$$

that is, $d(A) = \min\{\mathtt{r}(A), \mathtt{c}(A)\}$. Now let:

$$M(i, A) := \lambda pq\langle \mathtt{r}(A) - d(A) + i, \mathtt{c}(A) - d(A) + i, \mathtt{e}(A, d(A) - i + p, d(A) - i + q)\rangle$$

that is, $M(i, A)$ is the $i$-th principal submatrix of $A$. For example, if $A$ is a $3 \times 5$ matrix, then $M(1, A)$ is a $1 \times 3$ matrix, with the entries from the lower-right corner of $A$.

To prove that a property $\mathcal{P}$ holds for $A$, we prove that $\mathcal{P}$ holds for $M(1, A)$ (Basis Case), and we prove that if $\mathcal{P}$ holds for $M(i, A)$, it also holds for $M(i + 1, A)$ (Induction Step). From this we conclude, by the induction rule, that $\mathcal{P}$ holds for $M(d(A), A)$, and $M(d(A), A)$ is just $A$. Note that in the Basis Case we might have to prove that $\mathcal{P}$ holds for a row vector or a column vector, which is a $k \times 1$ or a $1 \times k$ matrix, and this in turn can also be done by induction (on $k$).

**T7** $\Sigma 0_{kl} = 0_{\text{field}}$

*Proof.* We prove this theorem in considerable detail, making use of the induction strategy outlined above. Recall that $0_{kl}$ abbreviates $\lambda pq\langle k, l, 0_{\text{field}}\rangle$, so $\mathtt{r}(0_{kl}) = k$ and $\mathtt{c}(0_{kl}) = l$, and so $d(0_{kl})$ is just $\min\{k, l\}$. The matrix $M(i, 0_{kl})$ is given by:

$$\lambda pq\langle k - \min\{k, l\} + i, l - \min\{k, l\} + i, \mathtt{e}(0_{kl}, \min\{k, l\} - i + p, \min\{k, l\} - i + q)\rangle$$

Since for all $p, q$ we have $\mathtt{e}(0_{kl}, \min\{k, l\} - i + p, \min\{k, l\} - i + q) = 0_{\text{field}}$, using the equality rule we can show that $M(i, 0_{kl}) = 0_{(k-\min\{k,l\}+i)(l-\min\{k,l\}+i)}$. Therefore, we now want to show by induction on $i$ that:

$$\Sigma 0_{(k-\min\{k,l\}+i)(l-\min\{k,l\}+i)} = 0_{\text{field}}$$

**Basis Case:** $i = 1$. Depending on whether or not $k \leq l$, $0_{(k-\min\{k,l\}+i)(l-\min\{k,l\}+i)}$ is a row vector of zeros, or a column vector of zeros. Assume first that $k \leq l$, and show that $\Sigma 0_{1j} = 0_{\text{field}}$, by induction on $j$. The Basis Case is $j = 1$, in which case $\mathtt{r}(0_{1j}) = \mathtt{c}(0_{1j}) = 1$, so we can use A30 to conclude that $\Sigma 0_{1j} = \mathtt{e}(0_{1j}, 1, 1) = 0_{\text{field}}$. For the Induction Step, assume that $\Sigma 0_{1j} = 0_{\text{field}}$ is true, for $j \geq 1$. By A31, and making use of the equality rule, we have that:

$$\Sigma 0_{1(j+1)} = \Sigma 0_{1j} + 0_{\text{field}}$$

By the induction hypothesis, $\Sigma 0_{1j} = 0_{\text{field}}$, and by A18, $0_{\text{field}} + 0_{\text{field}} = 0_{\text{field}}$, and therefore $\Sigma 0_{1(j+1)} = 0_{\text{field}}$. Now, with the induction rule we can conclude that $\Sigma 0_{1j} = 0_{\text{field}}$ holds for $j = l - \min\{k,l\} + 1$. If $k > l$, we can prove the result with A32 (by simply taking the transpose of $0_{1j}$ to prove that $\Sigma 0_{j1} = 0_{\text{field}}$).

**Induction Step:** Assume that $\Sigma 0_{(k-\min\{k,l\}+i)(l-\min\{k,l\}+i)} = 0_{\text{field}}$ holds for $i \geq 1$, and show that it holds for $i+1$. To show this we use the equality axioms (to show that $\lambda pq\langle k-1, l-1, \mathtt{e}(0_{kl}, p, q)\rangle = 0_{(k-1)(l-1)}$), and A33. Thus:

$$\begin{aligned}\Sigma 0_{(k-\min\{k,l\}+i+1)(l-\min\{k,l\}+i+1)} &= \Sigma 0_{1(l-\min\{k,l\}+i)} + \Sigma 0_{(k-\min\{k,l\}+i)1} \\ &\quad + \Sigma 0_{(k-\min\{k,l\}+i)(l-\min\{k,l\}+i)}\end{aligned}$$

The first two terms of the RHS are $0_{\text{field}}$ by the Basis Case. The last term of the RHS is $0_{\text{field}}$ by the induction hypothesis. Thus, by A18, $\Sigma 0_{(k-\min\{k,l\}+i+1)(l-\min\{k,l\}+i+1)} = 0_{\text{field}}$. Now using the induction rule, we conclude that $\Sigma 0_{(k-\min\{k,l\}+i)(l-\min\{k,l\}+i)} = 0_{\text{field}}$ holds for $i = \min\{k,l\}$, and therefore $\Sigma 0_{kl} = 0_{\text{field}}$. □

The next theorems show that $I_k$ has the required properties, i.e. it is indeed the identity for matrix multiplication.

**T8** $AI_{\mathtt{c}(A)} = A$ and $I_{\mathtt{r}(A)}A = A$

*Proof.* We just derive $AI_{\mathtt{c}(A)} = A$. First note that $\mathtt{r}(AI_{\mathtt{c}(A)}) = \mathtt{r}(A)$. Now:

$$\begin{aligned}\mathtt{e}(AI_{\mathtt{c}(A)}, i, j) &= \lambda kl\langle \mathtt{c}(A), 1, A_{ik}\rangle \cdot \lambda kl\langle \mathtt{r}(I_{\mathtt{c}(A)}), 1, (I_{\mathtt{c}(A)})_{kj}\rangle \\ &= \Sigma\lambda pq\langle \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(I_{\mathtt{c}(A)})\}, 1, A_{ip}(I_{\mathtt{c}(A)})_{pj}\rangle \\ &= \Sigma\lambda pq\langle \mathtt{c}(A), 1, A_{ip}\text{cond}(p = j, 1, 0)\rangle\end{aligned}$$

Now we show that:

$$A_{ij} = \Sigma\lambda pq\langle \mathtt{c}(A), 1, A_{ip}\text{cond}(p = j, 1, 0)\rangle \tag{3.1}$$

Consider two cases: in the first case [$i = 0$ or $\mathtt{r}(A) < i$ or $j = 0$ or $\mathtt{c}(A) < j$]. Then, $A_{ij} = 0$ and $A_{ip}\text{cond}(p = j, 1, 0) = 0$, and by T7, we have that $\Sigma\lambda pq\langle \mathtt{c}(A), 1, 0\rangle = 0$.

In the second case we assume [$1 \leq i \leq \mathtt{r}(A)$ and $1 \leq j \leq \mathtt{c}(A)$], and we prove equation (3.1) by induction on $\mathtt{c}(A)$:

Basis Case: $\mathtt{c}(A) = 1$. Then, by A30, the RHS of equation (3.1) is given by $A_{i1}\text{cond}(1 = j, 1, 0)$ which is just $A_{ij}$, since if $1 \leq j \leq \mathtt{c}(A) = 1$, then $j = 1$.

Induction Step: $\lambda pq\langle \mathtt{c}(A), 1, A_{ip}\mathrm{cond}(p = j, 1, 0)\rangle$ is a column vector, and its transpose is the row vector given by $\lambda qp\langle 1, \mathtt{c}(A), A_{ip}\mathrm{cond}(p = j, 1, 0)\rangle$. Now, using A31 and A32, we can rewrite equation (3.1) as:

$$A_{ij} = \Sigma\lambda qp\langle 1, \mathtt{c}(A) - 1, A_{ip}\mathrm{cond}(p = j, 1, 0)\rangle + A_{i\mathtt{c}(A)}$$

If $j = \mathtt{c}(A)$ we are done, and if not, we apply the Induction Hypothesis to $A_{ij} = \Sigma\lambda qp\langle 1, \mathtt{c}(A) - 1, A_{ip}\mathrm{cond}(p = j, 1, 0)\rangle$. □

The next four theorems are auxiliary to proving the associativity of matrix multiplication (which is theorem T13 below). The main idea behind the derivation of associativity of matrix multiplication is that we can sum all the entries of a matrix by summing along the rows first, or, by summing along the columns first, and in both cases we obtain the same result.

**T9** $\Sigma(cA) = c\Sigma(A)$

**T10** $\Sigma(A + B) = \Sigma(A) + \Sigma(B)$

In the next theorem we show that we can "fold" a matrix into a column vector, that is, if we take $\Sigma$ of each row, then the $\Sigma$ of the resulting column vector is the same as the $\Sigma$ of the original matrix. Using standard matrix notation this can be expressed as follows:

$$\Sigma \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \Sigma \begin{pmatrix} \Sigma( & a_{11} & \dots & a_{1n} & ) \\ & & \vdots & & \\ \Sigma( & a_{n1} & \dots & a_{nn} & ) \end{pmatrix}$$

and formally, this can be stated as follows:

**T11** $\Sigma A = \Sigma\lambda ij\langle \mathtt{r}(A), 1, \Sigma\lambda kl\langle 1, \mathtt{c}(A), A_{il}\rangle\rangle$

*Proof.* We prove this theorem by induction on the number of rows of $A$, that is by induction on $\mathtt{r}(A)$. In the basis case, $A$ has just one row, so we immediately have $\Sigma(A) = \Sigma((\Sigma(A)))$ by A30. Now the induction step. Suppose the claim holds for $\mathtt{r}(A) < n$. Then:

$$\Sigma \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \Sigma \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix} = a_{11} + \Sigma(R) + \Sigma(S) + \Sigma(M)$$

where we used (2.15). By the induction hypothesis:

$$\Sigma(M) = \Sigma \begin{pmatrix} \Sigma_i m_{1i} \\ \vdots \\ \Sigma_i m_{ni} \end{pmatrix}$$

Using linearity we combine $a_{11}$ and $\Sigma(R)$, and obtain

$$\Sigma \begin{pmatrix} a_{11} & r_1 & \dots & r_n \end{pmatrix} \tag{3.2}$$

and we combine $\Sigma(S)$ and $\Sigma(M)$ to obtain

$$\Sigma \begin{pmatrix} s_1 + \Sigma_i m_{1i} \\ \vdots \\ s_n + \Sigma_i m_{ni} \end{pmatrix} \tag{3.3}$$

and now, we use linearity one more time on (3.2) and (3.3) to obtain

$$\Sigma \begin{pmatrix} a_{11} + \Sigma_i r_i \\ s_1 + \Sigma_i m_{1i} \\ \vdots \\ s_n + \Sigma_i m_{ni} \end{pmatrix} = \begin{pmatrix} \Sigma_i a_{1i} \\ \vdots \\ \Sigma_i a_{ni} \end{pmatrix}$$

which finishes the proof of the theorem. □

Now, the "folding" theorem above (T11), together with T12 below, can express the fact that we can add up all the entries of a matrix by adding them along the rows first, or, along the columns first, and obtain the same result.

**T12** $\Sigma(A) = \Sigma(A^t)$

*Proof.* We prove it by induction on $\mathtt{r}(A)$. The proof is quite easy: By (2.15), $\Sigma(A^t) = a_{11} + \Sigma(S^t) + \Sigma(R^t) + \Sigma(M^t)$. By A32, $\Sigma(S^t) = \Sigma(S)$ and $\Sigma(R^t) = \Sigma(R)$. Finally, by the induction hypothesis, $\Sigma(M^t) = \Sigma(M)$, so indeed $\Sigma(A) = \Sigma(A^t)$. □

We are finally ready to prove associativity of matrix multiplication, but first we introduce some new notation to make the derivation more readable: instead of $\Sigma\lambda ij\langle m, n, t\rangle$ we will write $\sum_{i \le m, j \le n} t$.

**T13** $A(BC) = (AB)C$

*Proof.* First note that: $\mathtt{r}(A(BC)) = \mathtt{r}(A) = \mathtt{r}(AB) = \mathtt{r}((AB)C)$. Now, $\mathtt{e}(A(BC), i, j)$ is given by:

$$\underbrace{\lambda kl\langle \mathtt{c}(A), 1, \mathtt{e}(A, i, k)\rangle}_{\text{i}} \cdot \underbrace{\lambda kl\langle \mathtt{r}(BC), 1, \mathtt{e}(BC, k, j)\rangle}_{\text{ii}}$$

and $\mathtt{max}\{\mathtt{r}(\text{i}), \mathtt{r}(\text{ii})\} = \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(BC)\} = \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}$ and also $\mathtt{max}\{\mathtt{c}(\text{i}), \mathtt{c}(\text{ii})\} = \mathtt{max}\{1, 1\} = 1$. From this we have that $\mathtt{e}(A(BC), i, j)$ is given by:

$$\sum_{p \leq \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}, q \leq 1} A_{ip}(BC)_{pj}$$

and the $(p, j)$-th entry of $BC$ is given by $\sum_{r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}, s \leq 1} B_{pr}C_{rj}$, which, by T12, is equal to:

$$\sum_{s \leq 1, r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}} B_{pr}C_{rj}$$

So putting everything together we have that $\mathtt{e}(A(BC), i, j)$ is given by:

$$\sum_{p \leq \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}, q \leq 1} A_{ip} \left( \sum_{s \leq 1, r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}} B_{pr}C_{rj} \right)$$

and now using T9 we can put $A_{ip}$ inside the second $\Sigma$, and then "unfolding" (T11), we obtain:

$$= \sum_{p \leq \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}, r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}} A_{ip}(B_{pr}C_{rj})$$

and by associativity of multiplication of field elements (A25), and T12, we obtain:

$$= \sum_{r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}, p \leq \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}} (A_{ip}B_{pr})C_{rj}$$

and "folding" back (T11 again), we obtain:

$$= \sum_{r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}, q \leq 1} \sum_{s \leq 1, p \leq \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}} (A_{ip}B_{pr})C_{rj}$$

using T9 and commutativity of field multiplication (A23) we obtain:

$$= \sum_{r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}, q \leq 1} \left( \sum_{s \leq 1, p \leq \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}} A_{ip}B_{pr} \right) C_{rj}$$

$$= \sum_{r \leq \mathtt{max}\{\mathtt{c}(B), \mathtt{r}(C)\}, q \leq 1} (AB)_{ir}C_{rj}$$

Which is just $\mathtt{e}((AB)C, i, j)$, and we are done. □

Finally, we show left and right distributivity, but first we need one more theorem for the defined function $\mathtt{max}$:

**T14** $\mathtt{max}\{i, \mathtt{max}\{j, k\}\} = \mathtt{max}\{\mathtt{max}\{i, j\}, \mathtt{max}\{i, k\}\}$

**T15** $A(B + C) = AB + AC$

*Proof.* First note that $\mathtt{r}(A(B+C)) = \mathtt{r}(A) = \mathtt{max}\{\mathtt{r}(A), \mathtt{r}(A)\}$, and since $\mathtt{r}(A) = \mathtt{r}(AB)$ and $\mathtt{r}(A) = \mathtt{r}(AC)$, we have that this is equal to: $\mathtt{max}\{\mathtt{r}(AB), \mathtt{r}(AC)\} = \mathtt{r}(AB + AC)$. Now, $\mathtt{e}(A(B + C), i, j)$ is given by:

$$\begin{aligned}&\lambda kl\langle \mathtt{c}(A), 1, \mathtt{e}(A, i, k)\rangle \cdot \lambda kl\langle \mathtt{r}(B + C), 1, \mathtt{e}(B + C, k, j)\rangle \\ =&\Sigma\lambda rs\langle \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B + C)\}, 1, A_{ir}(B + C)_{rj}\rangle\end{aligned}$$

Now, using the distributivity of field multiplication (A26), we obtain:

$$=\Sigma\lambda rs\langle \mathtt{max}\{\mathtt{c}(A), \mathtt{max}\{\mathtt{r}(B), \mathtt{r}(C)\}\}, 1, A_{ir}B_{rj} + A_{ir}C_{rj}\rangle$$

we use T14 to show that:

$$\mathtt{max}\{\mathtt{c}(A), \mathtt{max}\{\mathtt{r}(B), \mathtt{r}(C)\}\} = \mathtt{max}\{\mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}, \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(C)\}\}$$

and also T10 to conclude:

$$\begin{aligned}=&\Sigma\lambda rs\langle \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(B)\}, 1, A_{ir}B_{rj}\rangle + \Sigma\lambda rs\langle \mathtt{max}\{\mathtt{c}(A), \mathtt{r}(C)\}, 1, A_{ir}C_{rj}\rangle \\ =&\mathtt{e}(AB, i, j) + \mathtt{e}(AC, i, j)\end{aligned}$$

and we are done. □

**T16** $(B + C)A = BA + CA$

Similar to the derivation of left distributivity given above (T15).

### 3.1.2 Module properties

**T17** $(a + b)A = aA + bA$

**T18** $a(A + B) = aA + aB$

**T19** $(ab)A = a(bB)$

### 3.1.3 Inner product

The following theorems show that our dot product is in fact an inner product:

**T20** $A \cdot B = B \cdot A$

**T21** $A \cdot (B + C) = A \cdot B + A \cdot C$

**T22** $aA \cdot B = a(A \cdot B)$

### 3.1.4 Miscellaneous theorems

**T23** $a(AB) = (aA)B \wedge (aA)B = A(aB)$

**T24** $(AB)^t = B^t A^t$

**T25** $I_k^t = I_k \wedge 0_{kl}^t = 0_{lk}$

**T26** $(A^t)^t = A$

## 3.2 Hard matrix identities

In this section we present four matrix identities which we call *hard matrix identities*. They are hard in the sense that they seem to require computing inverses in their derivations, and therefore appear not to be provable in the theory LA.

$$AB = I, AC = I \rightarrow B = C \qquad \text{I}$$

$$AB = I \rightarrow AC \neq 0, C = 0 \qquad \text{II}$$

$$AB = I \rightarrow BA = I \qquad \text{III}$$

$$AB = I \rightarrow A^t B^t = I \qquad \text{IV}$$

Identity I states that right inverses are unique, identity II states that units are not zero-divisors, and identity III states that a right inverse is an inverse. Identity III was proposed by Cook as a candidate for the separation of Frege and Extended Frege propositional proof systems.

We **conjecture** that the identities I–IV are hard for Frege, however, it might be easier to prove a weaker statement: the identities I–IV are independent of LA.

It is interesting to note that matrix products cannot be expressed feasibly in bounded-depth Frege (directly, without extension variables). This is essentially because the parity function is hard for bounded-depth circuits[1], and computing the $(i,j)$-th entry of $AB$ over $\mathbb{Z}_2$ is the same as computing $\texttt{PARITY}(a_{i1}b_{1j},\ldots,a_{in}b_{nj})$.

It is an open problem whether these identities can be proven in poly-bounded Frege or even poly-bounded $\mathrm{NC}^i$-Frege, for any $i$. In Section 8.3 we show that hard matrix identities can be proven in poly-bounded P/poly-Frege (i.e., in poly-bounded Extended Frege).

It turns out that it is enough to show that one of these identities (we always choose $AB = I \to BA = I$) can be proven in poly-bounded Extended Frege, to conclude that all four can be proven in poly-bounded Extended Frege. The reason is that their equivalence can be shown with poly-bounded Frege proofs (in fact, as Theorem 3.2.1 below shows, they can be proven equivalent in LA).

**Theorem 3.2.1** LA proves the equivalence I $\Leftrightarrow$ II $\Leftrightarrow$ III $\Leftrightarrow$ IV.

*Proof.* We show that I $\Rightarrow$ II $\Rightarrow$ III $\Rightarrow$ IV $\Rightarrow$ I.
**I $\Rightarrow$ II** Assume $AB = I \wedge AC = 0$. By A4, $AB + AC = I + 0$, and by T1 and T15, $A(B + C) = I$. Using I, $B = B + C$, so by T2, $C = 0$.
**II $\Rightarrow$ III** Assume $AB = I$. By A1 and A4, $(AB)A = IA$, by T2, $(AB)A + (-1)IA = 0$, by T13 and T23, $A(BA) + A(-1)I = 0$, and by T15, $A(BA + (-1)I) = 0$. By II, $BA + (-1)I = 0$, and by T2, $BA = I$.
**III $\Rightarrow$ IV** Assume $AB = I$. By III, $BA = I$, and by A29 and Eq, $(BA)^t = I^t$. By T24, we obtain $A^tB^t = I$.
**IV $\Rightarrow$ I** Assume $AB = I \wedge AC = I$. By T2 $AB+(-1)AC = 0$, by T23, $AB+A(-1)C = 0$, by T15, $A(B + (-1)C) = 0$, by T13, $(BA)(B + (-1)C) = 0$. Now, using transpose property T24, we get $(B + (-1)C)^t(BA)^t = 0$, and since $AB = I$, by IV, $A^tB^t = I$, so by T24 again, $(BA)^t = I$, so we obtain that $(B + (-1)C)^t = 0$, so $B + (-1)C = 0$, so $B = C$. □

Consider now the following identity due to C. Rackoff:

$$\lambda ij\langle 1, \mathsf{c}(B), B_{ij}\rangle = 0_{1\mathsf{c}(B)} \to AB \neq I_{\mathtt{r}(A)} \qquad \text{V}$$

[1]See [SP95, Chapter 11] for a good presentation of the lower bound for the parity function due to Furst, Saxe and Sipser. The original is in [FSS84, pp. 13–27].

This identity states that if the top row of a matrix is zero, then the matrix cannot have a left inverse, that is:

$$A \left( \begin{array}{cccc} 0 & 0 & \dots & 0 \\ \hline & & & \\ & & & \end{array} \right) \neq I$$

Using the rank function (where $\mathtt{rank}(A)$ is the largest number of linearly independent rows/columns of $A$), identity V can be proven as follows:

$$\mathtt{rank}(AB) = \min\{\mathtt{rank}(A), \mathtt{rank}(B)\}$$

Since $B$ has the top row of zeros, $\mathtt{rank}(B) < \mathtt{c}(B)$, so that $\mathtt{rank}(AB) < \mathtt{c}(B)$. But $\mathtt{c}(B) = \mathtt{r}(A)$, so $\mathtt{rank}(AB) < \mathtt{rank}(I)$, where $I$ is the $\mathtt{r}(A) \times \mathtt{r}(A)$ identity matrix.

**Lemma 3.2.1** LA proves that III implies V.

*Proof.* Suppose the top row of $B$ is zero. Then the top row of $BA$ is zero. If $AB = I$, then by III, $BA = I$, so $AB \neq I$. □

It is an open question whether III follows from V in LA, and it is also an open question whether LA can prove V, which, somehow, seems to be a "weaker" identity than the four identities above. Interestingly, it can be shown in LA that the Odd Town Theorem follows from V.

The Odd Town Theorem[2] states the following: Suppose a town has $n$ citizens, and that there is a set of clubs, each consisting of citizens, such that each club has an odd number of members, and such that every two clubs have an even number of members in common. Then there is no more than $n$ clubs.

**Lemma 3.2.2** LA proves that V implies the "Odd Town Theorem".

*Proof.* We want the underlying field to be $\mathbb{Z}_2$, so we need the condition that $a = 0 \vee a = 1$. Let $A$ be the incidence matrix for the Odd Town problem, defined as follows: $\mathtt{r}(A)$ is the number of clubs in Odd Town, and $\mathtt{c}(A)$ is the number of citizen in Odd Town, and, if the assumption is true (i.e. each club has an odd number of members, and every two clubs have an even number of members in common), then the $(i, j)$-th entry of $AA^t$ is $\delta_{ij}$, so that $AA^t = I_{\mathtt{r}(A)}$.

[2]See [BF92, page 9] for the "Odd Town Theorem" and many related combinatorial principles. Also see [BBP94, page 5] for a discussion of hard combinatorial candidates for Frege from examples based on Linear Programming—the authors mention the "Odd Town Theorem".

Suppose that $\mathtt{r}(A) > \mathtt{c}(A)$. Then we can pad $A$ with $\mathtt{r}(A) - \mathtt{c}(A)$ columns of zeros, and call the result $A'$. Then the first $\mathtt{r}(A) - \mathtt{c}(A)$ rows of $(A')^t$ consist of zeros. However the $(i, j)$ entry of $(A')(A')^t$ is the same as the $(i, j)$ entry of $AA^t$, i.e. $(A')(A')^t = I_{\mathtt{r}(A)}$, which according to V is a contradiction. This finishes the proof. $\square$

# Chapter 4

# LA with Matrix Powering

In this chapter we expand LA by adding to it a new function, P, for computing powers of matrices. We call the new theory LAP, and we give its precise definition in section 4.1.

Expressing powers of matrices allows us to define Berkowitz's algorithm in the new theory. Berkowitz's algorithm, which we present in section 4.2, computes the coefficients of the characteristic polynomial of a matrix $A$ via iterated matrix product. That is, Berkowitz's algorithm computes the coefficients of the polynomial $p_A(x) = \det(xI - A)$. From $p_A(x)$ we can immediately obtain the adjoint of $A$, $\mathrm{adj}(A)$, and the determinant of $A$, $\det(A)$. Therefore, Berkowitz's algorithm allows us to compute and argue about inverses.

Berkowitz's algorithm is the fastest known algorithm[1] for computing inverses and determinants (it is an $\mathrm{NC}^2$ algorithm, while, for example, Gaussian Elimination is a sequential polytime algorithm), and it yields itself to a natural and simple formalization in our theory.

## 4.1 The theory LAP

### 4.1.1 Language

We have the same language as for LA, except for the new function symbol P, which is a 2-ary function where the first argument is of type index, and the second argument is of

[1] There are two other parallel algorithms for computing the coefficients of the char polynomial of a matrix: Chistov's algorithm and Csanky's algorithm. Chistov's algorithm is more difficult to formalize, and Csanky's algorithm works only for fields of characteristic 0; see [vzG93, section 13.4] for all the details.

type matrix. The intended meaning of $\mathtt{P}(m, A)$ is $A^m$. We denote this new language by $\mathcal{L}_{\mathrm{LAP}}$.

### 4.1.2 Terms and formulas

We expand the definition of terms and formulas in LA given in section 2.2.1. The basis case remains the same. In the induction step we add the following case:

10. If $m$ is a term of type index, and $T$ is a term of type matrix, then $\mathtt{P}(m, T)$ is a term of type matrix.

### 4.1.3 Axioms

We have the same axioms as in LA, that is A1–33, but we also add two new axioms that define the behavior of P: A34 and A35, stated below.

In the definition of P we consider two cases: first we assume that $m = 0$, in which case we want $\mathtt{P}(m, T)$ to be the identity (axiom A34). In the second case, we assume that $m > 0$, and compute $\mathtt{P}(m, T)$ recursively (axiom A35).

$$\textbf{A34} \quad m = 0 \rightarrow \mathtt{P}(m, A) = I_{\mathtt{r}(A)}$$

$$\textbf{A35} \quad \rightarrow \mathtt{P}(m{+}1, A) = \mathtt{P}(m, A){*}A$$

Table 4.1: Axioms for P

We now define PK-LAP analogously to PK-LA. Since both definitions are so similar, some explanatory details are omitted this time; see section 2.4 for more explanations.

**Definition 4.1.1** We define the proof system PK-LAP to be a system of sequent calculus proofs, where all the initial sequents are either of the form $\alpha \rightarrow \alpha$ (for a formula $\alpha$ over $\mathcal{L}_{\mathrm{LAP}}$), or are given by one of the axiom schemas A1–35, and all the other sequents (if any) follow from their predecessor(s) in the tree by one of the rules for propositional consequence, or by Ind, or by Eq.

**Definition 4.1.2** The theory LAP is the set of sequents over $\mathcal{L}_{\mathrm{LAP}}$ which have PK-LAP derivations.

**Lemma 4.1.1** LA $\subseteq$ LAP

*Proof.* Immediate from the definitions of PK-LA and PK-LAP. □

## 4.2 Berkowitz's algorithm

For a given matrix $A$, Berkowitz's algorithm computes the coefficients of the characteristic polynomial, $p_A(x)$, of $A$. In the context of the theory LAP, given an $n \times n$ matrix $A$, $p_A$ is an $(n+1) \times 1$ column vector containing the coefficients of the char poly of $A$, that is $p_A$ is $\begin{pmatrix} p_n & p_{n-1} & \dots & p_0 \end{pmatrix}^t$. The $p_i$'s are the coefficients of the $n$-th degree polynomial given by $\det(xI - A)$, but we will not prove this in our theories; we will prove the properties of the char poly directly from the definition given by Berkowitz's algorithm (See definitions 4.2.2 and 4.2.3). We will also denote the coefficients of the char poly of a matrix $A$ by $(p_A)_i$, to avoid ambiguities.

The theory LAP has three types: indices, field elements, and matrices. Thus, it is not possible to write a polynomial $p(x)$ with an indeterminate $x$. So, we denote polynomials by matrix variables, where the correspondence is the following:

$$\text{polynomial } p(x) = p_n x^n + p_{n-1}x^{n-1} + \cdots + p_0 \quad \longleftrightarrow \quad \text{column vector } p = \begin{pmatrix} p_n \\ p_{n-1} \\ \vdots \\ p_0 \end{pmatrix}$$

We can evaluate the polynomial $p$ at field element or a matrix, using the dot product (see (2.10) on page 19, for a definition of the dot product). So, $p(a)$ is given by:

$$\Sigma(p \cdot \lambda ij\langle n+1, 1, a^{n+1-i}\rangle)$$

(see (4.11), page 54, for the definition of $a^{n+1-i}$), and $p(A)$ (where $A$ is assumed to be $n \times n$, but this is not a crucial assumption) is given by:

$$\lambda ij\langle n, n, \mathsf{e}(p \cdot \lambda kl\langle n, 1, \mathsf{e}(\mathsf{P}(n-k, A), i, j)\rangle, i, j\rangle$$

The usual properties of polynomials, for example $(p+q)(a) = p(a) + q(a)$ or $(c \cdot p)(a) = c \cdot (p(a))$, are easy to prove in LAP.

Berkowitz's algorithm is based on Samuelson's identity, and we present the construction of Berkowitz's algorithm from Samuelson's identity (and Lemma 4.2.2 due to Paul Beame) in the next section. This construction relies on the cofactor expansion, and the

definition of the char poly given by $p_A(x) = \det(xI - A)$. This construction provides a proof of correctness, which unfortunately is infeasible; it is infeasible because it relies on an infeasible proof of the cofactor expansion of the determinant, and on an infeasible proof of the Cayley-Hamilton Theorem (in order to prove Lemma 4.2.2).

### 4.2.1 Samuelson's identity

We follow Berkowitz's paper ([Ber84]), but we make some modifications (for example, we define the char poly to be $\det(xI - A)$ rather than $\det(A - xI)$). The main idea behind Berkowitz's algorithm is Samuelson's identity, which relates the char polynomial of a matrix to the char polynomial of its principal submatrix. Thus, the coefficients of the char polynomial of an $n \times n$ matrix $A$ below are computed in terms of the coefficients of the char polynomial of $M$:

$$A = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

where $R, S$ and $M$ are $1 \times (n-1)$, $(n-1) \times 1$ and $(n-1) \times (n-1)$ submatrices, respectively.

**Lemma 4.2.1 (Samuelson's Identity)** Let $p(x)$ and $q(x)$ be the char polynomials of $A$ and $M$, respectively. Then:

$$p(x) = (x - a_{11})q(x) - R * \text{adj}(xI - M) * S$$

Recall that the adjoint of a matrix $A$ is the transpose of the matrix of cofactors of $A$; that is, the $(i, j)$-the entry of $\text{adj}(A)$ is given by $(-1)^{i+j} \det(A[j|i])$. Also recall that $A[k|l]$ is the matrix obtained from $A$ by deleting the $k$-th row and the $l$-th column. We also make up the following notation: $A[-|l]$ denotes that only the $l$-th column has been deleted. Similarly, $A[k|-]$ denotes that only the $k$-th row has been deleted, and $A[-|-] = A$.

*Proof.*

$$\begin{aligned} p(x) &= \det(xI - A) \\ &= \det \begin{pmatrix} x - a_{11} & -R \\ -S & xI - M \end{pmatrix} \end{aligned}$$

using the cofactor expansion along the first row:

$$= (x - a_{11}) \det(xI - M) + \sum_{j=1}^{n-1} (-1)^j (-r_j) \det(\underbrace{-S(xI - M)[-|j]}_{(*)})$$

where $R = (r_1 r_2 \ldots r_{n-1})$, and the matrix indicated by $(*)$ is given as follows: the first column is $S$, and the remaining columns are given by $(xI - M)$ with the $j$-th column deleted. We expand $\det(-S(xI - M)[-|j])$ along the first column, i.e., along the column $S = (s_1 s_2 \ldots s_{n-1})^T$:

$$= (x - a_{11})q(x) + \sum_{j=1}^{n-1}(-1)^j(-r_j)\sum_{i=1}^{n-1}(-1)^{i+1}(-s_i)\det(xI - M)[i|j]$$

and rearranging:

$$= (x - a_{11})q(x) - \sum_{i=1}^{n-1}\left(\sum_{j=1}^{n-1} r_j(-1)^{i+j}\det(xI - M)[i|j]\right)s_i$$
$$= (x - a_{11})q(x) - R * \mathrm{adj}(xI - M) * S$$

and we are done. □

**Lemma 4.2.2** Let $q(x) = q_{n-1}x^{n-1} + \cdots + q_1 x + q_0$ be the char polynomial of $M$, and let:

$$B(x) = \sum_{k=2}^{n}(q_{n-1}M^{k-2} + \cdots + q_{n-k+1}I)x^{n-k} \tag{4.1}$$

Then $B(x) = \mathrm{adj}(xI - M)$.

**Example 4.2.1** If $n = 4$, then

$$B(x) = Iq_3x^2 + (Mq_3 + Iq_2)x + (M^2q_3 + Mq_2 + Iq_1)$$

*Proof.* First note that:

$$\mathrm{adj}(xI - M) * (xI - M) = \det(xI - M)I = q(x)I$$

Now multiply $B(x)$ by $(xI - M)$, and using the Cayley-Hamilton Theorem, we can conclude that $B(x) * (xI - M) = q(x)I$. Thus, the result follows as $q(x)$ is not the zero polynomial; i.e., $(xI - M)$ is *not* singular. □

From Lemma 4.2.1 and Lemma 4.2.2 we have that:

$$p(x) = (x - a_{11})q(x) - R * B(x) * S \tag{4.2}$$

### 4.2.2 Expressing the char poly as a product of matrices

Using (4.2), we can express the char poly of a matrix as iterated matrix product. Again, suppose that $A$ is of the form:

$$\begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

**Definition 4.2.1** We say that an $n \times m$ matrix is *Toeplitz* if the values on each diagonal are the same. We say that a matrix is *upper triangular* if all the values below the main diagonal are zero. A matrix is *lower triangular* if all the values above the main diagonal are zero.

If we express equation (4.2) in matrix form we obtain:

$$p = C_1 q \tag{4.3}$$

where $C_1$ is an $(n+1) \times n$ Toeplitz lower triangular matrix, and where the entries in the first column are defined as follows:

$$c_{i1} = \begin{cases} 1 & \text{if } i = 1 \\ -a_{11} & \text{if } i = 2 \\ -(RM^{i-3}S) & \text{if } i \geq 3 \end{cases} \tag{4.4}$$

**Example 4.2.2** If $A$ is a $4 \times 4$ matrix, then $p = C_1 q$ is given by:

$$\begin{pmatrix} p_4 \\ p_3 \\ p_2 \\ p_1 \\ p_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -a_{11} & 1 & 0 & 0 \\ -RS & -a_{11} & 1 & 0 \\ -RMS & -RS & -a_{11} & 1 \\ -RM^2S & -RMS & -RS & -a_{11} \end{pmatrix} \begin{pmatrix} q_3 \\ q_2 \\ q_1 \\ q_0 \end{pmatrix}$$

Berkowitz's algorithm consists in repeating this for $q$, and so on, and eventually expressing $p$ as a product of matrices:

$$p = C_1 C_2 \cdots C_n$$

**Definition 4.2.2 (Berkowitz's algorithm)** Let $A$ be an $n \times n$ matrix. *Berkowitz's algorithms* computes an $(n+1) \times 1$ column vector $p_A$ as follows:

Let $C_j$ be an $(n+2-j) \times (n+1-j)$ Toeplitz and lower-triangular matrix, where the entries in the first column are define as follows:

$$\begin{cases} 1 & \text{if } i = 1 \\ -a_{jj} & \text{if } i = 2 \\ -R_j M_j^{i-3} S_j & \text{if } 3 \leq i \leq n+2-j \end{cases} \tag{4.5}$$

where $M_j$ is the $j$-th principal submatrix, so $M_1 = A[1|1]$, $M_2 = M_1[1|1]$, and in general $M_{j+1} = M_j[1|1]$, and $R_j$ and $S_j$ are given by:

$$\begin{pmatrix} a_{j(j+1)} & a_{j(j+2)} & \dots & a_{jn} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} a_{(j+1)j} & a_{(j+2)j} & \dots & a_{nj} \end{pmatrix}^t$$

respectively (see Figure 4.1). Then $p_A = C_1 C_2 \cdots C_n$.

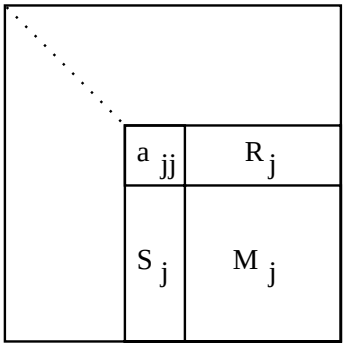



Figure 4.1: $a_{jj}, R_j, S_j, M_j$

Note that Berkowitz's algorithm is field independent (there are no divisions in the computation of $p_A$), and therefore, since Berkowitz's algorithm is a cornerstone of our theory of Linear Algebra, all our results are field independent.

The following definitions (characteristic polynomial, adjoint, and determinant), are definitions in terms of Berkowitz's algorithm. We will show later that the adjoint and the determinant, defined from Berkowitz's algorithm, correspond to the usual definitions of the adjoint and the determinant. Corollary 6.3.1 states that LAP proves, from the cofactor expansion, that the usual definition of the adjoint (as a matrix of cofactors) corresponds to the definition of the adjoint from Berkowitz's algorithm. Since the cofactor expansion formula follows from the Cayley-Hamilton Theorem (see Chapter 6 for all these results), and we give a feasible proof of the Cayley-Hamilton Theorem (Chapter 8), we also have a feasible proof of the fact that the usual definition of the adjoint corresponds to the definition in terms of Berkowitz's algorithm. Same comments apply to the determinant.

**Definition 4.2.3 (Characteristic polynomial)** We want to define the characteristic polynomial (char poly) in terms of Berkowitz's algorithm. To be precise, we define the *coefficients of the char poly*, for a given matrix $A$, to be the output of Berkowitz's algorithm, i.e., to be the entries of the column vector $p_A = C_1 C_2 \cdots C_n$. We define the *char poly*, for a given matrix $A$, to be the polynomial whose coefficients are the output of Berkowitz's algorithm. In practice we do not make this distinction, and when we say char poly, we mean both the column vector of coefficients given by Berkowitz's algorithm, and the polynomial $p_A(x)$ with these coefficients.

This definition of the char polynomial corresponds to the *true* char polynomial in the following sense: the output of Berkowitz's algorithm is a column vector $p$, given by:

$$\begin{pmatrix} p_n & p_{n-1} & \dots & p_0 \end{pmatrix}^t$$

where $p_i$ is the coefficient of $x^i$ in $\det(xI - A)$.

**Definition 4.2.4 (Adjoint)** Let $p$ be the char poly of $A$. Then the *adjoint* of $A$, denoted $\mathrm{adj}(A)$, is defined as follows:

$$\mathrm{adj}(A) := (-1)^{n-1}(p_n A^{n-1} + p_{n-2} A^{n-3} + \ldots + p_1 I) \tag{4.6}$$

Note that this definition of the adjoint is equivalent to the usual definition of the adjoint in terms of determinants of minors; see Lemma 6.3.2, where we show that LAP proves (from the C-H Theorem) that our adjoint is equal to the adjoint given by the transpose of the matrix of cofactors.

**Definition 4.2.5 (Determinant)** Let $p$ be the char poly of $A$. Then the *determinant* of $A$, denoted $\det(A)$, is defined as follows:

$$\det(A) := (-1)^n p_0 \tag{4.7}$$

This definition of the determinant is equivalent to the usual definition given in terms of the cofactor expansion formula; see Section 6.1 where we show that our definition of the determinant satisfies the axiomatic definition of the determinant, and hence the det function computed by Berkowitz's algorithm is the *true* det function.

When proving results by induction on the size of matrices, we will often use the following identity:

$$\det(A) = a_{11} \det(M) - R\,\mathrm{adj}(M) S \tag{4.8}$$

We prove this identity in Lemma 5.1.3 (in fact, we show that LAP can prove this identity). This identity is just Samuelson's Identity (Lemma 4.2.1) with $x$ replaced by zero, however, Samuelson's Identity uses the "traditional" definition of the adjoint in terms of cofactors, and equation (4.8) uses our definition of the adjoint (as in Definition 4.2.4). At this point we do not have a feasible proof of Samuelson's Identity (we will have it when we prove the Cayley-Hamilton feasibly in Chapter 8), so at this point we cannot conclude (feasibly) equation (4.8) by letting $x = 0$ is Samuelson's identity.

### 4.2.3 Expressing the char poly in LAP

The point of introducing the new symbol $\mathtt{P}$ into LA is that we can now express iterated matrix products. Let $A_1, A_2, \ldots, A_m$, be a sequence of square matrices of equal size (if they are not of equal size they can be padded with zeros, and the actual product can be extracted from the product of the padded matrices at the end). To compute the iterated matrix product $A_1 A_2 \cdots A_m$, we place these matrices into a single big matrix $C$, above the main diagonal of $C$. More precisely, assume that the $A_i$'s are $n \times n$ matrices. Then, $C$ is a $(m+1)n \times (m+1)n$ matrix of the form:

$$\begin{pmatrix} 0 & A_1 & 0 & \cdots & 0 \\ 0 & 0 & A_2 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \cdots & A_m \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Now, compute $C^m$. The product $A_1 A_2 \ldots A_m$ is the $n \times n$ upper-right corner of $C^m$.

Given a matrix $A$, we compute its char poly $p_A$ as follows:

$$p_A := \lambda ij \langle n+1, 1, \mathtt{e}(\mathtt{P}(n, \mathtt{D}(A)), i, n(n+1)) \rangle \tag{4.9}$$

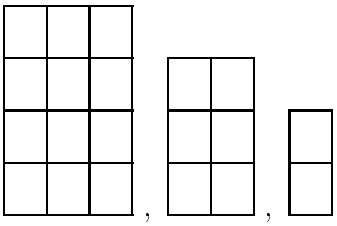
where $n := \max\{\mathtt{r}(A), \mathtt{c}(A)\}$ (so in effect, if $A$ is not a square matrix we compute the char poly of its padded version), and where $\mathtt{D}(A)$ is the following matrix:

$$\mathtt{D}(A) := \begin{pmatrix} 0 & \mathtt{C}(1,A) & 0 & \cdots & 0 \\ 0 & 0 & \mathtt{C}(2,A) & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \mathtt{C}(n,A) \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

that is, $\mathtt{D}$ consists of $(n+1) \times (n+1)$ blocks of size $(n+1) \times (n+1)$ each, and all these blocks, except those which are above the main diagonal, are zero. Therefore, when we raise this matrix to the $n$-th power, we obtain the product of the $\mathtt{C}(k, A)$'s in the upper-right corner; hence $\mathtt{P}(n, \mathtt{D}(A))$ in (4.9).

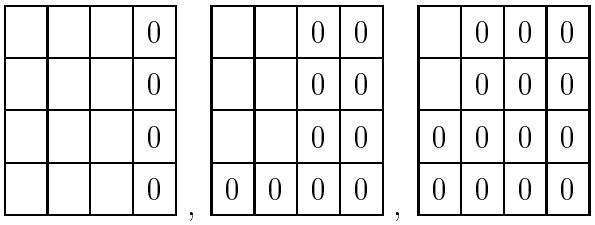
In the definition of Berkowitz's algorithm, we see that the matrices $C_1, \ldots, C_n$ are of different sizes.

**Example 4.2.3** If $n = 3$ then $C_1, C_2, C_3$ look as follows:



We want the $\mathtt{C}(k, A)$'s to be square matrices of the same size to be able to define $\mathtt{D}(A)$, so we pad them with zeros to convert them to $(n+1) \times (n+1)$ matrices.

**Example 4.2.4** After the padding, the matrices from example 4.2.3 look as follows:



Formally, $\mathtt{C}(k, A) := \lambda ij\langle n+1, n+1,$

$$\mathrm{cond}(i \leq k+1 \vee j \leq k, \left\{ \begin{array}{ll} 0 & i < j \\ 1 & i = j \\ -A_{kk} & i = j+1 \\ \mathtt{e}(-\mathtt{R}(A,k) * \mathtt{P}(\mathtt{M}(A,k), i-3) * \mathtt{S}(A,k), 1, 1) & j+2 \leq i \end{array} \right\}, 0\rangle$$

Note that the expression between "{" and "}" can be given formally with four nested conditionals, and the defined matrix terms $\mathtt{R}$, $\mathtt{M}$, and $\mathtt{S}$, are given as follows:

$$\begin{aligned} \mathtt{R}(A,k) &:= \lambda ij\langle n-k, 1, \mathrm{cond}(i=1, \mathtt{e}(A, k, k+j), 0)\rangle \\ \mathtt{S}(A,k) &:= \lambda ij\langle n-k, 1, \mathrm{cond}(j=1, \mathtt{e}(A, k+i, k), 0)\rangle \\ \mathtt{M}(A,k) &:= \lambda ij\langle n-k, n-k, \mathtt{e}(A, k+i, k+j)\rangle \end{aligned} \tag{4.10}$$

In (2.12) we already defined $\mathtt{R}, \mathtt{S}, \mathtt{M}$, but for the case where $k = 1$. The new definitions in (4.10) extend $\mathtt{R}, \mathtt{S}, \mathtt{M}$ to all the values of $k$.

**Example 4.2.5** Suppose that

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Then:

$$\mathtt{R}(A,1) := \lambda ij\langle 2, 1, \mathrm{cond}(i = 1, \mathtt{e}(A, 1, 1 + j), 0)\rangle = \begin{pmatrix} a_{12} & a_{13} \end{pmatrix}$$

$$\mathtt{S}(A,1) := \lambda ij\langle 2, 1, \mathrm{cond}(j = 1, \mathtt{e}(A, 1 + i, 1), 0)\rangle = \begin{pmatrix} a_{21} \\ a_{31} \end{pmatrix}$$

$$\mathtt{M}(A,1) := \lambda ij\langle 2, 2, \mathtt{e}(A, 1 + i, 1 + j)\rangle = \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix}$$

We define $\mathtt{D}(A)$ to be a $n(n+1) \times n(n+1)$ matrix where the $(i,j)$-th entry is given by:

$$\mathrm{cond}(\mathtt{div}(i, n+1) + 1 = \mathtt{div}(j, n+1), \mathtt{e}(\mathtt{C}(\mathtt{div}(i, n+1), A), \mathtt{rem}(i, n), \mathtt{rem}(j, n)), 0)$$

We used the quotient function $\mathtt{div}$ and the reminder function $\mathtt{rem}$ to compute the entries of $\mathtt{D}(A)$. Recall that $\mathtt{D}(A)$ consists of $(n+1) \times (n+1)$ blocks, each block of size $(n+1) \times (n+1)$, and that only the blocks above the blocks on the main diagonal are possibly non-zero. This means that the $(i,j)$-th entry of $\mathtt{D}(A)$ is zero unless

$$i = (n+1) * q_1 + r_1 \qquad 0 \leq r_1 < n+1$$
$$j = (n+1) * q_2 + r_2 \qquad 0 \leq r_2 < n+1$$

and $q_1 + 1 = q_2$ (which ensures that we are in the $q_1$-th block above the blocks on the main diagonal), and in that block we are considering the $(r_1, r_2)$-th entry.

### 4.2.4 Expressing adj and det in LAP

We can define the adjoint and the determinant in LAP. First we define $t^n$ for a general field term $t$ and a general index term $n$ as follows:

$$t^n := \mathtt{e}(\mathtt{P}(n, \lambda ij\langle 1, 1, t\rangle), 1, 1) \tag{4.11}$$

The idea is that $t^n$ is the (only) entry of the $n$-th power of the matrix $(t)$, i.e. $(t)^n = (t^n)$. Now note that the $(i,j)$-th entry of $\mathrm{adj}(A)$ is $(-1)^n$ times the dot product of:

$$\begin{pmatrix} p_n \\ p_{n-1} \\ \vdots \\ p_1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \mathtt{e}(A^{n-1}, i, j) \\ \mathtt{e}(A^{n-2}, i, j) \\ \vdots \\ \mathtt{e}(I_n, i, j) \end{pmatrix} \tag{4.12}$$

where the matrix on the left is just $p_A$ (without the last entry, $p_0$), so the adjoint is given by:

$$\mathtt{adj}(A) := (-1)^{n-1} \lambda ij \langle n, n, \mathtt{e}(p_A \cdot \lambda kl \langle n, 1, \mathtt{e}(A^{n-k}, i, j)\rangle, i, j)\rangle$$

and the determinant is simply given by:

$$\mathtt{det}(A) := (-1)^n \mathtt{e}(p_A, n+1, 1)$$

where $\mathtt{e}(p_A, n+1, 1)$ is $p_0$, i.e. the constant coefficient of the char poly of the matrix $A$.

## 4.3 Berkowitz's algorithm and clow sequences

Clow sequences provide a simple way of understanding the computations in Berkowitz's algorithm. It turns out (and this is an observation due to Valiant, see [Val92, Section 3]) that Berkowitz's algorithm computes sums of restricted clow sequences. Clow sequences are easy to define (they are just generalized permutations), and it is not difficult to see how Berkowitz's algorithm computes sums of clow sequences.

Besides giving us insight into Berkowitz's algorithm, clow sequences are a potential tool for proving the Cayley-Hamilton Theorem directly in LAP (thus far, we only have a polytime proof of the C-H Theorem, given in Chapter 8). This is especially interesting in light of a dynamic programming algorithm for computing clows, given in [MV97, Table 1]. If we could somehow prove the correctness of this algorithm in LAP, we could use it to prove the C-H Theorem in LAP. So far this is only speculation, but the point is that maybe a "clow sequences approach" to the determinant could prove the C-H Theorem in $\mathrm{NC}^2$, rather than in polytime.

A substantial part of the material in the rest of this section comes from [MV97] where the authors build upon a purely combinatorial interpretation of the Cayley-Hamilton Theorem given in [Str83]. Unfortunately, the combinatorial proof of the C-H Theorem given in [Str83] is infeasible.

Recall the Lagrange expansion for the determinant:

$$\det(A) = \sum_{\sigma \in S_n} \text{sign}(\sigma) a_{1\sigma(1)} \cdots a_{n\sigma(n)}$$

The summation is over all permutations on $n$ elements. The *sign of a permutation* $\sigma$, $\text{sign}(\sigma)$, is defined as follows:

$$\text{sign}(\sigma) = (-1)^{\text{number of transpositions in } \sigma}$$

To move to a combinatorial setting, we interpret $\sigma \in S_n$ as a directed graph $G_\sigma$ on $n$ vertices.

**Example 4.3.1** The permutation given by:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 1 & 2 & 4 & 6 & 5 \end{pmatrix}$$

corresponds to the directed graph $G_\sigma$ given by Figure 4.2 below.

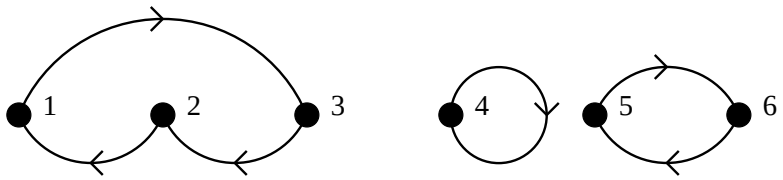



Figure 4.2: $G_\sigma$

Given a matrix $A$, define the weight of $G_\sigma$, $w(G_\sigma)$, as the product of $a_{ij}$'s such that $(i,j) \in G_\sigma$. Consider $G_\sigma$ given by Figure 4.2: $w(G_\sigma) = a_{13}a_{32}a_{21}a_{44}a_{56}a_{65}$. Thus, using this new terminology:

$$\det(A) = \sum_{\sigma \in S_n} \text{sign}(\sigma) w(G_\sigma)$$

The Lagrange expression cannot be converted directly into an efficient algorithm for the determinant, because the summation is over $n!$ monomials.

Any efficient algorithm should implicitly count over all monomials; the bottleneck in doing so directly is that permutations are not easily "factorizable" to allow for a simple implementation. We will get around this problem by enlarging the summation from cycle covers to clow sequences.

**Definition 4.3.1** A *clow* is a walk $(w_1, \ldots, w_l)$ starting from vertex $w_1$ and ending at the same vertex, where any $(w_i, w_{i+1})$ is an edge in the graph. Vertex $w_1$ is the least-numbered vertex in the clow, and it is called the *head* of the clow. We also require that the head occur only once in the clow. This means that there is exactly one incoming edge $(w_l, w_1)$, and one outgoing edge $(w_1, w_2)$ at $w_1$, and $w_i \neq w_1$ for $i \neq 1$. The *length* of a clow $(w_1, \ldots, w_l)$ is $l$.

**Example 4.3.2** Consider the clow $C$ given by $(1, 2, 3, 2, 3)$ on four vertices. The head of clow $C$ is vertex 1, and the length of $C$ is 6. See Figure 4.3.

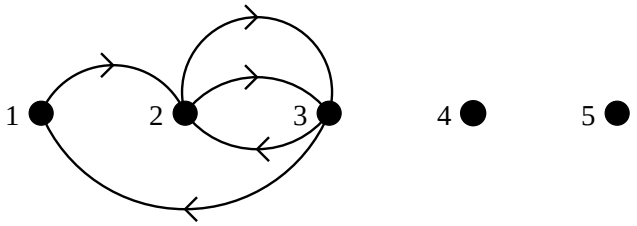



Figure 4.3: Clow $C$

**Definition 4.3.2** A *clow sequence* is a sequence of clows $(C_1, \ldots, C_k)$ with the following two properties: (i) The sequence is ordered by the heads: $\text{head}(C_1) < \ldots < \text{head}(C_k)$. (ii) The total number of edges, counted with multiplicity, adds to $n$; that is, the lengths of the clows add up to $n$.

Note that a cycle cover is a special type of a clow sequence.

We will now show how to associate a sign with a clow sequence that is consistent with the definition of the sign of a cycle cover. The sign of a cycle cover can be shown to be $(-1)^{n+k}$, where $n$ is the number of vertices in the graph, and $k$ is the number of components in the cycle cover.

**Definition 4.3.3** We define the sign of a clow sequence to be $(-1)^{n+k}$ where $n$ is the number of vertices in the graph, and $k$ is the number of clows in the sequence.

**Example 4.3.3** We list the clow sequences associated with the three vertices $\{1, 2, 3\}$.

We give the sign of the corresponding clow sequences in the right-most column:

| | | |
|---|---|---|
| 1. | $(1),(2),(3)$ | $(-1)^{3+3}=1$ |
| 2. | $(1,2),(3)$ | $(-1)^{3+2}=-1$ |
| 3. | $(1,2,2)$ | $(-1)^{3+1}=1$ |
| 4. | $(1,2),(2)$ | $(-1)^{3+2}=-1$ |
| 5. | $(1),(2,3)$ | $(-1)^{3+2}=-1$ |
| 6. | $(1,2,3)$ | $(-1)^{3+1}=1$ |
| 7. | $(1,3,3)$ | $(-1)^{3+1}=1$ |
| 8. | $(1,3),(3)$ | $(-1)^{3+2}=-1$ |
| 9. | $(1,3,2)$ | $(-1)^{3+1}=1$ |
| 10. | $(1,3),(2)$ | $(-1)^{3+2}=-1$ |
| 11. | $(2,3,3)$ | $(-1)^{3+1}=1$ |
| 12. | $(2,3),(3)$ | $(-1)^{3+2}=-1$ |

Notice that the number of permutations on 3 vertices is $3!=6$, and indeed, the clow sequences $\{3,4,7,8,11,12\}$ do *not* correspond to cycle covers. Notice that we listed these clow sequences which do not correspond to cycle covers by pairs: $\{3,4\},\{7,8\},\{11,12\}$. Consider the first pair: $\{3,4\}$. We will later define the weight of a clow (simply the product of the labels of the edges), but notice that clow sequence 3 corresponds to $a_{12}a_{22}a_{21}$ and clow sequence 4 corresponds to $a_{12}a_{21}a_{22}$, which is the same value; however, they have opposite signs, so they cancel each other out. Same for pairs $\{7,8\}$ and $\{11,12\}$. We make this informal observation precise with the following definitions, and in Theorem 4.3.1 we show that clow sequences which do not correspond to cycle covers cancel out.

We will associate a weight with a clow sequence that is consistent with the contribution of a cycle cover.

**Definition 4.3.4** The weight of a clow $C$, $w(C)$, is the product of the weights of the edges in the walk while accounting for multiplicity.

**Example 4.3.4** Given a matrix $A$, the weight of clow $C$ in example 4.3.2 is given by:

$$w((1,2,3,2,3))=a_{12}a_{23}^2a_{32}a_{31}$$

**Definition 4.3.5** The weight of a clow sequence $C=(C_1,\ldots,C_k)$ is:

$$w(C)=\prod_{i=1}^{k}w(C_i).$$

**Theorem 4.3.1**

$$\det(A) = \sum_{C \text{ is a clow sequence}} \operatorname{sign}(C) w(C)$$

*Proof.* The idea of the proof of Theorem 4.3.1 (see [MV97, pp. 5–8]) is that clow sequences which are not cycle covers cancel out. □

In [Val92, Section 3] Valiant points out that Berkowitz's algorithm computes sums of certain clow sequences; it computes the sums of clow sequences whose first head is the first vertex. Since the heads are ordered, if the first head is not the first vertex, then the given clow sequence is not a cycle cover (i.e., not a permutation), and hence it cancels out at the end, so sums of clow sequences with this restriction still compute correctly the determinant, and other coefficients of the characteristic polynomial.

More precisely, let $A$ be an $n \times n$ matrix, and let $p_n, p_{n-1}, \ldots, p_0$ be the coefficients of the char poly of $A$ as computed by Berkowitz's algorithm. Then, $p_n$ is the sum of clow sequences of length 0, $p_{n-1}$ is the sum of clow sequences of length 1, and in general $p_{n-i}$ is the sum of clow sequences of length $i$. In particular, $p_0$ is the determinant of $A$. Vertex 1 is the first head in the clow sequences computing each $p_{n-i}$ $(i > 0)$, and vertex 2 is the first head in the clow sequences computing each $q_{(n-1)-j}$ $(j > 0)$, where the $q_{(n-1)-j}$'s are the coefficients of the char polynomial of $M = A[1|1]$, etc.

We illustrate these computations with an example where $A$ is a $3 \times 3$ matrix.

**Example 4.3.5** Suppose that $A$ is a $3 \times 3$ matrix, $M = A[1|1]$ as usual, and $p_3, p_2, p_1, p_0$ are the coefficients of the char poly of $A$ and $q_2, q_1, q_0$ are the coefficients of the char poly or $M$, computed by Berkowitz's algorithm. Thus:

$$\begin{pmatrix} p_3 \\ p_2 \\ p_1 \\ p_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -a_{11} & 1 & 0 \\ -RS & -a_{11} & 1 \\ -RMS & -RS & -a_{11} \end{pmatrix} \begin{pmatrix} q_2 \\ q_1 \\ q_0 \end{pmatrix} = \begin{pmatrix} q_2 \\ -a_{11}q_2 + q_1 \\ -RSq_2 - a_{11}q_1 + q_0 \\ -RMSq_2 - RSq_1 - a_{11}q_0 \end{pmatrix} \tag{4.13}$$

The coefficients $q_2, q_1, q_0$ are computed by clow sequences on $M$, that is, by clow sequences on vertices $\{2, 3\}$, where the head of the first clow is always 2. See Figure 4.4. Since $q_2$ is the sum of clows of length zero (and 1 by default), so is $p_3$. Now consider $p_2$, which by definition is supposed to be the sum of clow sequences of length one on all three vertices, where the head of the first clow is vertex 1; see Figure 4.5. But this is the sum of clow sequences of length one on vertices 2 and 3 (i.e., $q_1$), plus the clow of length one on vertex

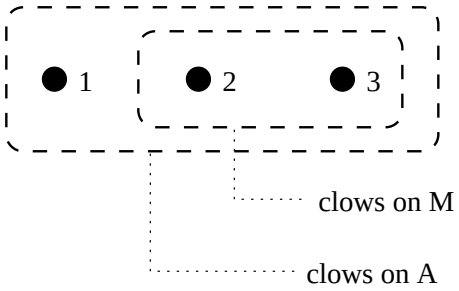



Figure 4.4: Clows on $A$ and $M = A[1|1]$

1, which is just $a_{11}$. All these clows have sign $-1$, hence the sum is $-a_{11}q_2 + q_1$ (again, $q_2 = 1$).

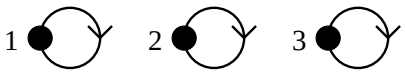



Figure 4.5: Clows of length one on all three vertices

Consider $p_1$: since $p_1 = p_{3-2}$, it follows that it is the sum of clow sequences of length two. We are going to show now how the term $-RSq_2 - a_{11}q_1 + q_0$ computes the sum of all these clow sequences.

There is just one clow of length two on vertices 2 and 3, it corresponds to $q_0$ and it is shown in Figure 4.6.

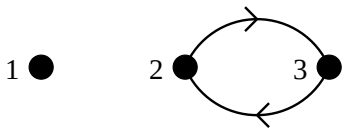



Figure 4.6: The single clow of length two on vertices 2 and 3

There are two clows of length two which include a self loop at vertex 1. These clows correspond to the term $-a_{11}q_1$. Note that the negative sign comes from the fact that $q_1$ has a negative value, but the parity of these clows is even. Both clows are shown in Figure 4.7.

Finally, we consider the clow sequences of length two, where there is no self loop at vertex 1. Since vertex 1 must be included, there are only two possibilities, both shown

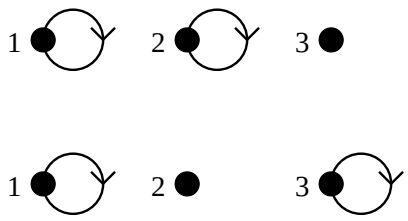



Figure 4.7: Clows of length two *with* a self loop at vertex 1

on Figure 4.8. These clows correspond to the term $-RSq_2$ which is equal to:

$$-\begin{pmatrix} a_{12} & a_{13} \end{pmatrix}\begin{pmatrix} a_{21} \\ a_{31} \end{pmatrix} = -a_{12}a_{21} - a_{13}a_{31}$$

since $q_2 = 1$.

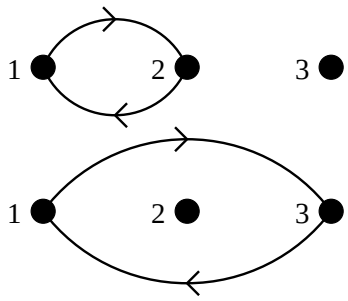



Figure 4.8: Clows of length two *without* a self loop at vertex 1

We do not show how to get $p_0$, but hopefully at this point the idea is clear.

# Chapter 5

# The Characteristic Polynomial

In Chapter 4 we showed that the char poly can be expressed as a term over $\mathcal{L}_{\text{LAP}}$. In this chapter we study the properties of the char poly that can be proven in LAP. In particular we show that we LAP proves the Cayley-Hamilton Theorem, and the multiplicativity of the determinant, for triangular matrices. It is an open question whether LAP can prove these properties for general matrices[1].

In Section 5.1 we prove some basic results in LAP that will be useful later. We also prove (in LAP) properties of the char poly which do not depend on the matrices being triangular. In Section 5.3 we show that hard matrix identities follow from the C-H Theorem (in LAP).

The most important property of the characteristic polynomial (char poly) is stated by the Cayley-Hamilton Theorem (C-H Theorem) which says the following: if $p_A$ is the char poly of $A$, then $p_A(A) = 0$. That is, the C-H Theorem states that the characteristic polynomial of $A$ is an annihilating polynomial of $A$. In general, we say that $p$ is an *annihilating polynomial* of a square matrix $A$, if:

$$p(A) = p_n A^n + p_{n-1}A^{n-1} + \cdots + p_1 A + p_0 I = 0 = \text{the zero matrix}$$

We can also state the C-H Theorem as:

$$A\text{adj}(A) = \text{adj}(A)A = \det(A)I \tag{5.1}$$

given our definition of the adjoint (see (4.6) on page 51).

[1] In Chapter 6 we show that LAP also proves the equivalence of some of the fundamental principles of Linear Algebra (for general matrices); see Table 6.1 on page 71. In Chapter 8 we show that the extension $\forall$LAP of LAP can prove the C-H Theorem (for general matrices), and therefore the C-H Theorem has a feasible proof, as $\forall$LAP can be interpreted in a standard poly-time theory.

We define the *correctness of Berkowitz's algorithm* to be the following property: given a matrix $A$, the polynomial obtained from Berkowitz's algorithm, $p_A$, is an annihilating polynomial of $A$. As a small clarification, note that for a matrix $A$, the output of Berkowitz's algorithm is a column vector, so when we say that "$p_A$ is obtained from Berkowitz's algorithm", we mean that the coefficients of $p_A$ are given by the entries of this column vector.

Thus, the correctness of Berkowitz's algorithm is the mechanism for proving the C-H Theorem; we define the polynomial computed by Berkowitz's algorithm to be the char poly, and hence, if we prove that it is an annihilating poly, then we also prove the C-H Theorem.

Another crucial property of the char poly is the *multiplicativity of the determinant*, (the determinant is defined from the constant coefficient of the char poly; see (4.7) on page 51) given by the identity:

$$\det(AB) = \det(A)\det(B) \tag{5.2}$$

The provability of (5.1) and (5.2), and other properties, is the subject of the next chapters. It turns out that while we can prove (5.1) and (5.2) for triangular matrices (see Section 5.2 of this chapter), we need to extend LAP to $\forall$LAP (which is LAP with induction on formulas with universally quantified matrix variables) in order to prove (5.1) and (5.2) for general matrices.

## 5.1 Basic properties

In this section we prove some basic results in LAP.

**Lemma 5.1.1** LAP proves that $a^n a^m = a^{n+m}$ and $A^n A^m = A^{n+m}$.

*Proof.* Both claims can be proven by induction on $n$. The **Basis Case** is when $n = 0$, so that $n + m = m$. Using (4.11) we have that $a^0 := \mathsf{e}(\mathsf{P}(0, \lambda ij\langle 1,1,a\rangle), 1, 1)$, and by A34, we have that $\mathsf{P}(0, \lambda ij\langle 1,1,a\rangle) = I_{\mathsf{r}(\lambda ij\langle 1,1,a\rangle)} = I_1$, so that $a^0 = \mathsf{e}(I_1, 1, 1) = 1$, and $1 \cdot a^m = a^m$ and $a^{0+m} = a^m$. For the **Induction Step** assume that the claim holds for $n$ and show that it holds for $n + 1$. Using A35, and the associativity of addition of index elements, we can easily show that $a^{(n+1)+m} = a^{n+m}a$, and $a^{n+1}a^m = a^n a a^m = a^n a^m a$, which is $a^{n+m}a$, by the induction hypothesis. Proving that $A^n A^m = A^{n+m}$ is similar. □

**Lemma 5.1.2** LAP proves that $(-1)^{\text{even power}} = 1$.

*Proof.* From (4.11) we know that $(-1)^{2n} := \mathtt{e}(\mathtt{P}(2n, \lambda ij\langle 1, 1, (-1)\rangle), 1, 1)$. So, we are going to prove by induction on $n$ that $\mathtt{e}(\mathtt{P}(2n, \lambda ij\langle 1, 1, (-1)\rangle), 1, 1) = 1$. The **Basis Case** is $n = 0$, so using A34, we get $\mathtt{P}(0, \lambda ij\langle 1, 1, (-1)\rangle) = I_{\mathtt{r}(\lambda ij\langle 1,1,(-1)\rangle)} = I_1$, and $\mathtt{e}(I_1, 1, 1) = 1$. For the **Induction Step**, suppose that $n > 0$. Using basic index operations we have that $2n = 2(n-1) + 2$. Using the IH we have that $(-1)^{2(n-1)} = 1$, and by basic arguments we have that $(-1)^2 = 1$. Now using Lemma 5.1.1, we have that $1 = (-1)^{2(n-1)}(-1)^2 = (-1)^{2(n-1)+2} = (-1)^{2n}$ and we are done. □

Let $p_A$ denote (as usual) the char poly of $A$ as computed by Berkowitz's algorithm. Let $(p_A)_i$ denote the $i$-th coefficient of the char poly $p_A$.

**Lemma 5.1.3** LAP proves that for any $A$, $\det(A) = a_{11}\det(M) - R\text{adj}(M)S$.

*Proof.* We use Definitions 4.2.4 and 4.2.5, that is the definitions of the adjoint and the determinant given in terms of Berkowitz's algorithm.

$$\det(A) = (-1)^n(p_A)_0$$

by definition of the determinant,

$$= (-1)^n(-a_{11}(p_M)_0 - (-1)^{n-2}R\text{adj}(M)S)$$

from Berkowitz's algorithm, and the definition of the adjoint—this is how we compute $(p_A)_0$ from $p_M$,

$$= a_{11}(-1)^{n-1}(p_M)_0 - R\text{adj}(M)S$$

by manipulating powers of $(-1)$ and by Lemma 5.1.2, $(-1)^{\text{even power}} = 1$,

$$= a_{11}\det(M) - R\text{adj}(M)S$$

This argument can be clearly formalized in LAP. □

**Lemma 5.1.4** LAP proves that for any $A$, $(p_A)_n = 1$, i.e., $p_A$ is a monic polynomial of degree $n$.

*Proof.* This can be easily proven by induction on $n$; just note that the top entry of the first column of any $C_i$ (recall that $p_A = C_1 C_2 \cdots C_n$) is 1, and the $C_i$'s are lower triangular. Thus, the top entry of $C_i C_{i+1} \cdots C_n$ is always 1. More formally, suppose that $(p_M)_{n-1} = 1$. By Berkowitz's algorithm, $(p_A)_n = (p_M)_{n-1}$ and we are done. □

**Lemma 5.1.5** LAP proves that for any $A$, $(p_A)_{n-1} = -\mathrm{tr}(A) = -\sum_{i=1}^{n} a_{ii}$.

*Proof.* This can also be proven easily by induction on $n$. So suppose that the claim holds for $M$, that is, $(p_M)_{n-2} = -\mathrm{tr}(M)$. From Berkowitz's algorithm we can see that $(p_A)_{n-1} = -a_{11} \cdot (p_M)_{n-1} + 1 \cdot (p_M)_{n-2}$. By Lemma 5.1.4, $(p_M)_{n-1} = 1$ and by the induction hypothesis $(p_M)_{n-2} = -\mathrm{tr}(M)$, so $(p_A)_{n-1} = -a_{11} - \mathrm{tr}(M) = -\mathrm{tr}(A)$. □

The matrix $I_{ij}$ is obtained from the identity matrix by interchanging the $i$-th and the $j$-th rows. The effect of multiplying $A$ on the left by $I_{ij}$ is that of interchanging the $i$-th and the $j$-th rows of $A$. On the other hand, $AI_{ij}$ is $A$ with the $i$-th and $j$-th columns interchanged. We sometimes abbreviate $I_{i(i+1)}$ by $I_i$. In Section 6.1 we show that:

$$I_{ij} = I_{i(i+1)} I_{(i+1)(i+2)} \cdots I_{(j-1)j} I_{(j-1)(j-2)} \cdots I_{(i+1)i}$$

that is: any permutation can be written as a product of transpositions; see proof of the Corollary 6.1.1.

**Lemma 5.1.6** LAP proves that, for $i \neq j$, $\det(I_{ij}) = -1$.

*Proof.* We prove the lemma by induction on the size of $I_{ij}$. Suppose first that $i, j > 1$. Then:

$$I_{ij} = \begin{pmatrix} 1 & 0 \\ 0 & I_{(i-1)(j-1)} \end{pmatrix}$$

where $I_{(i-1)(j-1)}$ is of size one less than $I_{ij}$. By Berkowitz's algorithm we have that $\det(I_{ij}) = \det(I_{(i-1)(j-1)})$, and by the Induction Hypothesis, $\det(I_{(i-1)(j-1)}) = -1$, so we are done in this case.

Otherwise, suppose that $i = 1$, $j > 1$. From Berkowitz's algorithm we have that:

$$\det(I_{1j}) = 0 \cdot \det(I_{1j}[1|1]) - e_j \mathrm{adj}(I_{1j}[1|1]) e_j$$

and $\mathrm{adj}(I_{1j}[1|1])$ is a matrix of zeros, except for the $(j, j)$-th position where it has a 1. To show this we argue by induction on the size of $I_{1j}[1|1]$ (it is not a difficult proof using

the definition of the adjoint, and the fact that $I_{1j}[1|1]$ is a "constant" matrix of 1s on the diagonal, zeros everywhere else, except for a single zero in the $(j,j)$ position). From this we have that $\det(I_{1j}) = -1$ as required. □

**Lemma 5.1.7** LAP proves that $A$ and $A^t$ have the same char poly, i.e., $p_A = p_{A^t}$.

*Proof.* The proof is by induction on the size of $A$. The Basis Case is trivial because $(a)^t = (a)$. Suppose now that $A$ is an $n \times n$ matrix, $n > 1$. By the IH we know that $p_M = p_{M^t}$. Furthermore, if we consider the matrix $C_1$ in the definition of Berkowitz's algorithm, we see that the entries 1 and $-a_{11}$ do not change under transposition of $A$, and also, since $S(M^t)^k R$ is a $1 \times 1$ matrix, it follows that $S(M^t)^k R = (S(M^t)^k)R)^t = RM^k S$, so in fact $C_1$ is the same for $A$ and $A^t$. This gives us the result. □

## 5.2 Triangular matrices

For the proofs in this section we are going to abuse notation a little bit, and write $p_A(x)$ for the characteristic poly of $A$, even though technically the char poly in LAP is a column vector $p_A$ containing the coefficients of the char poly of $A$. This will simplify our proofs.

Also note that in the Lemmas and Corollaries below, we always show that some property can be proven in LAP. We do this by giving a high-level proof of this property, where we only indicate what would the formal LAP proof consist of. It would be tedious and unreadable to give complete LAP proofs in each case. LAP has been designed in a way that permits us a certain degree of freedom when presenting proofs that can be formalized in LAP.

Basically we assume that any proof that relies on matrix powering, and induction on terms of type index, and uses basic matrix properties, can be formalized in LAP.

**Lemma 5.2.1** LAP proves that if for all $i$, $R_i = 0$ or $S_i = 0$, then $p_A(A) = 0$.

Before we prove this Lemma, note that $p_A(A)$ denotes the matrix given by:

$$(p_A)_n A^n + (p_A)_{n-1} A^{n-1} + \cdots + (p_A)_1 A + (p_A)_0 I_{\mathtt{r}(A)}$$

so in fact, $p_A(A) = 0$ should really be stated as:

$$\mathtt{r}(A) = \mathtt{c}(A) \to p_A(A) = 0_{\mathtt{r}(A)\mathtt{c}(A)}$$

where $p_A(A)$ is an abbreviation for the following constructed matrix:

$$\lambda ij\langle \mathtt{r}(A), \mathtt{c}(A), \mathtt{e}(p_A \cdot \lambda kl\langle \mathtt{r}(A), 1, \mathtt{e}(\mathtt{P}(\mathtt{r}(A) - k, A), i, j)\rangle, i, j\rangle$$

where $p_A$ has been defined on page 52, definition (4.9).

*Proof.* The proof is by induction on the size of $A$. The Basis Case is trivial. For the Induction Step assume that $A$ is of the usual form:

$$A = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

Suppose that $S = 0$ (the case where $R = 0$ is analogous). Then, from Berkowitz's algorithm, we have that:

$$p_A(x) = (x - a_{11})p_M(x)$$

so $p_A(A) = (A - a_{11}I)p_M(A)$, and:

$$p_M(A) = \begin{pmatrix} p_M(a_{11}) & X \\ 0 & p_M(M) \end{pmatrix}$$

where $X$ is some $1 \times (n-1)$ matrix. Now, using the IH, $p_M(M) = 0$. Thus:

$$(A - a_{11}I)p_A(A) = \begin{pmatrix} 0 & R \\ 0 & M - a_{11}I \end{pmatrix} \begin{pmatrix} p_M(a_{11}) & X \\ 0 & 0 \end{pmatrix} = 0$$

Thus, $p_A(A) = 0$. □

**Corollary 5.2.1** LAP proves the C-H Theorem for triangular matrices.

*Proof.* By Lemma 5.2.1, $p_A(A) = 0$ if for all $i$ $R_i = 0$ or $S_i = 0$. If $A$ is triangular, then $R_i = 0$ for all $i$ or $S_i = 0$ for all $i$. □

**Lemma 5.2.2** LAP proves that if for all $i$, $R_i = 0$ or $S_i = 0$, then $\det(A) = \prod_i a_{ii}$.

*Proof.* First of all, we can express $\prod_{i=1}^{n} a_{ii}$ in LAP as follows:

$$\mathtt{e}(n+1, n+1, \mathtt{P}(n, \lambda ij\langle n+1, n+1, \mathrm{cond}(\mathtt{e}(i, i, A), 0_{\mathrm{field}}, j = i+1)\rangle)) \tag{5.3}$$

(that is, as the $(n+1, n+1)$ entry of the $n$-th power of the matrix with the $a_{ii}$'s on the diagonal above the main diagonal, and zeros elsewhere). Suppose now that $A$ is such

that for all $i$, $R_i = 0$ or $S_i = 0$. Then:

$$p_A = C_1 C_2 \cdots C_n, \qquad \text{where } C_i = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -a_{11} & 1 & \dots & 0 \\ & & \ddots & 1 \\ & 0 & & -a_{ii} \end{pmatrix} \tag{5.4}$$

Now, using induction on $n$, we can show that the bottom row of $p_A$ is equal to (5.3). The **Basis Case** is $n = 1$, and it is easy since the bottom entry of $A$ is just $a_{11}$, and (5.3) is just $a_{11}$ as well. For the **Induction Step** assume that this holds for $n \times n$ matrices, so that the bottom row of $C_2 C_3 \cdots C_n$ is equal to $a_{22} a_{33} \cdots a_{(n+1)(n+1)}$. Now multiply $C_2 C_3 \cdots C_n$ by $C_1$ on the left to get the result. □

**Corollary 5.2.2** LAP proves that the determinant of a triangular matrix is the product of the elements on the diagonal.

*Proof.* By Lemma 5.2.2 the determinant is the product of the elements on the diagonal if for all $i$, $R_i = 0$ or $S_i = 0$. In the case of a triangular matrix one or the other always holds. □

**Lemma 5.2.3** LAP proves that if $A$ and $B$ are both upper or lower-triangular, then $\det(AB) = \det(A)\det(B)$.

*Proof.* Suppose that $A, B$ are both upper triangular matrices (the case of lower triangular matrices is analogous). Then $AB$ is also an upper triangular matrix. To see this consider the entry $(i, j)$ of $AB$ where $i > j$. This entry is given by $\sum_{k=1}^{n} a_{ik} b_{kj}$. Since both $A, B$ are upper triangular, it follows that $a_{ik} b_{kj} = 0$ if $i > k$ or $j < k$ which is always the case as $i < j$. Thus the $(i, j)$-th entry of $AB$ is zero when $i < j$, so $AB$ is upper triangular. By Corollary 5.2.2 the determinants of $A, B$ and $AB$ are the products of the elements on the respective diagonals. It is easy to show that the $(i, i)$ entry of $AB$ is just $a_{ii} b_{ii}$ (using the same argument that we did to show that $AB$ is upper triangular). The Lemma now follows. □

## 5.3 Hard matrix identities

In this section we show that $AB = I \rightarrow BA = I$, and hence all hard matrix identities, follow (in LAP) from the Cayley-Hamilton Theorem. It is interesting to note that we know nothing about the converse; that is, what role do hard matrix identities play in the proof of the C-H Theorem? Our proof of the C-H Theorem, given in Chapter 8, does not come anywhere near hard matrix identities.

**Theorem 5.3.1** LAP proves that the Cayley-Hamilton Theorem implies hard matrix identities.

*Proof.* Suppose that $AB = I$, and let $p$ be the char poly of $A$. First note that it can be proven (in LA in fact) that $AB = I \rightarrow A(BA - I) = 0$. To see this note that $AB = I$ implies that $(AB)A = IA = A$, and by associativity $A(BA) = (AB)A = A$, so $A(BA) + (-1)A = 0$, so $A(BA + ((-1)A)I = 0$, so $A(BA) + A((-1)I) = 0$. Now using distributivity we obtain $A(BA + (-1)I) = 0$.

Thus, to show that $BA = I$, it is enough to show that $A$ has *some* left inverse $C$ (which of course turns out to be $B$) and use the identity $AB = I \rightarrow A(BA - I) = 0$ as follows: $C(A(BA - I)) = 0$ implies (by associativity) that $(CA)(BA - I) = 0$, and if $C$ is the left inverse of $A$, we obtain $I(BA - I) = 0$, from which $BA = I$ follows.

We construct the left inverse of $A$ using:

$$\text{The Cayley-Hamilton Theorem: } p(A) = 0 \ (p = p_A)$$
$$\text{and the identity: } 1 \leq i, AB = I \rightarrow A^i B^i = I$$

The identity follows in LAP by induction on $i$; just note that the Basis Case is the claim $AB = I$, and the Induction Step can be proven as follows: $A^{i+1}B^{i+1} = (A^iA)(BB^i)$, and now using associativity, this is equal to $A^i(AB)B^i$ which is just $A^iB^i$, which is $I$ by the induction hypothesis.

We now concentrate on the characteristic polynomial of $A$, $p$. By the C-H Theorem $p(A) = 0$. Let $p_n, p_{n-1}, \ldots, p_0$ be the coefficients of the characteristic polynomial, so that $p(A) = p_nA^n + p_{n-1}A^{n-1} + \cdots + p_0I = 0$. Suppose that $p_0$ is not zero. Then:

$$(p_nA^{n-1} + p_{n-1}A^{n-2} + \cdots p_1I)A = -p_0I$$

Dividing both sides by $-p_0$ we obtain the left inverse of $A$ as desired.

Suppose now that $p_0 = 0$. Let $i$ be the largest index such that $p_0 = p_1 = \ldots = p_i = 0$ and $p_{i+1} \neq 0$. Note that such an $i$ exists, and furthermore, $i + 1 \leq n$, since $p_n = 1$ (as was proven in LAP, by induction on $n$, in Lemma 5.1.4). Let $q$ be the polynomial with coefficients $p_n, p_{n-1}, \ldots, p_{i+1}$, so that $0 = p(A) = q(A)A^i$. Since by the above $A^iB^i = I$, it follows that $0 = 0B^i = q(A)A^iB^i = q(A)I = q(A)$. Since the constant coefficient of $q$ is $p_{i+1} \neq 0$, we can repeat the above argument to conclude that $A$ has a left inverse. $\square$

In Chapter 8 we give a feasible proof of the C-H Theorem, which, together with Theorem 5.3.1 gives us a feasible proof of $AB = I \rightarrow BA = I$, and hence, by results in Section 3.2, feasible proofs of hard matrix identities.

Note that the main thing that we need in the above proof is an annihilating polynomial. The C-H Theorem states that the char poly is an annihilating polynomial, so $AB = I \rightarrow BA = I$ follows from the C-H Theorem, but *any* annihilating polynomial would do.

Since $\{I, A, A^2, \ldots, A^{n^2}\}$ is a linearly dependent set of matrices, for $A$ an $n \times n$ matrix, there are non-zero coefficients that constitute an annihilating polynomial of $A$; if we could compute theses coefficients (without using Gaussian Elimination, but rather in $\mathrm{NC}^2$), and show, in LAP, that they form an annihilating polynomial, we would have an LAP proofs of hard matrix identities without the C-H Theorem; is that possible?

# Chapter 6

# Equivalences in LAP

In this Chapter we show that LAP proves the following implications:

| | | | |
|---|---|---|---|
| C-H Theorem | $\Longrightarrow$ | Axiomatic dfn of det | (Section 6.1) |
| Axiomatic dfn of det | $\Longrightarrow$ | Cofactor Expansion | (Section 6.2) |
| Cofactor Expansion | $\Longrightarrow$ | C-H Theorem | (Section 6.3) |
| Multiplicativity of det | $\Longrightarrow$ | C-H Theorem | (Section 6.4) |

Table 6.1: Flowchart for Chapter 6

In Section 6.4 we show that LAP also proves the multiplicativity of the determinant from the C-H Theorem and the following identity:

$$\det(A) = 0 \rightarrow AB \neq I \tag{6.1}$$

Thus, LAP proves the equivalence of the C-H Theorem, the axiomatic definition of the determinant, and the cofactor expansion. In Chapter 8 we will give a feasible proof of identity (6.1) (but not an LAP proof), from which it follows that we can give a feasible proof of the multiplicativity of the determinant from the C-H Theorem.

It is an open question whether identity (6.1) has an LAP proof, and whether we can prove, in LAP, that the multiplicativity of the determinant follows from the C-H Theorem. In fact, our proof of identity (6.1) (see Section 8.3.2) relies on the Gaussian Elimination algorithm.

## 6.1 The axiomatic definition of determinant

The *axiomatic definition of the determinant* states that for any matrix $A$, the following three conditions hold:

- det is multilinear in the rows and columns of $A$
- det is alternating in the rows and columns of $A$
- if $A = I$, then $\det(A) = 1$

In this section we show that the axiomatic definition of the determinant follows from the Cayley-Hamilton Theorem, and that this can be shown in LAP. The condition $\det(A) = 1$ is easy, and multilinearity in the first row (and column) is easy as well. Thus the whole proof hinges on a LAP proof of alternation from the C-H Theorem. Our final result, Corollary 6.1.2, shows that alternation for a matrix $A$ follows (in LAP) by applying the C-H Theorem to minors of permutations of rows and columns of $A$.

Note that from this it follows that det (as defined in 4.2.5), is the *true* determinant.

Multilinearity in the first row and column follows immediately from the algorithm; thus, we will have multilinearity for all rows and columns if we prove alternation. By Corollary 5.2.2, $\det(I) = 1$ as required. Thus, all we have to prove is alternation, which is the difficult part of the proof.

It is in fact enough to prove alternation in the rows, as alternation in the columns will follow from alternation in the rows by $\det(A) = \det(A^t)$—Lemma 5.1.7.

The strategy for showing alternation in the rows is the following: we first show that for any matrix $A$, $A$ and $I_1AI_1$ have the same char poly (Lemma 6.1.1). Recall that $I_1$ is an abbreviation for $I_{12}$, which in turn is the matrix obtained from the identity matrix by permuting the first two rows. In general, $I_{ij}$ is the identity matrix with rows $i$ and $j$ interchanged. Therefore $I_{ij}A$ is $A$ with rows $i$ and $j$ interchanged, and $AI_{ij}$ is $A$ with columns $i$ and $j$ interchanged. Finally, $I_i$ abbreviates $I_{i(i+1)}$.

Once we prove that $A$ and $I_1AI_1$ have the same char poly, we can also show that $A$ and $I_iAI_i$ have the same char poly (Lemma 6.1.2). From this we get that $A$ and $I_{ij}AI_{ij}$ have the same char poly (as any permutation is a product of transpositions; see Corollary 6.1.1).

Also in Lemma 6.1.1 we show that $\det(A) = -\det(I_1A)$. From this it follows that $\det(A) = -\det(I_{1i}A)$ for all $i$, since we can bring the $i$-th row to the second position (via

$I_{2i}AI_{2i}$), apply Lemma 6.1.1, and reorder things (by applying $I_{2i}AI_{2i}$ once more). Since $I_{ij} = I_{1i}I_{1j}I_{1i}$, this gives us alternation in the rows.

Note that we require the Cayley-Hamilton Theorem in the proof of every Lemma. Also note that we prove that $A$ and $I_{ij}AI_{ij}$ have the same char poly, i.e. $p_{I_{ij}AI_{ij}} = p_A$, to be able to reorder the matrix to prove alternation.

**Lemma 6.1.1** Let $A$ be an $n \times n$ matrix, and let $M_2$ be the second principal submatrix of $A$ (i.e., $M_2$ is $A$ *without* the first two rows and the first two columns). Then, LAP proves that $p_{M_2}(M_2) = 0$ implies:

- $p_{(I_1AI_1)} = p_A$ (i.e., $I_1AI_1$ and $A$ have the same characteristic poly)
- $\det(A) = -\det(I_1A)$

*Proof.* The proof consists of Claims 6.1.1 and 6.1.2, given below. □

Since we want to study the effect of interchanging the first two rows and columns of $A$, we let $A$ be of the following form:

$$A = \begin{pmatrix} a & b & R \\ c & d & P \\ S & Q & M_2 \end{pmatrix}$$

where $M_2$ is an $(n-2)\times(n-2)$ matrix, $a, b, c, d$ are entries, and $R, P, S^t, Q^t$ are $1\times(n-2)$ matrices. We are going to consider $I_1AI_1$ and $I_1A$. To this end we define $\sigma A := I_1AI_1$ and we define $\tau A := I_1A$. In terms of entries of $A$, $\sigma$ and $\tau$ are given as follows:

$$\begin{array}{cc} a,b,c,d \overset{\sigma}{\mapsto} d,c,b,a & a,b,c,d \overset{\tau}{\mapsto} c,d,a,b \\ R,S,P,Q \overset{\sigma}{\mapsto} P,Q,R,S & R,P \overset{\tau}{\mapsto} P,R \\ M_2 \overset{\sigma}{\mapsto} M_2 & S,Q,M_2 \overset{\tau}{\mapsto} S,Q,M_2 \end{array}$$

To illustrate the main idea, we show that $A$ and $\sigma A$ have the same char poly, in the case where $M_2$ is a $1 \times 1$ matrix (so $A$ is a $3 \times 3$ matrix). Let $p_A = C_1C_2C_3$.

From Berkowitz's algorithm, $C_1C_2$ is given by:

$$\begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} & -a & 1 \\ -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} & -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} & -a \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -d & 1 \\ -PQ & -d \end{pmatrix}$$

which is:

$$\begin{pmatrix} 1 & 0 \\ -a-d & 1 \\ -bc-RS+ad-PQ & -a-d \\ -bPS-cRQ-RM_2S+dRS+aPQ & -bc-RS+ad \end{pmatrix} \tag{6.2}$$

It is easy to see that all the entries in (6.2), except for those in the last row, remain invariant under $\sigma$.

However, the same is not true for the two entries in the bottom row. If we permute the first two rows and columns, the left entry of the bottom row is left with $-PM_2Q$ in place of $-RM_2S$, and the right entry is left with $-PQ$, in place of $-RS$; neither term appears before the permutation.

The reason why $-PM_2Q$ and $-PQ$ do not matter is because, when we multiply (6.2) by $C_3 = \begin{pmatrix} 1 & -M_2 \end{pmatrix}^t$ (which is the char poly of the $1 \times 1$ matrix $M_2$) these two terms cancel each other out: $-PM_2Q + PQM_2 = 0$.

In general, if $A$ is any $n \times n$ matrix, then we can show that all the entries in $C_1C_2$ are invariant under $\sigma$, except for the entries in the last row. These entries will be left with the following terms:

$$-PM_2^{n-2}Q \quad -PM_2^{n-3}Q \quad \ldots \quad -PQ \tag{6.3}$$

which did not appear before we applied $\sigma$. However, as before, they do not matter, because $C_3C_4 \cdots C_n$ computes the char poly of $M_2$, so when we multiply all the matrices out, the terms in (6.3) will simply disappear (by the Cayley-Hamilton Theorem).

To prove Lemma 6.1.1 we start by showing that all the entries in $C_1C_2$, except those in the last row, are invariant under $\sigma$. This is Claim 6.1.1.

**Claim 6.1.1** Let $A$ be an $n \times n$ matrix, for some $n \geq 3$. Then, LAP proves that all the entries in $C_1C_2$, except for those in the last row, remain invariant under $\sigma$.

*Proof.* Note that $(C_1C_2)[n+1|-]$ is a lower-triangular Toeplitz matrix. We consider the first column of $(C_1C_2)[n+1|-]$. The top three entries of the first column are:

$$\begin{gathered} 1 \\ -a-d \\ -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} + ad - PQ = -bc - RS + ad - PQ \end{gathered}$$

By inspection, they are all invariant under $\sigma$.

The $(k+1)$-st entry in the first column, for $k \geq 3$, is given by taking the dot-product of the following two vectors:

$$\begin{pmatrix} 1 \\ -a \\ -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} \\ -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} \\ \vdots \\ -\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}^{k-2}\begin{pmatrix} c \\ S \end{pmatrix} \end{pmatrix}, \quad \begin{pmatrix} -PM_2^{k-2}Q \\ -PM_2^{k-3}Q \\ \vdots \\ -PQ \\ -d \\ 1 \end{pmatrix} \tag{6.4}$$

We are going to prove that this dot-product is invariant under $\sigma$, by induction on $k$. The **Basis Case** is $k = 3$, where the dot product is given by:

$$-\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} + d\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} + aPQ - PM_2Q \tag{6.5}$$

and the invariance under $\sigma$ again follows by inspection.

For the **Induction Step**, consider the $(k+1)$-st entry $(k \geq 3)$ of the first column of $(C_1C_2)[n+1|-]$:

$$\begin{pmatrix} b & R \end{pmatrix}\begin{pmatrix} w & X \\ Y & Z \end{pmatrix}\begin{pmatrix} c \\ S \end{pmatrix} + aPM_2^{k-3}Q - PM_2^{k-2}Q \tag{6.6}$$

where $w, X, Y, Z$ are given as follows:

$$\begin{pmatrix} w & X \\ Y & Z \end{pmatrix} = -\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}^{k-2} + d\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}^{k-3} + \sum_{i=0}^{k-4} PM_2^{k-4-i}Q\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}^{i} \tag{6.7}$$

Assume that (6.6) is invariant under $\sigma$ (this is our Induction Hypothesis). The $(k+2)$-nd entry $(k \geq 3)$ of the first column of $(C_1C_2)[n+1|-]$ is given by:

$$\begin{pmatrix} b & R \end{pmatrix}\left(\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}\begin{pmatrix} w & X \\ Y & Z \end{pmatrix} + (PM_2^{k-3}Q)I\right)\begin{pmatrix} c \\ S \end{pmatrix} + aPM_2^{k-2}Q - PM_2^{k-1}Q \tag{6.8}$$

We must show that (6.8) is invariant under $\sigma$ using the Induction Hypothesis. To see this, first note that the expression:

$$\begin{pmatrix} d & P \\ Q & M_2 \end{pmatrix}\begin{pmatrix} w & X \\ Y & Z \end{pmatrix} + (PM_2^{k-3}Q)I$$

in (6.8) is just (6.7) where instead of $k-2, k-3, k-4$ we have $k-1, k-2, k-3$. Since in (6.8) we have $aPM_2^{k-2}Q - PM_2^{k-1}Q$ (as opposed to $aPM_2^{k-3}Q - PM_2^{k-2}Q$ in (6.6)) it follows that the symmetry under $\sigma$ is preserved. This is an elementary argument, using powers of matrices and induction on indices, and hence it can be formalized in LAP. $\square$

**Claim 6.1.2** Let $A$ be an $n \times n$ matrix, for some $n \geq 3$. Then, LAP proves that $p_{M_2}(M_2) = 0$ implies that the entry in the bottom row of $C_1C_2C_3 \cdots C_n$ remain invariant under $\sigma$ and changes sign under $\tau$.

*Proof.* The bottom row of $C_1C_2$ is given by the dot product of the two vectors in (6.4) without their top rows. Thus, in the bottom row of $C_1C_2$, we are missing $-PM_2^{k-2}Q$'s in the summations.

If we add these missing terms accross the bottom row (starting with the left-most), that is, if we add:

$$-PM_2^{n-2}Q, -PM_2^{n-3}Q, \ldots, -PM_2Q, -PQ \tag{6.9}$$

to the entries in the bottom row, respectively, we can conclude, by the previous claim, that the result is invariant under $\sigma$.

We have that $p_{M_2}(M_2) = 0$, so $-Pp_{M_2}(M_2)Q = 0$, and since $p_{M_2} = C_3C_4 \ldots C_n$, it follows that if we multiply the bottom row of $C_1C_2$, where the terms listed in (6.9) have been added, by $p_{M_2} = C_3C_4 \cdots C_n$, these terms will disappear.

Hence, to prove the invariance under $\sigma$ of the bottom entry of $C_1C_2 \cdots C_n$, we first add the extra terms in (6.9) to the bottom row of $C_1C_2$, use the previous claim to conclude the invariance of the resulting bottom row of $C_1C_2$ under $\sigma$ (which does not affect $C_3C_4 \cdots C_n$), and then show that the extra terms disappear by $p_{M_2}(M_2) = 0$ (that is, by the Cayley-Hamilton Theorem applied to $M_2$).

The fact that the bottom row of $C_1C_2C_3 \cdots C_n$ changes sign under $\tau$ is also a small variation of the argument given here and given in the proof of Claim 6.1.1. $\square$

**Lemma 6.1.2** Let $A$ be an $n \times n$ matrix. Then LAP proves that $p_{M_{i+1}}(M_{i+1}) = 0$ implies $p_{(I_i A I_i)} = p_A$.

*Proof.* See Figure 6.1, and note that if $i \geq n-1$ then $M_{i+1}$ is not defined, but this is not a problem, since we do not need the C-H Theorem to prove $p_{I_{n-1}AI_{n-1}} = p_A$.

The case $i = 1$ is Lemma 6.1.1, so we can assume that $1 < i < n-1$.

Using the fact that $I_i^2 = I$, we have:

$$RM^jS = R(I_iI_i)M^j(I_iI_i)S = (RI_i)(I_iM^jI_i)(I_iS) = (RI_i)(I_iMI_i)^j(I_iS) \tag{6.10}$$

Here we use induction on $j$ in the last step. The Basis Case is $j = 1$, so $I_iMI_i = I_iMI_i$ just by equality axioms. For the Induction Step, note that:

$$I_iM^{j+1}I_i = I_iM^jMI_i = I_iM^j(I_iI_i)MI_i = (I_iM^jI_i)(I_iMI_i)$$

and by the induction hypothesis, $I_iM^jI_i = (I_iMI_i)^j$, so we are done.

From Berkowitz's algorithm we know that the char poly of $A$ is given by the following product of matrices:

$$C_1C_2\cdots C_{i-1}C_i\cdots C_n$$

Let $C_1'C_2'\cdots C_n'$ be the char poly of $I_iAI_i$. As an aside, note that we defined Berkowitz's algorithm as a term over $\mathcal{L}_{\text{LAP}}$ in Section 4.2.3. There, we padded the matrices $C_1, \ldots, C_n$ with zeros to make them all of equal size, and we put them in one big matrix $C$. Then, by computing the $n$-th power of $C$, we obtain the iterated matrix product $C_1C_2\cdots C_n$. Here, whenever we talk of iterated matrix products, we have this construction in mind.

Using Lemma 6.1.1 and $p_{M_{i+1}}(M_{i+1}) = 0$, we know that if we interchange the first two rows and the first two columns of $M_{i-1}$ (which are contained in the $i$-th and $(i+1)$-st rows and columns of $A$), the char poly of $M_{i-1}$ remains invariant. This gives us:

$$C_iC_{i+1}\cdots C_n = C_i'C_{i+1}'\cdots C_n' \tag{6.11}$$

Now we are going to prove that for $1 \leq k \leq i-1$, $C_k = C_k'$. To see this, consider the first column of $C_k'$ (it is enough to consider the first column as these are Toeplitz matrices). We are going to examine all the entries in this columns:

- The first entry is 1, which is a constant.
- The second entry is $a_{kk}$, just as in $C_k$ since $k \leq i-1$.
- $R_kM_k^jS_k$ is replaced by $(R_kI_{i+1-k})(I_{i+1-k}M_kI_{i+1-k})^j(I_{i+1-k}S_k)$, but by (6.10) these two are equal. (Note that $0 \leq j \leq n-k-1$).

Thus, $C_k = C_k'$, for $1 \leq k \leq i-1$ and so $C_1C_2\cdots C_{i-1} = C_1'C_2'\cdots C_{i-1}'$. Combining this with (6.11) gives us:

$$C_1C_2\cdots C_n = C_1'C_2'\cdots C_n'$$

and so $A$ and $I_iAI_i$ have the same char polynomial, i.e., $p_{(I_iAI_i)} = p_A$. □

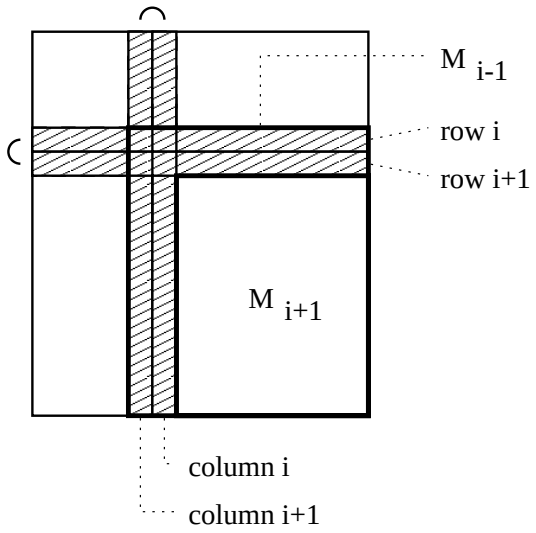



Figure 6.1: Matrix $A$: $p_{M_{i+1}}(M_{i+1}) = 0 \Longrightarrow p_{(I_i A I_i)} = p_A$

**Corollary 6.1.1** Let $A$ be an $n \times n$ matrix, and let $1 \leq i < j \leq n$. LAP proves, using the C-H Theorem on $(n-1) \times (n-1)$ matrices, that $p_{I_{ij} A I_{ij}} = p_A$.

*Proof.* First of all, to prove this Corollary to Lemma 6.1.2, we are going to list explicitly the matrices for which we require the C-H Theorem: we need the following principal submatrices of $A$: $\{M_{i+1}, \ldots, M_j\}$ as well as the matrices $\{M'_{j-1}, \ldots, M'_{i+1}\}$ which are obtained from the corresponding principal submatrices, by replacing, in $A$, the $j$-th row by the $i$-th row, and the $j$-th column by the $i$-th column. The details are given in Figure 6.2.

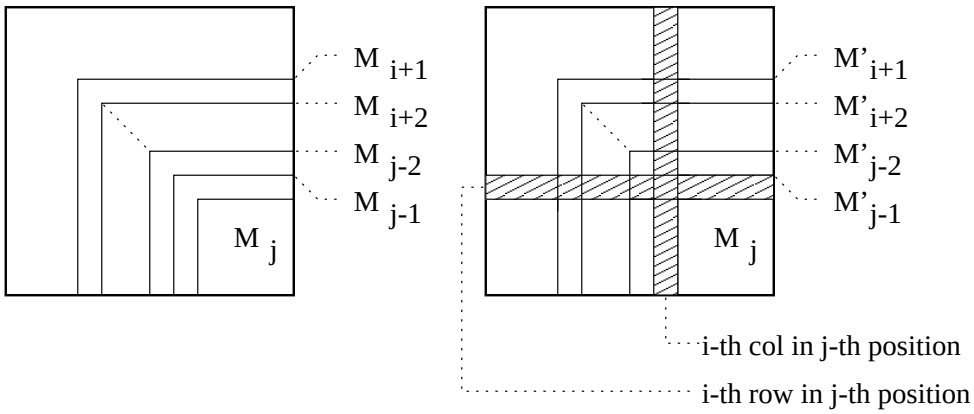



Figure 6.2: $\{M_{i+1}, \ldots, M_j\}$ and $\{M'_{j-1}, \ldots, M'_{i+1}\}$

To see why we require the C-H Theorem on precisely the matrices listed above, we illustrate how we derive $p_{(I_{13}AI_{13})} = p_A$ (see Figure 6.3). Using $p_{M_2}(M_2) = 0$ and Lemma 6.1.2 we interchange the first two rows (and the first two columns, but for clarity, we do not show the columns). Then, using $p_{M_3}(M_3) = 0$ and Lemma 6.1.2, we interchange rows two and three, so at this point, the original row one is in position. We still need to take the original row three from position two to position one. This requires the use of $p_{M_2'}(M_2') = 0$ and Lemma 6.1.2. The prime comes from the fact that what used to be row three, has now been replaced by row one. So using $p_{M_2'}(M_2') = 0$, we exchange row two and one, and everything is in position.

Now the same argument, but in the general case, relies on the fact that:

$$I_{ij} = I_{i(i+1)}I_{(i+1)(i+2)} \cdots I_{(j-1)j}I_{(j-1)(j-2)} \cdots I_{(i+1)i} \tag{6.12}$$

i.e., any permutation can be written as a product of transpositions. Using Lemma 6.1.2 at each step, we are done. Equation (6.12) can be proven in LAP as follows: first note that $I_{ij} = I_{1i}I_{1j}I_{1i}$, so it is enough to prove that $I_{1i}$ is equal to a product of transpositions, for any $i$.

We use induction on $i$. The **Basis Case** is $i = 2$, and $I_{12}$ is a transposition, so there is nothing to prove. Now the **Induction Step**. Assume the claim holds for $I_{1i}$, and show that it holds for $I_{1(i+1)}$. This follows from the fact that $I_{1(i+1)} = I_{1i}I_{i(i+1)}I_{1i}$. □

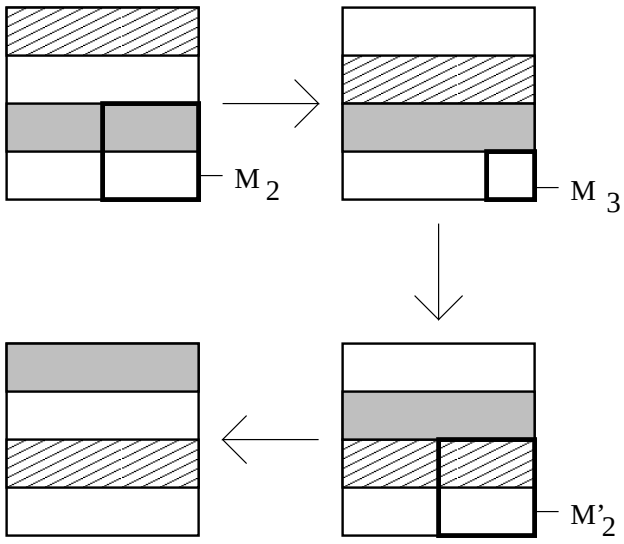



Figure 6.3: Example of $p_{(I_{13}AI_{13})} = p_A$

**Corollary 6.1.2** LAP proves, using the C-H Theorem, that det is alternating in the rows, i.e., $\det(A) = -\det(I_{ij}A)$.

*Proof.* Since $I_{ij} = I_{1i}I_{1j}I_{1i}$, it is enough to prove this for $I_{1j}$. If $j = 2$ we are done by Lemma 6.1.1. If $j > 2$, then use $I_{2j}$ to bring the $j$-th row to the second position, and by Corollary 6.1.1, $A$ and $I_{2j}AI_{2j}$ have the same char polynomials. Now apply $I_{12}$ with Lemma 6.1.1, and use $I_{2j}$ once again to put things back in order. □

**Example 6.1.1** Suppose that we want to show that $\det(A) = -\det(I_{15}A)$. Consider:

$$A \xrightarrow{(1)} I_{25}AI_{25} \xrightarrow{(2)} I_{12}I_{25}AI_{25} \xrightarrow{(3)} I_{25}I_{12}I_{25}AI_{25}I_{25} = I_{15}A$$

By Corollary 6.1.1, (1) preserves the char poly, and hence it preserves the determinant. By Lemma 6.1.1, (2) changes the sign of the determinant. By Corollary 6.1.1 again, (3) preserves the determinant. Therefore, $\det(A) = -\det(I_{15}A)$.

## 6.2 The cofactor expansion

Let $A$ be an $n \times n$ matrix. The *cofactor expansion formula for* $A$ states the following:

$$\text{for } 1 \leq i \leq n,\ \det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A[i|j]) \tag{6.13}$$

where $A[i|j]$ denotes the matrix obtained from $A$ by removing the $i$-th row and the $j$-th column. For each $i$, the RHS of the equation is called the *cofactor expansion of $A$ along the $i$-th row*, and (6.13) states that we obtain $\det(A)$ expanding along any row of $A$.

Note that from (6.13), it follows by Lemma 5.1.7 that we also have the cofactor expansion along columns.

**Lemma 6.2.1** LAP proves that the cofactor expansion formula (6.13) follows from the axiomatic definition of the determinant.

*Proof.* We first show that the cofactor expansion of $A$ along the first row is equal to $\det(A)$. Define $A_j$, for $1 \leq j \leq n$, to be $A$, with the first row replaced by zeros, except for the $(1, j)$-th entry which remains unchanged. Then, using multilinearity along the first row of $A$, we obtain:

$$\det(A) = \det(A_1) + \det(A_2) + \cdots + \det(A_n) \tag{6.14}$$

Consider $A_j$, for $j > 1$. If we interchange the first column and the $j$-th column, and then, with $(j-2)$ transpositions we bring the first column (which is now in the $j$-th position) to the second position, we obtain, by alternation and Lemma 5.1.3, the following:

$$\det(A_j) = (-1)^{j-1} a_{1j} \det(A[1|j]) = (-1)^{1+j} a_{1j} \det(A[1|j])$$

From this, and from equation (6.14), we obtain the cofactor expansion along the first row, that is, we obtain (6.13) for $i = 1$.

If we want to carry out the cofactor expansion along the $i$-th row (where $i > 1$), we interchange the first and the $i$-th row, and then we bring the first row (which is now in the $i$-th position) to the second row with $(i-2)$ transposition. Denote this new matrix $A'$, and note that $\det(A') = (-1)^{i-1} \det(A)$. Now, expanding along the first row of $A'$, we obtain (6.13) for $i > 1$. □

## 6.3 The adjoint as a matrix of cofactors

In this section we show that LAP proves the Cayley-Hamilton Theorem from the cofactor expansion formula (i.e., from (6.13)). To this end, we first show that (6.13) implies the axiomatic definition of determinant:

**Lemma 6.3.1** LAP proves the axiomatic definition of the determinant from the cofactor expansion formula.

*Proof.* We want to show that we can get multilinearity, alternation and $\det(I) = 1$ from (6.13). To show multilinearity along row (column) $i$, we just expand along row (column) $i$. To show $\det(I) = 1$ use induction on the size of $I$; in fact, showing that $\det(I) = 1$ can be done in LAP without any assumptions—see Corollary 5.2.2.

Alternation follows from multilinearity and from:

$$\text{If two rows (columns) of } A \text{ are equal } \rightarrow \det(A) = 0$$

To see that alternation follows from these two things:

$$0 = \det\begin{pmatrix} \\ R_i + R_j \\ \\ R_i + R_j \\ \\ \end{pmatrix} = \det\begin{pmatrix} \\ R_i \\ \\ R_i \\ \\ \end{pmatrix} + \det\begin{pmatrix} \\ R_i \\ \\ R_j \\ \\ \end{pmatrix} + \det\begin{pmatrix} \\ R_j \\ \\ R_i \\ \\ \end{pmatrix} + \det\begin{pmatrix} \\ R_j \\ \\ R_j \\ \\ \end{pmatrix}$$

using multilinearity on rows $R_i$ and $R_j$; note that the first and last expressions on the RHS are zero, since two rows are equal. So suppose that rows $i$ and $j$ of $A$ are identical. To show that $\det(A) = 0$, we expand along row $i$ first to obtain:

$$\det(A) = \sum_{k=1}^{n} (-1)^{i+k} a_{ik} \det(A[i|k])$$

and then we expand each minor $A[i|k]$ along the row that corresponds to the $j$-th row of $A$. Note that we end up with $n(n-1)$ terms; polynomially many in the size of $A$. Since row $i$ is identical to the row $j$, we can pair each term with its negation; hence the result is zero, so $\det(A) = 0$. □

The following lemma shows that LAP proves, from the axiomatic definition of det, that our definition of the adjoint is equivalent to the usual definition of the adjoint as the transpose of the matrix of cofactors.

**Lemma 6.3.2** LAP proves that $\text{adj}(A) = ((-1)^{i+j} \det(A[j|i]))_{ij}$, i.e. that $\text{adj}(A)$ is the transpose of the matrix of cofactors of $A$, from the axiomatic definition of det.

*Proof.* Consider the following matrix:

$$C = \begin{pmatrix} 0 & e_i^t \\ e_j & A \end{pmatrix}$$

where $e_i$ is a column vector with zeros everywhere except in the $i$-th position where it has a 1, and $e_j$ is a row vector with a 1 in the $j$-th position. By Lemma 5.1.3, we have that:

$$\det(C) = -e_i^t \text{adj}(A) e_j = \ (i,j)\text{-th entry of } -\text{adj}(A)$$

On the other hand, from alternation on $C$, we have that $\det(C) = (-1)^{i+j+1} \det(A[j|i])$. To see this, note that we need $(j+1)$ transpositions to bring the $j$-th row of $A$ to the first row in the matrix $C$, to obtain the following matrix:

$$C' = \begin{pmatrix} 1 & A_j \\ 0 & e_i^t \\ 0 & A[j|-] \end{pmatrix}$$

where $A_j$ denotes the $j$-th row of $A$, and $A[j|-]$ denotes $A$ with the $j$-th row deleted. Then, by Lemma 5.1.3, we have:

$$\det(C') = \det \begin{pmatrix} e_i^t \\ A[j|-] \end{pmatrix}$$

and now with $i$ transpositions, we bring the $i$-th column of $\begin{pmatrix} e_i^t \\ A[j|-] \end{pmatrix}$ to the first column, to obtain: $\begin{pmatrix} 1 & 0 \\ 0 & A[j|i] \end{pmatrix}$. Therefore, $\det(C') = (-1)^i \det(A[j|i])$ finishing the proof of the Lemma. □

Since by Lemma 6.3.1 the axiomatic definition of det follows from the cofactor expansion formula, we have the following Corollary to Lemma 6.3.2:

**Corollary 6.3.1** LAP proves that $\mathrm{adj}(A) = ((-1)^{i+j} \det(A[j|i]))_{ij}$ from the cofactor expansion formula.

Note that $p_A(A) = 0$ can also be stated as $A\mathrm{adj}(A) = \det(A)I$, using our definitions of the adjoint and the determinant (see page 51). Thus, the following Lemma shows that LAP proves the C-H Theorem from the cofactor expansion formula.

**Lemma 6.3.3** LAP proves $A\mathrm{adj}(A) = \mathrm{adj}(A)A = \det(A)I$ from the cofactor expansion formula.

*Proof.* We show first that $A\mathrm{adj}(A) = \det(A)I$. The $(i,j)$-th entry of $A\mathrm{adj}(A)$ is by Corollary 6.3.1 equal to:

$$a_{i1}(-1)^{j+1} \det(A[j|1]) + \cdots + a_{in}(-1)^{j+n} \det(A[j|n]) \tag{6.15}$$

If $i = j$, this is the cofactor expansion along the $i$-th row. Suppose now that $i \neq j$. Let $A'$ be the matrix $A$ with the $j$-th row replaced by the $i$-th row. Then, by alternation (which we have by Lemma 6.3.1), $\det(A') = 0$. Now, (6.15) is the cofactor expansion of $A'$ along the $j$-th row, and therefore, it is 0. This proves that $A\mathrm{adj}(A) = \det(A)I$, and by definition of the adjoint, $\mathrm{adj}(A)A = A\mathrm{adj}(A)$, so we are done. □

## 6.4 The multiplicativity of the determinant

The multiplicativity of the determinant is the property: $\det(AB) = \det(A)\det(B)$. This turns out to be a very strong property, from which all other properties (including the Cayley-Hamilton Theorem) follow readily in LAP.

**Lemma 6.4.1** LAP proves that the multiplicativity of the determinant implies the C-H Theorem.

*Proof.* From the multiplicativity of the determinant we have that (by Lemma 5.1.6) $\det(I_{12}AI_{12}) = \det(I_1)\det(A)\det(I_1) = \det(A)$ for any matrix $A$. Suppose we want to prove the C-H Theorem for some $n \times n$ matrix $M$. Define $A$ as follows:

$$A = \begin{pmatrix} a & b & R \\ c & d & P \\ S & Q & M \end{pmatrix} = \begin{pmatrix} 0 & 0 & e_i^t \\ 0 & 0 & 0 \\ e_j & 0 & M \end{pmatrix}$$

Let $C_1C_2C_3\cdots C_{n+2}$ be the char poly of $A$ (and $C_3\cdots C_{n+2}$ the char poly of $M$). From Berkowitz's algorithm it is easy to see that for $A$ defined this way the bottom row of $C_1C_2$ is given by:

$$e_i^tM^ne_j \quad e_i^tM^{n-1}e_j \quad \ldots \quad e_i^tIe_j \quad 0$$

so the bottom row of $C_1C_2C_3\cdots C_{n+2}$ is simply $e_i^tp(M)e_j$ where $p$ is the char poly of $M$.

On the other hand, using $\det(A) = \det(I_{12}AI_{12})$ and Berkowitz's algorithm, we have that:

$$\det(A) = \det\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & e_i^t \\ 0 & e_j & M \end{pmatrix} = 0$$

so that $e_i^tp(M)e_j = 0$, and since we can choose any $i, j$, we have that $p(M) = 0$. □

What about the other direction? That is, can we prove the following implication in LAP: C-H Theorem $\Longrightarrow$ Multiplicativity of the determinant? The answer is "yes", *if* LAP can prove the following determinant identity:

$$\det(A) = 0 \rightarrow AB \neq I \tag{6.16}$$

That is, LAP can prove the multiplicativity of the determinant from the C-H Theorem and (6.16). We suspect, however, that LAP can prove (6.16) from the C-H Theorem, so that the C-H Theorem is enough to prove multiplicativity. At this point, we do *not* have a LAP proof of (6.16)[1] from the C-H Theorem.

**Lemma 6.4.2** LAP can prove the multiplicativity of the determinant from the C-H Theorem and the property given by (6.16).

[1]However we have a feasible proof of (6.16) based on Gaussian Elimination—see Section 8.3.2. Therefore, since we have a LAP proof of multiplicativity of det from the C-H Theorem and from (6.16)—see Lemma 6.4.2, it follows that we have a feasible proof of multiplicativity of det from the C-H Theorem; Open Problem 9.4: is there a direct LAP proof of multiplicativity of det from the C-H Theorem?

*Proof.* We prove the Lemma by induction on the size of the matrices; so assume that $A, B$ are square $n \times n$ matrices. Since we assume the Cayley-Hamilton Theorem, by the results in the previous sections we also have at our disposal the cofactor expansion and the axiomatic definition of the determinant.

Suppose first that the determinants of all the minors of $A$ (or $B$) are zero. Then, using the cofactor expansion we obtain $\det(A) = 0$. We now want to show that $\det(AB) = 0$ as well.

Suppose that $\det(AB) \neq 0$. Then, by the C-H Theorem, $AB$ has an inverse $C$, i.e., $(AB)C = I$. But then $A(BC) = I$, so $A$ is invertible, contrary to (6.16). Therefore, $\det(AB) = 0$, so that in this case $\det(A)\det(B) = \det(AB)$.

Suppose now that both $A$ and $B$ have a minor whose determinant is not zero. We can assume that it is the principal submatrix whose determinant is not zero (as $A$ and $I_{1i}AI_{1j}$ have the same determinant, so we can bring any non-singular minor to be the principal subminor). So assume that $M_A, M_B$ are non-singular, where:

$$A = \begin{pmatrix} a & R_A \\ S_A & M_A \end{pmatrix} \qquad B = \begin{pmatrix} b & R_B \\ S_B & M_B \end{pmatrix}$$

By the Induction Hypothesis we know that $\det(M_A M_B) = \det(M_A)\det(M_B)$. Also note that:

$$AB = \begin{pmatrix} ab + R_A S_B & aR_B + R_A M_B \\ bS_A + M_A S_B & S_A R_B + M_A M_B \end{pmatrix}$$

Now using Berkowitz's algorithm:

$$\det(A)\det(B) = (a\det(M_A) - R_A\mathrm{adj}(M_A)S_A)(b\det(M_B) - R_B\mathrm{adj}(M_B)S_B) \tag{6.17}$$

We want to show that $\det(AB)$ is equal to (6.17). Again, using Berkowitz's algorithm:

$$\begin{aligned} \det(AB) = {} & (ab + R_A S_B)\det(S_A R_B + M_A M_B) \\ & - (aR_B + R_A M_B)\mathrm{adj}(S_A R_B + M_A M_B)(bS_A + M_A S_B) \end{aligned} \tag{6.18}$$

We now show that the right hand sides of (6.17) and (6.18) are equal.

By Lemma 6.4.3:

$$\det(S_A R_B + M_A M_B) = \det(M_A M_B) + R_B\mathrm{adj}(M_A M_B)S_A \tag{6.19}$$

Using the IH, $\det(M_A M_B) = \det(M_A)\det(M_B)$, and using Lemma 6.3.3 and the fact that $\det(M_A) \neq 0$ and $\det(M_B) \neq 0$ we obtain: $\mathrm{adj}(M_A M_B) = \mathrm{adj}(M_B)\mathrm{adj}(M_A)$. To

see this note that by the C-H Theorem $(M_AM_B)\text{adj}(M_AM_B) = \det(M_AM_B)I$. We now multiply both sides of this equation by $\text{adj}(M_A)$ to obtain, by the C-H Theorem again, $\det(M_A)M_B\text{adj}(M_AM_B) = \det(M_AM_B)\text{adj}(M_A)$. Now multiply both sides by $\text{adj}(M_B)$ to obtain:

$$\det(M_A)\det(M_B)\text{adj}(M_AM_B) = \det(M_AM_B)\text{adj}(M_B)\text{adj}(M_A)$$

Since $\det(M_AM_B) = \det(M_A)\det(M_B)$, and $\det(M_A)\det(M_B) \neq 0$, we obtain our result.

Therefore, from (6.19) we obtain:

$$\det(S_AR_B + M_AM_B) = \det(M_A)\det(M_B) + R_B\text{adj}(M_B)\text{adj}(M_A)S_A \tag{6.19$'$}$$

Using Lemma 6.4.4 and $\text{adj}(M_AM_B) = \text{adj}(M_B)\text{adj}(M_A)$, we obtain:

$$\begin{aligned} R_B\text{adj}(S_AR_B + M_AM_B) &= R_B\text{adj}(M_B)\text{adj}(M_A) \\ \text{adj}(S_AR_B + M_AM_B)S_A &= \text{adj}(M_B)\text{adj}(M_A)S_A \end{aligned} \tag{6.20}$$

Finally, we have to prove the following identity:

$$\begin{aligned} R_AM_B\text{adj}(S_AR_B+M_AM_B)M_AS_B = \\ R_AS_B\det(M_A)\det(M_B) - R_B\text{adj}(M_B)S_BR_A\text{adj}(M_A)S_A \\ + (R_AS_B)R_B\text{adj}(M_B)\text{adj}(M_A)S_A \end{aligned} \tag{6.21}$$

First of all, by Lemma 6.3.3 we have:

$$(S_AR_B + M_AM_B)\text{adj}(S_AR_B + M_AM_B) = \det(S_AR_B + M_AM_B)$$

Using Lemmas 6.4.3 and 6.4.4, we get:

$$S_AR_B\text{adj}(M_AM_B) + M_AM_B\text{adj}(S_AR_B + M_AM_B) = (\det(M_AM_B) + R_B\text{adj}(M_AM_B)S_A)I$$

We have already shown above that $\text{adj}(M_AM_B) = \text{adj}(M_B)\text{adj}(M_A)$ using our Induction Hypothesis: $\det(M_AM_B) = \det(M_A)\det(M_B)$. So, if we multiply both sides of the above equation by $\text{adj}(M_A)$ on the left, and by $M_A$ on the right, we obtain:

$$\begin{aligned} \text{adj}(M_A)S_AR_B\text{adj}(M_B)\det(M_A) + \det(M_A)M_B\text{adj}(S_AR_B + M_AM_B)M_A = \\ \det(M_A)(\det(M_A)\det(M_B) + R_B\text{adj}(M_B)\text{adj}(M_A)S_A)I \end{aligned}$$

Since by assumption $\det(M_A) \neq 0$, we can divide both sides of the equation by $\det(M_A)$ to obtain:

$$\begin{aligned} \text{adj}(M_A)S_AR_B\text{adj}(M_B) + M_B\text{adj}(S_AR_B + M_AM_B)M_A = \\ (\det(M_A)\det(M_B) + R_B\text{adj}(M_B)\text{adj}(M_A)S_A)I \end{aligned}$$

If we now multiply both sides of the above equation, by $R_A$ on the left, and by $S_B$ on the right, we obtain (6.21) as desired.

We now substitute (6.19′), (6.20), and (6.21) into (6.18), and we obtain that the right hand side of (6.18) is equal to the right hand side of (6.17), and we are done. □

**Lemma 6.4.3** LAP proves, from the axiomatic definition of det, that:

$$\det(SR + M) = \det(M) + R\text{adj}(M)S \tag{6.22}$$

*Proof.* Consider the matrix:

$$C = \left(\begin{array}{c|c} 1 & -R \\ \hline S & M \end{array}\right)$$

Using Berkowitz's algorithm (the definition of det given in 4.8), it follows that:

$$\det(C) = \det(M) + R\text{adj}(M)S$$

We can add multiples of the first row of $C$ to the remaining rows of $C$, to clear out $S$, and obtain:

$$C' = \left(\begin{array}{c|c} 1 & -R \\ \hline 0 & SR + M \end{array}\right)$$

Using the axiomatic definition of det, we can conclude that $\det(C') = \det(C)$, and using (4.8) on $C'$ we obtain:

$$\det(C') = \det(SR + M)$$

and hence the Lemma follows. □

**Lemma 6.4.4** LAP proves, from the Cayley-Hamilton Theorem, that:

$$R\text{adj}(SR + M) = R\text{adj}(M)$$
$$\text{adj}(SR + M)S = \text{adj}(M)S$$

*Proof.* By Lemma 6.3.2 we know that $\mathrm{adj}(A)$ is the transpose of the matrix of cofactors of $A$. From this we can deduce the following identity:

$$\mathrm{adj}(A) = \begin{pmatrix} \det(M) & -R\mathrm{adj}(M) \\ -\mathrm{adj}(M)S & (1+a_{11})\mathrm{adj}(M) - \mathrm{adj}(SR+M) \end{pmatrix} \tag{6.23}$$

To see this we are going to consider the four standard submatrices. First of all, the $(1,1)$ entry of $\mathrm{adj}(A)$ is the determinant of the principal minor of $A$ times $(-1)^{1+1}$, i.e. $\det(M)$. The remaining entries along the first row are given by $(-1)^{1+i}\det(A[i|1])$, for $2 \leq i \leq n$. Note that for $2 \leq i \leq n$, $A[i|1]$ is given by:

$$\begin{pmatrix} R \\ M[i|-] \end{pmatrix} \tag{6.24}$$

where $M[i|-]$ denotes $M$ without the $i$-th row. To compute the determinant of the matrix given by (6.24) expand along the first row to obtain: $\sum_{j=1}^{n-1} r_j(-1)^{i+j}\det(M[i|j])$. This gives us $-R\mathrm{adj}(M)$ as desired. In the same way we can show that the entries in the first column below $(1,1)$ are given by $-\mathrm{adj}(M)S$.

We now show that the principal submatrix is given by $(1+a_{11})\mathrm{adj}(M) - \mathrm{adj}(SR+M)$. To see this first note that $(SR+M)[i|j] = S[i]R[j] + M[i|j]$, where $S[i], R[j]$ denote $S, R$ without the $i$-th row and $j$-th column, respectively. Now using Lemma 6.4.3 we have that $\det((SR+M)[i|j]) = \det(M[i|j]) + R[j]\mathrm{adj}(M[i|j])S[i]$. The $(i+1, j+1)$ entry of $\mathrm{adj}(A)^t$, $1 \leq i, j < n$, is given by:

$$(-1)^{i+j}(a_{11}\det(M[i|j]) - R[j]\mathrm{adj}(M[i|j])S[i])$$

as can be seen from Figure 6.4.

Therefore, the $(i+1, j+1)$ entry of $\mathrm{adj}(A)^t$ is given by:

$$(-1)^{i+j}(a_{11}\det(M[i|j]) + \det(M[i|j]) - \det((SR+M)[i|j]))$$

and we are done.

By Lemma 6.3.3 we know that:

$$\begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix} \begin{pmatrix} \det(M) & -R\mathrm{adj}(M) \\ -\mathrm{adj}(M)S & (1+a_{11})\mathrm{adj}(M) - \mathrm{adj}(SR+M) \end{pmatrix} = \det(A)I$$

In particular this means that:

$$-a_{11}R\mathrm{adj}(M) + R(1+a_{11})\mathrm{adj}(M) - R\mathrm{adj}(SR+M) = 0$$

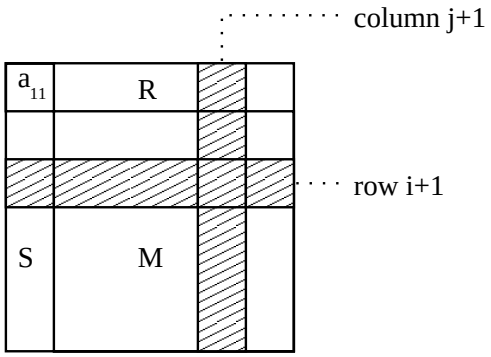



Figure 6.4: Showing that $\text{adj}(A)[1|1] = (1 + a_{11})\text{adj}(M) - \text{adj}(SR + M)$

and from this it follows that $R\text{adj}(SR + M) = R\text{adj}(M)$. Similarly, we can prove the second identity. □

**Corollary 6.4.1** LAP proves, from the C-H Theorem, the anti-multiplicativity of the adjoint for non-singular matrices, i.e.,

$$\det(A) \neq 0, \det(B) \neq 0 \rightarrow \text{adj}(AB) = \text{adj}(B)\text{adj}(A)$$

*Proof.* The proof is given in the proof of Lemma 6.4.2. □

As a closing remark, note that our proof of multiplicativity of the determinant from the Cayley-Hamilton Theorem is quite long and complicated. Of course, conceptually the simplest proof is based on the cofactor expansion, but then we must prove the induction hypothesis for too many matrices, rendering the proof infeasible. See [DF91, page 364, Theorem 25] for a simple, yet infeasible, proof of multiplicativity of determinant based on the Lagrange expansion.

# Chapter 7

# Translations

In this chapter we will show how to translate theorems of LA and LAP into families of tautologies with short propositional proofs. These translations are a potential tool for proving independence results. For example, if we can prove that $AB = I \rightarrow BA = I$ does not have efficient proofs in bounded-depth Frege with mod 2 gates, then it will follow that $AB = I \rightarrow BA = I$ is not a theorem of LA. See Section 9.1 for more details.

We will show that the theorems of LA can be translated into families of propositional tautologies which have poly-bounded PK derivations (see Definition 2.4.2 for the system PK). In fact, we prove a tighter result. We are going to show that the theorems of LA (when the underlying field is $\mathbb{Z}_p$, $p$ a prime) translate into propositional tautologies with short bounded-depth PK[$p$] derivations, where PK[$p$] is PK with $\mathtt{MOD}_{p,i}$ gates (i.e., modular gates, for counting modulo the prime $p$). In Lemma 7.1.1 we show that PK can $p$-simulate PK[$a$] for any $a$. We will also show that the theorems of LAP have quasi-poly-bounded ($O(2^{\log^2 n})$) PK derivations.

The main parameter in the translations is the size of matrices, and the underlying field plays a minor technical role. In section 7.2, we give a detailed translation for the case where the underlying field is $\mathbb{Z}_2$. This is the simplest case, since the field elements $\mathbb{Z}_2 = \{0, 1\}$ translate directly into boolean values $\{\mathtt{T}, \mathtt{F}\}$. We consider very briefly the fields $\mathbb{Z}_p$, for a general prime $p$, and $\mathbb{Q}$, in Section 7.3.

Since our theories LA and LAP are field independent, it is no surprise that all the ideas necessary for the translation occur already in the case $\mathbb{Z}_2$. Translating over $\mathbb{Z}_p$, $p > 2$, and over $\mathbb{Q}$ is just a question of finding some representation for the field elements in terms of several boolean variables. Thus, we concentrate on the technically simplest case: $\mathbb{Z}_2$.

## 7.1 The propositional proof system PK$[a]$

The definition and axiomatization of modular connectives that we present in this section come from [Kra95, Chapter 12.6].

**Definition 7.1.1** Let $a \geq 2$ be fixed and let $i$ be in $\{0, \ldots, a-1\}$. Then, $\texttt{MOD}_{a,i}$ is a propositional connective of unbounded arity such that:

$$\texttt{MOD}_{a,i}(\Gamma) \text{ is true} \quad \text{iff} \quad |\{j : A_j \text{ is true}\}| \pmod a = i$$

where $\Gamma$ is a cedent of formulas, that is, $\Gamma = A_1, \ldots, A_k$, and the $A_i$'s are formulas, and $k \geq 0$. See Table 7.1 for the propositional axioms for $\texttt{MOD}_{a,i}$.

**Axiom-1** $\rightarrow \texttt{MOD}_{a,0}(\emptyset)$

**Axiom-2** $\rightarrow \neg\texttt{MOD}_{a,i}(\emptyset)$
for $i$ in $\{1, \ldots, a-1\}$

**Axiom-3** $\rightarrow \texttt{MOD}_{a,i}(\Gamma, A) \equiv \left[(\texttt{MOD}_{a,i}(\Gamma) \wedge \neg A) \vee (\texttt{MOD}_{a,i-1}(\Gamma) \wedge A)\right]$
for $i$ in $\{0, \ldots, a-1\}$, where $i-1$ means $i-1 \pmod a$, and where, as above, $\Gamma$ is a (possibly empty) cedent of formulas.

Table 7.1: Axioms for $\texttt{MOD}_{a,i}$

**Definition 7.1.2** The system PK$[a]$ denotes the system PK whose language is extended by the connectives $\texttt{MOD}_{a,i}$ for $i$ in $\{0, \ldots, a-1\}$, and that is augmented by the preceding three axiom schemas.

**Lemma 7.1.1** PK $p$-simulates PK$[a]$ for any $a$. In other words, there is a polytime function that given a PK$[a]$ derivation $\pi$ of a tautology $\tau$ (without modular connectives), outputs a PK proof $\pi'$ of $\tau$.

*Proof.* We are just going to prove this Lemma in the case $a = 2$. The general case, for $a > 2$, can be proven in a similar way. So, in the simulation we want to translate $\texttt{MOD}_{2,1}$ into a boolean formula over over $\{\wedge, \vee, \neg\}$ with a polynomial increase in size (polynomial in the size of the arguments of $\texttt{MOD}_{2,1}$).

Note that the result for $\texttt{MOD}_{2,0}$ will follow since $\texttt{MOD}_{2,0} = \overline{\texttt{MOD}}_{2,1}$.

First note that $\texttt{MOD}_{2,1}(x_1, x_2, \ldots, x_n)$ can be expressed as a balanced binary tree of XOR gates, of depth $\log n$, and size $n$. Now note:

$$\texttt{XOR}(x, y) \equiv (x \wedge \neg y) \vee (\neg x \wedge y)$$
$$\overline{\texttt{XOR}}(x, y) \equiv (x \wedge y) \vee (\neg x \wedge \neg y)$$

So each XOR gate can be replaced by such a formula. Since for a given XOR gate in the above mentioned balanced binary tree we need to provide the input and its negation, in each case we need to compute XOR and $\overline{\texttt{XOR}}$.

In short we obtain a circuit with gates $\{\wedge, \vee, \neg\}$ of depth $O(\log n)$ and size $O(n)$, and hence we get the corresponding poly-size boolean formula. □

The following definitions come from [Kra95, Chapter 4.3].

**Definition 7.1.3** The *logical depth* of a formula $A$, denoted $\text{ldp}(A)$, is the maximum nesting of connectives in $A$. More precisely:

1. $\text{ldp}(\texttt{F}) = \text{ldp}(\texttt{T}) = \text{ldp}(a) = 0$, for any atomic variable $a$.
2. $\text{ldp}(\neg A) = 1 + \text{ldp}(A)$, for any formula $A$.
3. $\text{ldp}(A \circ B) = 1 + \max\{\text{ldp}(A), \text{ldp}(B)\}$ for any formulas $A, B$ and a connective $\circ = \wedge, \vee$.
4. $\text{ldp}(\texttt{MOD}_{a,i}(A_1, A_2, \ldots, A_k)) = 1 + \max\{\text{ldp}(A_1), \text{ldp}(A_2), \ldots, \text{ldp}(A_k)\}$ for $i$ in the set $\{0, \ldots, a-1\}$.

**Definition 7.1.4** The *depth* of a formula $A$, denoted $\text{dp}(A)$, is the maximum number of alternations of connectives in $A$. More precisely:

1. $\text{dp}(A) = 0$ iff $\text{ldp}(A) = 0$.
2. $\text{dp}(\neg\neg\cdots\neg A) = \text{dp}(A)$ if the number of negations is even, and it is $1 + \text{dp}(A)$ otherwise (we are assuming that the outermost connective of $A$ is *not* "$\neg$").
3. Fix $\circ$ to be $\wedge$ or $\vee$. Then:
   (a) $\text{dp}(A \circ B) = \max\{\text{dp}(A), \text{dp}(B)\}$ if the outermost connective in both $A$ and $B$ is $\circ$.

(b) $\mathrm{dp}(A \circ B) = 1 + \max\{\mathrm{dp}(A), \mathrm{dp}(B)\}$ if neither $A$ nor $B$ has $\circ$ as the outermost connective.

(c) $\mathrm{dp}(A \circ B) = \max\{1 + \mathrm{dp}(A), \mathrm{dp}(B)\}$ if $\circ$ is the outermost connective of $B$ but it is *not* the outermost connective of $A$.

(d) $\mathrm{dp}(A \circ B) = \max\{\mathrm{dp}(A), 1 + \mathrm{dp}(B)\}$ if $\circ$ is the outermost connective of $A$ but it is *not* the outermost connective of $B$.

4. Consider $\phi = \mathtt{MOD}_{a,i}(A_1, A_2, \ldots, A_{l+k})$ for $i$ in $\{0, 1, \ldots, a-1\}$. Let $\{A_1, \ldots, A_l\}$ be the set of $A_j$'s whose outermost connective is not $\mathtt{MOD}_{a,i}$, and let $\{A_{l+1}, \ldots, A_{l+k}\}$ be the set of $A_j$'s whose outermost connective is $\mathtt{MOD}_{a,i}$. Then:

$$\mathrm{dp}(\phi) = \max\{1 + \mathrm{dp}(A_1), \ldots, 1 + \mathrm{dp}(A_l), \mathrm{dp}(A_{l+1}), \ldots, \mathrm{dp}(A_{l+k})\}$$

If $l$ or $k$ are zero, then the corresponding sets are empty. However $l + k > 0$, that is, they are not both zero.

**Definition 7.1.5** Let $P$ be a PK[$a$] derivation. Then, the logical depth of $P$, denoted $\mathrm{ldp}(P)$, is the maximal logical depth over all formulas in $P$. Similarly, the depth of $P$, denoted $\mathrm{dp}(P)$, is the maximal depth over all formulas in $P$.

**Definition 7.1.6** If $A$ is a propositional formula, we define size($A$) to be the number of variables, connectives and constants (`T` or `F`) in $A$. If $P$ is a PK[$a$] derivation, we define size($P$) to be the sum of the sizes of all the formulas in $P$.

## 7.2 Translating theorems of LA over $\mathbb{Z}_2$

In this section we translate the theorems of LA over the standard model $\mathcal{S}_{\mathbb{Z}_2}$, where the indices are in $\mathbb{N}$ and the field elements are in $\mathbb{Z}_2$, into families of propositional tautologies with short PK[2] derivations of bounded depth.

The main idea is the following: given a formula $\alpha$ over the language $\mathcal{L}_{\mathrm{LA}}$, we translate it into a family of "small" propositional formulas $\|\alpha\|_\sigma$ (where $\sigma$ indexes the family and assigns values to all terms of type index in $\alpha$), such that the following holds true: if $\alpha$ is true in the standard model, then $\|\alpha\|_\sigma$ is a propositional tautology for all $\sigma$, and furthermore, if $\alpha$ is a theorem of LA, then, for any $\sigma$, $\|\alpha\|_\sigma$ has a "short" PK[2] derivation of depth bounded by a constant.

The procedure for the translation is given in section 7.2.2, and the proof of correctness of the procedure is given in section 7.2.3.

### 7.2.1 Preliminaries

We are concerned with standard models (see section 2.2.5) where $\mathbb{F} = \mathbb{Z}_2$, i.e. with $\mathcal{S}_{\mathbb{Z}_2}$. We say that $\alpha$ is *true*, if it is true in $\mathcal{S}_{\mathbb{Z}_2}$, i.e. if $\mathcal{S}_{\mathbb{Z}_2} \vDash \alpha[\tau]$ for all object assignments $\tau$.

Given a formula $\alpha$ over $\mathcal{L}_{\mathrm{LA}}$, we let $\sigma$ be a partial object assignment of natural numbers to all the free index variables in $\alpha$, and all terms of the form $\mathtt{r}(A), \mathtt{c}(A)$, for all $A$. We define the norm of $\sigma$, denoted by $|\sigma|$, to be the largest value assignment of $\sigma$, i.e. $|\sigma| = \max\{\sigma(i), \sigma(j), \ldots, \sigma(\mathtt{r}(A)), \sigma(\mathtt{c}(A)), \ldots\}$. We define $\sigma(p_1/i_1)\cdots(p_n/i_n)$ to be the same as $\sigma$, except now the free index variables $i_1, \ldots, i_n$ are assigned the values $p_1, \ldots, p_n$, respectively.

We also let every field variable $a$ become a boolean variable $a$ (recall that the underlying field is $\mathbb{Z}_2$, so there is a direct and natural correspondence between field elements and boolean values). With each matrix variable $A$ we associate the following boolean variables: $A_{11}, A_{12}, \ldots, A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))}$. Finally, all propositional formulas are over the connectives $\{\wedge, \vee, \neg, \mathtt{MOD}_{2,0}, \mathtt{MOD}_{2,1}, \mathtt{T}, \mathtt{F}\}$. Note that $A \supset B$ is an abbreviation for $\neg A \vee B$, and $A \equiv B$ is an abbreviation for $(\neg A \vee B) \wedge (\neg B \vee A)$.

### 7.2.2 Procedure for the translation

We are given as input a sequent $S$ over $\mathcal{L}_{\mathrm{LA}}$, and a partial object assignment $\sigma$ (of natural numbers to all the free index variables in $S$, and to all terms $\mathtt{r}(A), \mathtt{c}(A)$, for all the matrix variables in $S$). The output is the propositional sequent $\|S\|_\sigma$.

A few remarks before we present the procedure: sequents are translated by translating all formulas in the antecedent and succedent. Thus $\|\alpha_1, \ldots, \alpha_k \to \beta_1, \ldots, \beta_l\|_\sigma$ becomes:

$$\|\alpha_1\|_\sigma, \ldots, \|\alpha_k\|_\sigma \to \|\beta_1\|_\sigma, \ldots, \|\beta_l\|_\sigma$$

so in the procedure below we only show how to translate formulas.

All terms of type index are translated to natural numbers, that is, if $m$ is a term of type index, then $\|m\|_\sigma \in \mathbb{N}$. Therefore, all atomic formulas of the form $m =_{\mathrm{index}} n$ and $m \leq_{\mathrm{index}} n$ are translated directly into $\mathtt{T}$ or $\mathtt{F}$.

All terms of type field are translated into propositional formulas, i.e. $\|t\|_\sigma$ is a propositional formula, and all formulas of the form $t =_{\mathrm{field}} u$ are translated into $\|t\|_\sigma \equiv \|u\|_\sigma$.

Finally, all formulas of the form $U =_{\text{matrix}} T$ are translated into a conjunction expressing that $T$ and $U$ are equal entry by entry.

We now define the recursive procedure for the translation, which takes as input a formula $\alpha$ over $\mathcal{L}_{\text{LA}}$ and $\sigma$, and outputs a propositional formula $\|\alpha\|_\sigma$.

If $\alpha$ is given by one of the following:

$$\alpha_1 \wedge \alpha_2 \qquad \alpha_1 \vee \alpha_2 \qquad \neg\alpha_1$$

then $\|\alpha\|_\sigma$ is given by one of the following:

$$\|\alpha_1\|_\sigma \wedge \|\alpha_2\|_\sigma \qquad \|\alpha_1\|_\sigma \vee \|\alpha_2\|_\sigma \qquad \neg\|\alpha_1\|_\sigma$$

respectively. Suppose now that $\alpha$ is an atomic formula. Then, $\alpha$ is one of the following:

$$m = n \qquad m \leq n \qquad t = u \qquad T = U$$

Here is what we do in each case:

$$\|m \leq n\|_\sigma \longmapsto \begin{cases} \mathtt{T} & \text{if } \|m\|_\sigma \leq \|n\|_\sigma \\ \mathtt{F} & \text{otherwise} \end{cases}$$

$$\|m = n\|_\sigma \longmapsto \begin{cases} \mathtt{T} & \text{if } \|m\|_\sigma = \|n\|_\sigma \\ \mathtt{F} & \text{otherwise} \end{cases}$$

$$\|t = u\|_\sigma \longmapsto (\|t\|_\sigma \equiv \|u\|_\sigma)$$

The case $\|T = U\|_\sigma$ is more complicated. If $T$ and $U$ do not have compatible sizes, that is, if $\|\mathtt{r}(T)\|_\sigma \neq \|\mathtt{r}(U)\|_\sigma$ or $\|\mathtt{c}(T)\|_\sigma \neq \|\mathtt{c}(U)\|_\sigma$, then:

$$\|T = U\|_\sigma \longmapsto \mathtt{F}$$

Suppose now that $T$ and $U$ have compatible sizes, and let $r, c$ be defined as follows:

$$r := \|\mathtt{r}(T)\|_\sigma = \|\mathtt{r}(U)\|_\sigma$$
$$c := \|\mathtt{c}(T)\|_\sigma = \|\mathtt{c}(U)\|_\sigma$$

Assume that $i, j$ are index variables that do not occur free in $T$ or $U$. Then:

$$\|T = U\|_\sigma \longmapsto \bigwedge_{1 \leq p \leq r, 1 \leq q \leq c} (\|\mathtt{e}(T, i, j)\|_{\sigma(p/i)(q/j)} \equiv \|\mathtt{e}(U, i, j)\|_{\sigma(p/i)(q/j)})$$

All that is left to do is to show how to translate terms of type index and field. We give a recursive (sub)procedure for this.

**Base Case:**

$$
\begin{aligned}
\|0_{\text{index}}\|_\sigma &\longmapsto 0 \in \mathbb{N} \\
\|1_{\text{index}}\|_\sigma &\longmapsto 1 \in \mathbb{N} \\
\|i\|_\sigma &\longmapsto \sigma(i) \in \mathbb{N} \\
\|0_{\text{field}}\|_\sigma &\longmapsto \texttt{F} \\
\|1_{\text{field}}\|_\sigma &\longmapsto \texttt{T} \\
\|a\|_\sigma &\longmapsto a
\end{aligned}
$$

Note that the "$a$" on the LHS is a field elements, and the "$a$" on the RHS is a boolean variable.

**Recursive Step:** Suppose $m, n$ are terms of type index. Then:

$$
\begin{aligned}
\|m +_{\text{index}} n\|_\sigma &\longmapsto \|m\|_\sigma + \|n\|_\sigma \\
\|m -_{\text{index}} n\|_\sigma &\longmapsto \max\{\|m\|_\sigma - \|n\|_\sigma, 0\} \\
\|m *_{\text{index}} n\|_\sigma &\longmapsto \|m\|_\sigma \cdot \|n\|_\sigma \\
\|\texttt{div}(m,n)\|_\sigma &\longmapsto \left\lfloor \frac{\|m\|_\sigma}{\|n\|_\sigma} \right\rfloor \\
\|\texttt{rem}(m,n)\|_\sigma &\longmapsto \|m\|_\sigma - \left\lfloor \frac{\|m\|_\sigma}{\|n\|_\sigma} \right\rfloor \cdot \|n\|_\sigma \\
\|\text{cond}(\beta,m,n)\|_\sigma &\longmapsto \begin{cases} \|m\|_\sigma & \text{if } \|\beta\|_\sigma \\ \|n\|_\sigma & \text{otherwise} \end{cases} \qquad (*)
\end{aligned}
$$

where on the right we have the usual $+, -, \cdot$ of $\mathbb{N}$.

Note that in the last rule (the rule marked by $(*)$), $\|\beta\|_\sigma$ is either `F` or `T` because, by definition of $\text{cond}(\beta, m, n)$ (item 9 in section 2.2.1), all the atomic subformulas of $\beta$ are of type index.

Suppose $t, u$ are terms of type field. Then:

$$\begin{aligned}
\|t +_{\text{field}} u\|_\sigma &\longmapsto \mathtt{MOD}_{2,1}(\|t\|_\sigma, \|u\|_\sigma) \\
\|t *_{\text{field}} u\|_\sigma &\longmapsto \|t\|_\sigma \wedge \|u\|_\sigma \\
\|-t\|_\sigma &\longmapsto \|t\|_\sigma \\
\|t^{-1}\|_\sigma &\longmapsto \|t\|_\sigma \\
\|\mathrm{cond}(\beta, t, u)\|_\sigma &\longmapsto \begin{cases} \|t\|_\sigma & \text{if } \|\beta\|_\sigma \\ \|u\|_\sigma & \text{otherwise} \end{cases} \qquad (**)
\end{aligned}$$

Again, note that $\|\beta\|_\sigma$ in rule $(**)$ is either $\mathtt{F}$ or $\mathtt{T}$.

We translate $\mathtt{r}(T), \mathtt{c}(T)$ as follows:

$$\begin{aligned}
\|\mathtt{r}(A)\|_\sigma &\longmapsto \sigma(\mathtt{r}(A)) \\
\|\mathtt{c}(A)\|_\sigma &\longmapsto \sigma(\mathtt{c}(A)) \\
\|\mathtt{r}(\lambda ij\langle m, n, t\rangle)\|_\sigma &\longmapsto \|m\|_\sigma \\
\|\mathtt{c}(\lambda ij\langle m, n, t\rangle)\|_\sigma &\longmapsto \|n\|_\sigma
\end{aligned}$$

With each matrix variable $A$ we associate the following set of $\sigma(\mathtt{r}(A))\cdot\sigma(\mathtt{c}(A))$ boolean variables: $A_{11}, A_{12}, \ldots, A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))}$. Thus, we translate $\mathtt{e}(A, m, n)$ as follows:

$$\|\mathtt{e}(A, m, n)\|_\sigma \longmapsto \begin{cases} A_{\|m\|_\sigma \|n\|_\sigma} & \text{if } \begin{array}{r} 1 \le \|m\|_\sigma \le \sigma(\mathtt{r}(A)) \\ 1 \le \|n\|_\sigma \le \sigma(\mathtt{c}(A)) \end{array} \\ \mathtt{F} & \text{otherwise} \end{cases}$$

and we translate constructed terms as follows:

$$\|\mathtt{e}(\lambda ij\langle m', n', t\rangle, m, n)\|_\sigma \longmapsto \begin{cases} \|t\|_{\sigma(\|m\|_\sigma/i)(\|n\|_\sigma/j)} & \text{if } \begin{array}{r} 1 \le \|m\|_\sigma \le \|m'\|_\sigma \\ 1 \le \|n\|_\sigma \le \|n'\|_\sigma \end{array} \\ \mathtt{F} & \text{otherwise} \end{cases}$$

Finally, we deal with $\Sigma(T)$ as follows:

$$\begin{aligned}
\|\Sigma(A)\|_\sigma &\longmapsto \mathtt{MOD}_{2,1}(A_{11}, A_{12}, \ldots, A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))}) \\
\|\Sigma(\lambda ij\langle m, n, t\rangle)\|_\sigma &\longmapsto \mathtt{MOD}_{2,1}(\{\|t\|_{\sigma(p/i)(q/j)}\}_{\substack{1\le p\le\|m\|_\sigma \\ 1\le q\le\|n\|_\sigma}})
\end{aligned}$$

This ends the procedure.

**Example 7.2.1** We translate $A + B = B + A$ where $A, B$ are $3 \times 3$ matrices.

$$\begin{aligned}
\|\mathtt{r}(A+B)\|_\sigma &= \|\mathrm{cond}(\mathtt{r}(A) \leq \mathtt{r}(B), \mathtt{r}(B), \mathtt{r}(A))\|_\sigma \\
&= \begin{cases} \|\mathtt{r}(B)\|_\sigma & \text{if } \|\mathtt{r}(A) \leq \mathtt{r}(B)\|_\sigma \\ \|\mathtt{r}(A)\|_\sigma & \text{otherwise} \end{cases} \\
&= \begin{cases} \sigma(\mathtt{r}(B)) & \text{if } \sigma(\mathtt{r}(A)) \leq \sigma(\mathtt{r}(B)) \\ \sigma(\mathtt{r}(A)) & \text{otherwise} \end{cases} \\
&= 3
\end{aligned}$$

Similarly: $\|\mathtt{r}(B+A)\|_\sigma = \|\mathtt{c}(A+B)\|_\sigma = \|\mathtt{c}(B+A)\|_\sigma = 3$. Thus, from $\|A+B = B+A\|_\sigma$ we obtain:

$$\bigwedge_{\substack{1 \leq p \leq 3 \\ 1 \leq q \leq 3}} (\|\mathtt{e}(A+B,i,j)\|_{\sigma(p/i)(q/j)} \equiv \|\mathtt{e}(B+A,i,j)\|_{\sigma(p/i)(q/j)})$$

and from $\|\mathtt{e}(A+B,i,j)\|_{\sigma(p/i)(q/j)}$ we obtain:

$$\begin{aligned}
&\longmapsto \|\mathtt{e}(A,i,j) + \mathtt{e}(B,i,j)\|_{\sigma(p/i)(q/j)} \\
&\longmapsto \mathtt{MOD}_{2,1}(\|\mathtt{e}(A,i,j)\|_{\sigma(p/i)(q/j)}, \|\mathtt{e}(B,i,j)\|_{\sigma(p/i)(q/j)}) \\
&\longmapsto \mathtt{MOD}_{2,1}(A_{\|i\|_{\sigma(p/i)(q/j)}\|j\|_{\sigma(p/i)(q/j)}}, B_{\|i\|_{\sigma(p/i)(q/j)}\|j\|_{\sigma(p/i)(q/j)}}) \\
&\longmapsto \mathtt{MOD}_{2,1}(A_{pq}, B_{pq})
\end{aligned}$$

We do the same for $\mathtt{e}(B+A,i,j)$, and we obtain:

$$\bigwedge_{\substack{1 \leq p \leq 3 \\ 1 \leq q \leq 3}} \mathtt{MOD}_{2,1}(A_{pq}, B_{pq}) \equiv \mathtt{MOD}_{2,1}(B_{pq}, A_{pq})$$

which is what we expected.

### 7.2.3 Correctness of the procedure

Recall that $|\sigma|$, the norm of $\sigma$, is the largest value assignment of $\sigma$. We state the *correctness* of the translating procedure in the form of the following two theorems:

**Theorem 7.2.1** If $S$ is a sequent over $\mathcal{L}_{\mathrm{LA}}$, then, there exists a polynomial $p_S$ and a constant $d_S$ such that for every $\sigma$, $\mathrm{size}(\|S\|_\sigma)$ is bounded by $p_S(|\sigma|)$, and $\mathrm{dp}(\|S\|_\sigma)$ is bounded by $d_S$. Furthermore, if $S$ is a true sequent then, the propositional sequent $\|S\|_\sigma$ is a tautology.

**Theorem 7.2.2** If $S$ is a sequent over $\mathcal{L}_{\mathrm{LA}}$, and $S$ is a theorem of LA, then, there exists a polynomial $q_S$ and a positive integer $d_S$ such that for every $\sigma$, $\|S\|_\sigma$ has a PK[2] derivation $P_{S,\sigma}$ such that size$(P_{S,\sigma})$ is bounded by $q_S(|\sigma|)$ and dp$(P_{S,\sigma})$ is bounded by the constant $d_S$.

*Proof.* (of theorem 7.2.1) Since sequents are made of (finitely many) formulas, it is enough to prove the theorem for formulas: If $\alpha$ is a formula over $\mathcal{L}_{\mathrm{LA}}$, then $\alpha$ is a boolean combination of atomic formulas of the form $m = n, m \leq n, t = u$, and $T = U$. The first two translate into $\mathtt{T}$ or $\mathtt{F}$, for any $\sigma$, so size$(\|m = n\|_\sigma)$ = size$(\|m \leq n\|_\sigma) = 1$, regardless of $\sigma$, and hence they are trivially bounded.

The atomic formula $t = u$ translates into $\|t\|_\sigma \equiv \|u\|_\sigma$, and the atomic formula $T = U$ translates into:

$$\bigwedge_{\substack{1\leq p\leq r\\ 1\leq q\leq c}} (\|\mathtt{e}(T,i,j)\|_{\sigma(p/i)(q/j)} \equiv \|\mathtt{e}(U,i,j)\|_{\sigma(p/i)(q/j)})$$

where $r := \sigma(\mathtt{r}(A)) = \sigma(\mathtt{r}(B))$ and $c := \sigma(\mathtt{c}(A)) = \sigma(\mathtt{c}(B))$. If the sizes are not compatible, then $T = U$ simply translates into $\mathtt{F}$, in which case size$(\|T = U\|_\sigma)$ = size$(\mathtt{F}) = 1$, so it is trivially bounded regardless of $\sigma$.

We need the following two claims to finish the proof:

**Claim 7.2.1** Given any term $m$ of type index, there exists a polynomial $p_m$ such that for any $\sigma$, $\|m\|_\sigma$ is bounded by $p_m(|\sigma|)$.

*Proof.* The proof of this claim is straightforward. □

**Claim 7.2.2** Given any term $t$ of type field, there exists a polynomial $p_t$ such that for any $\sigma$, size$(\|t\|_\sigma)$ is bounded by $p_t(|\sigma|)$.

*Proof.* The proof is by structural induction on $t$.
**Basis Case:** $t$ is of the form $0, 1$ or $a$, for some field variable $a$. Then, $t$ translates to $\mathtt{F}$, $\mathtt{T}$or $a$, respectively, and therefore size$(\|t\|_\sigma) = 1$, regardless of $\sigma$.
**Induction Step:** we consider all the possible ways to form a term of type field; we consider the cases that define terms of type field in the inductive definition of terms and formulas (see section 2.2.1):

case 2. Consider $t + u$. Since $\|t + u\|_\sigma \longmapsto \mathtt{MOD}_{2,1}(\|t\|_\sigma, \|u\|_\sigma)$, we have that:

$$\text{size}(\|t + u\|_\sigma) = \text{size}(\|t\|_\sigma) + \text{size}(\|u\|_\sigma) + 1$$

and by the IH:

$$\leq p_t(|\sigma|) + p_u(|\sigma|) + 1$$

The case $t * u$ is the same as $t + u$, except we have the connective $\wedge$ instead of $\mathtt{MOD}_{2,1}$.

case 3. Consider $-t$ and $t^{-1}$ . Since $\| - t\|_\sigma, \|t^{-1}\|_\sigma \longmapsto \|t\|_\sigma$, it follows that:

$$\text{size}(\| - t\|_\sigma) = \text{size}(\|t^{-1}\|_\sigma) = \text{size}(\|t\|_\sigma)$$

Now we apply the IH to $t$.

case 4. Consider $\Sigma(T)$. The translation depends on whether $T$ is a matrix variable, or a constructed term:

$$\|\Sigma(A)\|_\sigma \longmapsto \mathtt{MOD}_{2,1}(A_{11}, A_{12}, \ldots, A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))})$$
$$\|\Sigma(\lambda ij\langle m, n, t\rangle)\|_\sigma \longmapsto \mathtt{MOD}_{2,1}(\{\|t\|_{\sigma(p/i)(q/j)}\}_{\substack{1\leq p\leq \|m\|_\sigma \\ 1\leq q\leq \|n\|_\sigma}})$$

In the first case:

$$\begin{aligned}\text{size}(\|\Sigma(A)\|_\sigma) &= \text{size}(\mathtt{MOD}_{2,1}(A_{11}, A_{12}, \ldots, A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))})) \\ &= 1 + \sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A)) \leq 1 + |\sigma|^2\end{aligned}$$

In the second case:

$$\begin{aligned}\text{size}(\|\Sigma(\lambda ij\langle m, n, t\rangle)\|_\sigma) &= \text{size}(\mathtt{MOD}_{2,1}(\{\|t\|_{\sigma(p/i)(q/j)}\}_{\substack{1\leq p\leq \|m\|_\sigma \\ 1\leq q\leq \|n\|_\sigma}})) \\ &= 1 + \sum_{\substack{1\leq p\leq \|m\|_\sigma \\ 1\leq q\leq \|n\|_\sigma}} \text{size}(\|t\|_{\sigma(p/i)(q/j)})\end{aligned}$$

But by the IH $\text{size}(\|t\|_{\sigma(p/i)(q/j)})$ is bounded by $p_t(|\sigma(p/i)(q/j)|)$, and by claim 7.2.1: $1 \leq p \leq \|m\|_\sigma \leq p_m(|\sigma|)$ and $1 \leq q \leq \|n\|_\sigma \leq p_n(|\sigma|)$. Thus:

$$p_t(|\sigma(p/i)(q/j)|) \leq p_t(|\sigma| + p_m(|\sigma|) + p_n(|\sigma|)) \leq p'_t(|\sigma|)$$

and we are done.

case 5. Consider $\mathtt{e}(T, m, n)$. If $T$ is just a matrix variable, then:

$$\text{size}(\|\mathtt{e}(A, m, n)\|_\sigma) = \text{size}(A_{\|m\|_\sigma \|n\|_\sigma}) = 1$$

If $T$ is a constructed matrix, then:

$$\text{size}(\|\mathtt{e}(\lambda ij\langle m', n', t\rangle, m, n)\|_\sigma) = \text{size}(\|t\|_{\sigma(\|m\|_\sigma/i)(\|n\|_\sigma/j)}) \tag{7.1}$$

if $1 \leq \|m\|_\sigma \leq \|m'\|_\sigma$ and $1 \leq \|n\|_\sigma \leq \|n'\|_\sigma$, and otherwise:

$$\text{size}(\|\mathtt{e}(\lambda ij\langle m', n', t\rangle, m, n)\|_\sigma) = \text{size}(\mathtt{F}) = 1$$

in which case we are done.

So suppose that equation (7.1) is the case. Then, by the IH, we know that $\text{size}(\|t\|_{\sigma(\|m\|_\sigma/i)(\|n\|_\sigma/j)})$ is bounded by $p_t(|\sigma(\|m\|_\sigma/i)(\|n\|_\sigma/j)|)$, and by claim 7.2.1, $\|m\|_\sigma \leq p_m(|\sigma|)$ and $\|n\|_\sigma \leq p_n(|\sigma|)$, so we are done.

case 9. Consider $\text{cond}(\alpha, t, u)$. Since:

$$\|\text{cond}(\alpha, t, u)\|_\sigma \longmapsto \begin{cases} \|t\|_\sigma & \text{if } \|\alpha\|_\sigma \\ \|u\|_\sigma & \text{otherwise} \end{cases}$$

it follows that:

$$\begin{aligned} \text{size}(\|\text{cond}(\alpha, t, u)\|_\sigma) &\leq \max\{\text{size}(\|t\|_\sigma), \text{size}(\|u\|_\sigma)\} \\ &\leq \text{size}(\|t\|_\sigma) + \text{size}(\|u\|_\sigma) \end{aligned}$$

and the claim now follows by the IH.

This finishes the proof of claim 7.2.2, and of the first part of the theorem. □

We still have to prove that if $\mathcal{S}_{\mathbb{Z}_2} \vDash \alpha$, then, for any $\sigma$, the propositional formula $\|\alpha\|_\sigma$ is a tautology.

Suppose that $\mathcal{S}_{\mathbb{Z}_2} \vDash \alpha$. Let $\sigma$ be a fixed partial assignment to the free index variables in $\alpha$, and to all the terms of the form $\mathtt{r}(A)$ and $\mathtt{c}(A)$ for all matrix variables $A$ in $\alpha$. We define $\tau_\sigma$ to be an object assignment where $\tau_\sigma(i) = \sigma(i)$ if $i$ is a free index variable in $\alpha$, and $\tau_\sigma(\mathtt{r}(A)) = \sigma(\mathtt{r}(A))$ and $\tau_\sigma(\mathtt{c}(A)) = \sigma(\mathtt{c}(A))$ if $A$ is a matrix variable in $\alpha$; $\tau_\sigma$ *extends* $\sigma$. Since $\mathcal{S}_{\mathbb{Z}_2} \vDash \alpha$, it follows that $\mathcal{S}_{\mathbb{Z}_2} \vDash \alpha[\tau_\sigma]$, for all object assignment $\tau_\sigma$.

There is a natural correspondence between object assignments $\tau_\sigma$ to the field and matrix variables in $\alpha$, and truth value assignments $\tau'_\sigma$ to the propositional variables in

$\|\alpha\|_\sigma$, namely:

$$\tau_\sigma(a) = 1 \in \mathbb{Z}_2 \iff \tau'_\sigma(a) = \mathtt{T}$$

$$\tau_\sigma(A) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \vdots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \in M_{m\times n}(\mathbb{Z}_2) \iff \tau'_\sigma(A_{ij}) = a_{ij} \in \{\mathtt{T}, \mathtt{F}\}$$

for all $1 \leq i \leq m$ and $1 \leq j \leq n$. Now, the rest of the proof will follow from the next claim:

**Claim 7.2.3** Suppose that $\mathcal{S}_{\mathbb{Z}_2} \vDash \alpha$. Let $\sigma$ be a fixed partial object assignment to the free index variables in $\alpha$, and to the index terms of the form $\mathtt{r}(A)$ and $\mathtt{c}(A)$, for all matrix variables $A$ in $\alpha$. If $\tau_\sigma$ and $\tau'_\sigma$ are any object assignments, defined as in the paragraphs above, then:

$$\mathcal{S}_{\mathbb{Z}_2} \vDash (m = n)[\tau_\sigma] \quad \text{iff} \quad \|m = n\|_\sigma \longmapsto \mathtt{T}$$
$$\mathcal{S}_{\mathbb{Z}_2} \vDash (m \leq n)[\tau_\sigma] \quad \text{iff} \quad \|m \leq n\|_\sigma \longmapsto \mathtt{T}$$
$$\mathcal{S}_{\mathbb{Z}_2} \vDash (t = u)[\tau_\sigma] \quad \text{iff} \quad \tau'_\sigma(\|t = u\|_\sigma) = \mathtt{T}$$
$$\mathcal{S}_{\mathbb{Z}_2} \vDash (T = U)[\tau_\sigma] \quad \text{iff} \quad \tau'_{\sigma(p/i)(q/j)}(\|\mathtt{e}(T,i,j) = \mathtt{e}(U,i,j)\|_{\sigma(p/i)(q/j)}) = \mathtt{T}$$

for all $1 \leq p \leq r$ and $1 \leq q \leq c$, where $r := \tau_\sigma(\mathtt{r}(T)) = \sigma(\mathtt{r}(T)) = \sigma(\mathtt{r}(U))$ and $c := \tau_\sigma(\mathtt{c}(T)) = \sigma(\mathtt{c}(T)) = \sigma(\mathtt{c}(U))$.

*Proof.* Let $\mathcal{S} = \mathcal{S}_{\mathbb{Z}_2}$, and consider the first statement: $\mathcal{S} \vDash (m = n)[\tau_\sigma]$ iff $\|m = n\|_\sigma \longmapsto \mathtt{T}$. $\mathcal{S} \vDash (m = n)[\tau_\sigma]$ iff $m^{\mathcal{S}}[\tau_\sigma] = n^{\mathcal{S}}[\tau_\sigma]$. It is easy to show, by structural induction on terms $m$ of type index, that $m^{\mathcal{S}}[\tau_\sigma] = \|m\|_\sigma$. Therefore, $m^{\mathcal{S}}[\tau_\sigma] = n^{\mathcal{S}}[\tau_\sigma]$ iff $\|m\|_\sigma = \|n\|_\sigma$ iff $\|m = n\|_\sigma \longmapsto \mathtt{T}$.

The second statement, $\mathcal{S} \vDash (m \leq n)[\tau_\sigma]$ iff $\|m \leq n\|_\sigma \longmapsto \mathtt{T}$, can be proven similarly.

Consider $\mathcal{S} \vDash (t = u)[\tau_\sigma]$ iff $\tau'_\sigma(\|t = u\|_\sigma) = \mathtt{T}$. $\mathcal{S} \vDash (t = u)[\tau_\sigma]$ iff $t^{\mathcal{S}}[\tau_\sigma] = u^{\mathcal{S}}[\tau_\sigma]$. Again, it is easy to show, by structural induction on terms $t$ of type field, that $t^{\mathcal{S}}[\tau_\sigma] = 1 \in \mathbb{Z}_2$ iff $\tau'_\sigma(\|t = u\|_\sigma) = \mathtt{T}$. Therefore, $t^{\mathcal{S}}[\tau_\sigma] = u^{\mathcal{S}}[\tau_\sigma]$ iff $\tau'_\sigma(\|t\|_\sigma \equiv \|u\|_\sigma) = \mathtt{T}$ iff $\tau'_\sigma(\|t = u\|_\sigma) = \mathtt{T}$.

The last statement follows from the definition of $\|T = U\|_\sigma$, and from the previous statement. This ends the proof of claim 7.2.3. □

This ends the proof of theorem 7.2.1. □

We need the following four results for the proof of theorem 7.2.2:

**Lemma 7.2.1** PK[$a$] derivations have the substitution property; that is, if $P$ is a PK[$a$] derivation of $\alpha(x)$, where $x$ is an atom, then $P[\beta/x]$ is a proof of $\alpha(\beta/x)$. (Here $\alpha$ and $\beta$ are propositional formulas, and $x$ is a propositional variable. Since LA has the (derived) substitution rule, we need an analogous thing for PK[$a$] in order to carry out the translations.)

*Proof.* This is a straightforward proof by induction on the length of the derivation. The **Basis Case** is when $\alpha(x)$ is an axiom, in which case $\alpha(\beta/x)$ is also an axiom (note that $\beta$ does *not* have modular gates, as PK[$a$] is a proof system for the classical tautologies). In the **Induction Step** we consider all the rules (and the modular axioms), and we show that if the property holds for the top sequent(s) it holds for the bottom. □

**Lemma 7.2.2** If $\alpha(a)$ is a formula over $\mathcal{L}_{\mathrm{LA}}$ (and $a$ is a field variable), then:

$$\|\alpha(t/a)\|_\sigma =_{\mathrm{synt}} \|\alpha(a)\|_\sigma(\|t\|_\sigma/a) \tag{7.2}$$

for any term $t$ of type field.

*Proof.* This is a straightforward proof by structural induction on the formula $\alpha$. The **Basis Case** is when $\alpha(a)$ is an atomic formula: $t = u, t \leq u$. We now prove this by structural induction on $t$ and $u$; in the inductive step of this second argument, we consider all the steps in the procedure for translating terms of type field. The main **Induction Step** is proving the result for boolean combinations of atomic formulas. □

**Lemma 7.2.3** If $\alpha(i)$ is a formula over $\mathcal{L}_{\mathrm{LA}}$ (and $i$ is an index variable), then:

$$\|\alpha(m/i)\|_\sigma =_{\mathrm{synt}} \|\alpha(i)\|_{\sigma(\|m\|_\sigma/i)} \tag{7.3}$$

for any term $m$ of type index.

*Proof.* Same straightforward idea as in the proof of Lemma 7.2.2, but since in the translation we actually evaluate terms of type index, in the right hand side of (7.3) we change (accordingly to the left hand side) the value of $\sigma$, rather than the boolean variables of the formula as in (7.2). □

**Corollary 7.2.1** If $S(\mathbf{a},\mathbf{i})$ is a sequent over $\mathcal{L}_{\mathrm{LA}}$, and $\mathbf{a} = a_1,\ldots,a_k$ and $\mathbf{i} = i_1,\ldots,i_l$ are field and index variables (respectively), then:

$$\|S(\mathbf{t}/\mathbf{a},\mathbf{m}/\mathbf{i})\|_\sigma =_{\mathrm{synt}} \|S(\mathbf{a},\mathbf{i})\|_{\sigma(\|\mathbf{m}\|_\sigma/\mathbf{i})}(\|\mathbf{t}\|_\sigma/\mathbf{a})$$

where $\mathbf{t} = t_1,\ldots,t_k$ and $\mathbf{m} = m_1,\ldots,m_l$ are terms of type field and index (respectively). Note that $\mathbf{t}/\mathbf{a}$ denotes $t_1/a_1,\ldots,t_k/a_k$.

*Proof.* Since sequents are cedents of formulas, this Corollary follows directly from (7.2) and (7.3). □

*Proof.* **(of Theorem 7.2.2)** Suppose that $S$ is a theorem of LA. Then, by Definition 2.4.6, $S$ has a PK-LA derivation $\pi = \{S_1, S_2, \ldots, S_n\}$, where $S =_{\mathrm{synt}} S_n$. We are going to prove the Theorem by induction on $n$. There are two cases: the axiom case, where $S_i$ is an axiom, and the rule case, where $S_i$ follows by one of the rules from previous sequents in the derivation $\pi$. Note that in the Basis Case of the induction, $S_1$ must be an axiom. In the Induction Step, $S_i$ either follows from previous sequents in $\pi$, or it is also an axiom.

The goal is to produce a PK[2] derivation $P$ of $\|S\|_\sigma$, that satisfies the conditions of the Theorem (it is poly-bounded in $|\sigma|$ and all formulas have bounded depth). The fact that these conditions hold is obvious from the proof, so they will not be stated explicitly.

**Axiom Case**

We will go through the list of the axiom schemas, showing that each axiom translates into families of propositional tautologies with short PK[2] proofs of bounded depth. Note that by convention (2.14) — the convention that we made in section 2.3 — all substitution instances of axioms are also axioms. Corollary 7.2.1 will take care of the case where we replace variables of type index or field by appropriate terms. We still need to consider the cases where we replace variables of type matrix by constructed matrices.

A1: Since this is the first axiom that we deal with, we will do it in a little bit more detail than necessary. Recall that A1 is $x = x$. If we replace $x$ by a variable $i$ of type index, then we get $i = i$. Then, $\|i = i\|_\sigma \longmapsto \mathtt{T}$ because $\|i\|_\sigma = \|i\|_\sigma$ for any index variable $i$, and $\mathtt{T}$ has a trivial PK[2] proof.

If $x$ is the field variable $a$, then $\|a = a\|_\sigma \longmapsto a \equiv a$, which has a short tree-like PK[2] proof regardless of $\sigma$.

If $x$ is the matrix variable $A$, then:

$$\|A = A\|_\sigma \longmapsto \bigwedge_{\substack{1\le p\le \sigma(\mathtt{r}(A))\\ 1\le q\le \sigma(\mathtt{c}(A))}} A_{pq} \equiv A_{pq}$$

and since for each pair $p, q$, $A_{pq} \equiv A_{pq}$ has a short tree-like PK[2] proof of constant size, the conjunction $\bigwedge_{\substack{1\le p\le \sigma(\mathtt{r}(A))\\ 1\le q\le \sigma(\mathtt{c}(A))}} A_{pq} \equiv A_{pq}$ has a tree-like PK[2] proof of size polynomial in $|\sigma|$.

If $x$ is replaced by a constructed term $T =_{\text{synt}} \lambda ij\langle m, n, t\rangle$, then the expression $\|T = T\|_\sigma$ becomes:

$$\bigwedge_{\substack{1\le p\le \|m\|_\sigma\\ 1\le q\le \|n\|_\sigma}} (\|t\|_{\sigma(p/i)(q/j)} \equiv \|t\|_{\sigma(p/i)(q/j)}) \tag{7.4}$$

and since each $\|t\|_{\sigma(p/i)(q/j)} \equiv \|t\|_{\sigma(p/i)(q/j)}$ has a short PK[2] proof, so does the conjunction.

A2: Easy if $x, y$ are index or field variables. Consider $A = B \to B = A$. If $\sigma(\mathtt{r}(A)) = \sigma(\mathtt{r}(B))$ and $\sigma(\mathtt{c}(A)) = \sigma(\mathtt{c}(B))$, then $\|A = B \to B = A\|_\sigma$ becomes:

$$\bigwedge_{\substack{1\le p\le \sigma(\mathtt{r}(A))\\ 1\le q\le \sigma(\mathtt{c}(A))}} (A_{pq} \equiv B_{pq}) \to \bigwedge_{\substack{1\le p\le \sigma(\mathtt{r}(A))\\ 1\le q\le \sigma(\mathtt{c}(A))}} (B_{pq} \equiv A_{pq})$$

which clearly has a short PK[2] proof. Otherwise, if the sizes are not compatible, then $\|A = B \to B = A\|_\sigma \longmapsto \mathtt{F} \to \mathtt{F}$ which has a trivial PK[2] proof.

Suppose that $x$ and $y$ are replaced by the constructed terms $\lambda ij\langle m, n, t\rangle$ and $\lambda ij\langle m', n', t'\rangle$. Then we have:

$$\begin{aligned}\bigwedge_{\substack{1\le p\le \|m\|_\sigma\\ 1\le q\le \|n\|_\sigma}} &(\|t\|_{\sigma(p/i)(q/j)} \equiv \|t'\|_{\sigma(p/i)(q/j)})\\ &\to \bigwedge_{\substack{1\le p\le \|m\|_\sigma\\ 1\le q\le \|n\|_\sigma}} (\|t'\|_{\sigma(p/i)(q/j)} \equiv \|t\|_{\sigma(p/i)(q/j)})\end{aligned}$$

if the sizes are compatible, that is, if $\|m\|_\sigma = \|m'\|_\sigma$ and $\|n\|_\sigma = \|n'\|_\sigma$. If the sizes are not compatible we get $\mathtt{F} \to \mathtt{F}$.

A3: Same idea as A1 and A2.

A4: It is enough to consider $f \in \{\mathtt{r}, \mathtt{c}, \mathtt{e}, \Sigma\}$ as it is easy for the other function symbols.

Suppose $f = \mathtt{r}$ and consider $A = B \to \mathtt{r}(A) = \mathtt{r}(B)$. Just note that if $\|\mathtt{r}(A)\|_\sigma \neq \|\mathtt{r}(B)\|_\sigma$, then $\|A = B\|_\sigma \longmapsto \mathtt{F}$.

Suppose $f = \mathtt{e}$. Consider:

$$i_1 = i_2, j_1 = j_2, A = B \to \mathtt{e}(A, i_1, j_1) = \mathtt{e}(B, i_2, j_2)$$

Just note that if $\|i_1 = i_2\|_\sigma \longmapsto \mathtt{T}$ and $\|j_1 = j_2\|_\sigma \longmapsto \mathtt{T}$, then we have $\sigma(i_1) = \sigma(i_2)$ and $\sigma(j_1) = \sigma(j_2)$.

Finally consider $f = \Sigma$. In this case the axiom becomes:

$$A = B \to \Sigma(A) = \Sigma(B)$$

If the sizes are compatible, from $\|A = B \to \Sigma(A) = \Sigma(B)\|_\sigma$ we obtain:

$$\begin{aligned}
&\longmapsto \|A = B\|_\sigma \to \|\Sigma(A) = \Sigma(B)\|_\sigma \\
&\longmapsto \bigwedge_{\substack{1 \le p \le \sigma(\mathtt{r}(A)) \\ 1 \le q \le \sigma(\mathtt{c}(A))}} A_{pq} \equiv B_{pq} \\
&\qquad \to \mathtt{MOD}_{2,1}(\{A_{pq}\}_{\substack{1 \le p \le \sigma(\mathtt{r}(A)) \\ 1 \le q \le \sigma(\mathtt{c}(A))}}) \equiv \mathtt{MOD}_{2,1}(\{B_{pq}\}_{\substack{1 \le p \le \sigma(\mathtt{r}(A)) \\ 1 \le q \le \sigma(\mathtt{c}(A))}})
\end{aligned}$$

A5–A17: It is easy to check that each of these axioms is mapped to $\mathtt{T}$ for any $\sigma$.

A18: $\| \to 0 + a = a\|_\sigma$

$$\begin{aligned}
&\longmapsto \to \|0 + a\|_\sigma \equiv \|a\|_\sigma \\
&\longmapsto \to \mathtt{MOD}_{2,1}(\mathtt{F}, a) \equiv a
\end{aligned}$$

A19: $\| \to a + (-a) = 0\|_\sigma$

$$\begin{aligned}
&\longmapsto \to \|a + (-a) = 0\|_\sigma \\
&\longmapsto \to \|a + (-a)\|_\sigma \equiv \|0\|_\sigma \\
&\longmapsto \to \mathtt{MOD}_{2,1}(a, \| - a\|_\sigma) \equiv \mathtt{F} \\
&\longmapsto \to \mathtt{MOD}_{2,1}(a, a) \equiv \mathtt{F}
\end{aligned}$$

A20: $\| \to 1 * a = a \|_\sigma$

$$\begin{aligned}
&\longmapsto \to \|1 * a\|_\sigma \equiv \|a\|_\sigma \\
&\longmapsto \to (\|1\|_\sigma \wedge \|a\|_\sigma) \equiv \|a\|_\sigma \\
&\longmapsto \to (\mathtt{T} \wedge a) \equiv a
\end{aligned}$$

A21: $\|a \neq 0 \to a * (a^{-1}) = 1\|_\sigma$

$$\begin{aligned}
&\longmapsto \|a \neq 0\|_\sigma \to \|a * (a^{-1}) = 1\|_\sigma \\
&\longmapsto \neg(a \equiv \mathtt{F}) \to \|a * (a^{-1})\|_\sigma \equiv \mathtt{T} \\
&\longmapsto \neg(a \equiv \mathtt{F}) \to (a \wedge \|a^{-1}\|_\sigma) \equiv \mathtt{T} \\
&\longmapsto \neg(a \equiv \mathtt{F}) \to (a \wedge a) \equiv \mathtt{T}
\end{aligned}$$

A22: $\| \to a + b = b + a \|_\sigma$

$$\begin{aligned}
&\longmapsto \to \|a + b\|_\sigma \equiv \|b + a\|_\sigma \\
&\longmapsto \to \mathtt{MOD}_{2,1}(a, b) \equiv \mathtt{MOD}_{2,1}(b, a)
\end{aligned}$$

A23: $\| \to a * b = b * a \|_\sigma$

$$\begin{aligned}
&\longmapsto \to \|a * b\|_\sigma \equiv \|b * a\|_\sigma \\
&\longmapsto \to a \wedge b \equiv b \wedge a
\end{aligned}$$

A24: $\| \to a + (b + c) = (a + b) + c \|_\sigma$

$$\begin{aligned}
&\longmapsto \to \|a + (b + c)\|_\sigma \equiv \|(a + b) + c\|_\sigma \\
&\longmapsto \to \mathtt{MOD}_{2,1}(a, \|b + c\|_\sigma) \equiv \mathtt{MOD}_{2,1}(\|a + b\|_\sigma, c) \\
&\longmapsto \to \mathtt{MOD}_{2,1}(a, \mathtt{MOD}_{2,1}(b, c)) \equiv \mathtt{MOD}_{2,1}(\mathtt{MOD}_{2,1}(a, b), c)
\end{aligned}$$

A25: $\| \to a * (b * c) = (a * b) * c\|_\sigma$

$$\longmapsto \to a \wedge (b \wedge c) \equiv (a \wedge b) \wedge c$$

A26: $\| \to a * (b + c) = (a * b) + (a * c)\|_\sigma$

$$\longmapsto \to a \wedge \mathtt{MOD}_{2,1}(b, c) \equiv \mathtt{MOD}_{2,1}(a \wedge b, a \wedge c)$$

A27: $\|\alpha \to \mathrm{cond}(\alpha, a, b) = a\|_\sigma$

$$\longmapsto \|\alpha\|_\sigma \to \|\mathrm{cond}(\alpha, a, b) = a\|_\sigma$$
$$\longmapsto \|\alpha\|_\sigma \to \|\mathrm{cond}(\alpha, a, b)\|_\sigma \equiv a$$

Suppose $\|\alpha\|_\sigma \longmapsto \mathtt{T}$. Then $\|\mathrm{cond}(\alpha, a, b)\|_\sigma \longmapsto a$, so in this case:

$$\|\alpha \to \mathrm{cond}(\alpha, a, b) = a\|_\sigma \longmapsto \mathtt{T} \to a \equiv a$$

A28: $\|i = 0 \vee \mathtt{r}(A) < i \vee j = 0 \vee \mathtt{c}(A) < j \to \mathtt{e}(A, i, j) = 0\|_\sigma$ maps to

$$\|i = 0\|_\sigma \vee \|\mathtt{r}(A) < i\|_\sigma \vee \|j = 0\|_\sigma \vee \|\mathtt{c}(A) < j\|_\sigma \to \|\mathtt{e}(A, i, j)\|_\sigma \equiv \mathtt{F}$$

Suppose that the antecedent is true. Then, one of the formulas in the disjunction has to be true, and so $\|\mathtt{e}(A, i, j)\|_\sigma \longmapsto \mathtt{F}$, so the succedent is $\mathtt{F} \equiv \mathtt{F}$, and hence it is valid.

A29:

$$\| \to \mathtt{r}(\lambda ij\langle m, n, t\rangle) = m\|_\sigma \longmapsto\to \mathtt{T}$$
$$\| \to \mathtt{c}(\lambda ij\langle m, n, t\rangle) = m\|_\sigma \longmapsto\to \mathtt{T}$$

Now consider $\|1 \leq i, i \leq m, 1 \leq j, j \leq n \to \mathtt{e}(\lambda ij\langle m, n, t\rangle, i, j) = t\|_\sigma$ which maps to:

$$\|1 \leq i\|_\sigma, \|i \leq m\|_\sigma, \|1 \leq j\|_\sigma, \|j \leq n\|_\sigma \to \|\mathtt{e}(\lambda ij\langle m, n, t\rangle, i, j)\|_\sigma \equiv \|t\|_\sigma$$

If all the formulas in the antecedent are true, then:

$$\|\mathtt{e}(\lambda ij\langle m, n, t\rangle, i, j)\|_\sigma \longmapsto \|t\|_\sigma$$

so the succedent becomes $\|t\|_\sigma \equiv \|t\|_\sigma$, and the entire sequent becomes:

$$\mathtt{T}, \mathtt{T}, \mathtt{T}, \mathtt{T} \to \|t\|_\sigma \equiv \|t\|_\sigma$$

A30: $\|\mathtt{r}(A) = 1, \mathtt{c}(A) = 1 \rightarrow \Sigma(A) = \mathtt{e}(A,1,1)\|_\sigma$ becomes:

$$\|\mathtt{r}(A) = 1\|_\sigma, \|\mathtt{c}(A) = 1\|_\sigma \rightarrow \|\Sigma(A)\|_\sigma \equiv \|\mathtt{e}(A,1,1)\|_\sigma$$

and if both formulas in the antecedent are true, then:

$$\|\Sigma(A)\|_\sigma \longmapsto \mathtt{MOD}_{2,1}(A_{11})$$
$$\|\mathtt{e}(A,1,1)\|_\sigma \longmapsto A_{11}$$

so in this case the sequent becomes:

$$\mathtt{T}, \mathtt{T} \rightarrow \mathtt{MOD}_{2,1}(A_{11}) \equiv A_{11}$$

A31: $\|\mathtt{r}(A) = 1, 1 < \mathtt{c}(A) \rightarrow \Sigma(A) = \Sigma(\lambda ij\langle 1, \mathtt{c}(A) - 1, A_{ij}\rangle) + A_{1\mathtt{c}(A)}\|_\sigma$ becomes:

$$\|\mathtt{r}(A) = 1\|_\sigma, \|1 < \mathtt{c}(A)\|_\sigma$$
$$\rightarrow \|\Sigma(A)\|_\sigma \equiv \mathtt{MOD}_{2,1}(\|\Sigma(\lambda ij\langle 1, \mathtt{c}(A) - 1, A_{ij}\rangle)\|_\sigma, \|A_{1\mathtt{c}(A)}\|_\sigma)$$

Suppose that both formulas in the antecedent are true. Then we have:

$$\mathtt{T}, \mathtt{T} \rightarrow \mathtt{MOD}_{2,1}(A_{11}, \ldots, A_{1\sigma(\mathtt{c}(A))}) \equiv$$
$$\mathtt{MOD}_{2,1}(\mathtt{MOD}_{2,1}(A_{11}, \ldots, A_{1\sigma(\mathtt{c}(A))-1}), A_{1\sigma(\mathtt{c}(A))})$$

A32: $\|\mathtt{c}(A) = 1 \rightarrow \Sigma(A) = \Sigma(A^t)\|_\sigma$ becomes:

$$\|\mathtt{c}(A) = 1\|_\sigma \rightarrow \|\Sigma(A)\|_\sigma \equiv \|\Sigma(A^t)\|_\sigma$$

Suppose that the formula in the antecedent is true, and recall that $A^t$ is defined by $\lambda ij\langle \mathtt{c}(A), \mathtt{r}(A), A_{ji}\rangle$. We obtain:

$$\mathtt{T} \rightarrow \mathtt{MOD}_{2,1}(A_{11}, \ldots, A_{\sigma(\mathtt{r}(A))1}) \equiv \mathtt{MOD}_{2,1}(\{\|A_{ji}\|_{\sigma(p/i)(q/j)}\}_{\substack{1 \leq p \leq \sigma(\mathtt{c}(A)) = 1 \\ 1 \leq q \leq \sigma(\mathtt{r}(A))}})$$

which is simply:

$$\mathtt{T} \rightarrow \mathtt{MOD}_{2,1}(A_{11}, \ldots, A_{\sigma(\mathtt{r}(A))1}) \equiv \mathtt{MOD}_{2,1}(A_{11}, \ldots, A_{\sigma(\mathtt{r}(A))1})$$

A33: Suppose that $\|1 < \mathtt{r}(A)\|_\sigma, \|1 < \mathtt{c}(A)\|_\sigma \longmapsto \mathtt{T}$. Then, the RHS of the sequent maps to a propositional formula of the form $\phi_1 \equiv \phi_2$, where $\phi_1$ is $\|\Sigma(A)\|_\sigma$ and where

$\phi_2$ is a formula whose outermost gate is $\mathtt{MOD}_{2,1}$ with the following four arguments:

$$\begin{aligned}
\|\mathtt{e}(A,1,1)\|_\sigma &\longmapsto A_{11}\\
\|\Sigma\lambda ij\langle 1, \mathtt{c}(A)-1, \mathtt{e}(A,1,i+1)\rangle\|_\sigma &\longmapsto \mathtt{MOD}_{2,1}(A_{12},\ldots,A_{1\sigma(\mathtt{c}(A))})\\
\|\Sigma\lambda ij\langle \mathtt{r}(A)-1, 1, \mathtt{e}(A,i+1,1)\rangle\|_\sigma &\longmapsto \mathtt{MOD}_{2,1}(A_{21},\ldots,A_{\sigma(\mathtt{r}(A))1})\\
\|\Sigma\lambda ij\langle \mathtt{r}(A)-1, \mathtt{c}(A)-1, \mathtt{e}(A,i+1,j+1)\rangle\|_\sigma &\longmapsto \mathtt{MOD}_{2,1}(A_{22},\ldots,A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))})
\end{aligned}$$

Thus, it comes down to proving the following assertion:

$$\mathtt{MOD}_{2,1}(A) \equiv \mathtt{MOD}_{2,1}(A_{11}, \mathtt{MOD}_{2,1}(R), \mathtt{MOD}_{2,1}(S), \mathtt{MOD}_{2,1}(M))$$

where $\mathtt{MOD}_{2,1}(A)$ means the obvious: $\mathtt{MOD}_{2,1}(A_{11},\ldots,A_{\sigma(\mathtt{r}(A))\sigma(\mathtt{c}(A))})$. Similarly, $R, S$ and $M$ on the RHS abbreviate the corresponding sets of propositional variables.

**Rule Case**

We examine the three rules, and show that if the translations of the premises have feasible polysize PK[2] derivations of bounded depth, so does the translation of the conclusion.

Ind: Since by IH, $\|\alpha(i) \to \alpha(i+1/i)\|_\sigma$ has short PK[2] derivations for all $\sigma$, it follows that for any particular fixed $\sigma_0$, the propositional formulas:

$$\|\alpha(i)\|_{\sigma_0(p/i)} \to \|\alpha(i+1/i)\|_{\sigma_0(p/i)} \quad \text{for } 0 \le p < \sigma_0(n) \tag{7.5}$$

have short PK[2] derivations. From $\|\alpha(0/i)\|_{\sigma_0}$ and the formulas given by (7.5) we can conclude, with a sequence of modus ponens, $\|\alpha(n/i)\|_{\sigma_0}$.

Eq: Suppose that $\|\mathtt{r}(T) = \mathtt{r}(U)\|_\sigma \longmapsto \mathtt{T}$ and $\|\mathtt{c}(T) = \mathtt{c}(U)\|_\sigma \longmapsto \mathtt{T}$, and that $\|\mathtt{e}(T,i,j) = \mathtt{e}(U,i,j)\|_\sigma$ has a short PK[2] derivation for all $\sigma$, and in particular for all $\sigma(i) \in \{1,\ldots,r\}$ and $\sigma(j) \in \{1,\ldots,c\}$, where $r = \|\mathtt{r}(T)\|_\sigma = \|\mathtt{r}(U)\|_\sigma$ and $c = \|\mathtt{c}(T)\|_\sigma = \|\mathtt{c}(U)\|_\sigma$. Therefore, $\|T = U\|_\sigma$, which is given by:

$$\bigwedge_{\substack{1\le p\le r\\ 1\le q\le c}} \|\mathtt{e}(T,i,j)\|_{\sigma(p/i)(q/j)} \equiv \|\mathtt{e}(U,i,j)\|_{\sigma(p/i)(q/j)}$$

also has short PK[2] derivations.

PK rules: This case is easy.

This finishes the proof of theorem 7.2.2. □

## 7.3 Translating theorems of LA over $\mathbb{Z}_p$ and $\mathbb{Q}$

In this section we discuss briefly the translations of the theorems of LA over finite fields of the form $\mathbb{Z}_p$ ($p$ prime), and over the field of the rationals $\mathbb{Q}$. The finite field $\mathbb{Z}_p$, for $p$ a prime, is the set $\{0, 1, \ldots, p-1\}$ with mod $p$ addition and multiplication.

We will concentrate on the encoding of the field elements. We will not restate the theorems and proofs for bigger fields since they are analogous to the $\mathbb{Z}_2$ case which has been done in great detail in the previous section.

So suppose that $\mathbb{F} = \mathbb{Z}_p$, for $p$ a prime. In that case we cannot translate field elements directly; we need several propositional variables to represent a single field variable.

Since $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$ has finitely many elements, we can use the unary notation to represent (feasibly) each field element In the translations, we are going to associate with each field variable $a$ the following boolean variables: $a^1, a^2, \ldots, a^{p-1}$, which are going to encode (in unary) the value of $a$. For example, if $p = 5$ and $a = 3$, then $a^5 a^4 a^3 a^2 a^1$ is `F F T T T`.

If $t$ is a term of type field, then $\|t\|^j_\sigma$ denotes the value of the $j$ position in the unary representation of $t$. In particular, $\|a\|^j_\sigma$ is $a^j$.

We allow the connectives $\mathtt{MOD}_{p,i}$, for $0 \leq i \leq p-1$, and we include its defining axioms (see section 7.1). Also, for each sequence of variables $a^1, a^2, \ldots, a^p$ encoding the field variable $a$, we add the following set of axioms:

$$a_{i+1} \supset a_i \qquad \text{for } 1 \leq i \leq p-1$$

This ensures that field elements are properly encoded in unary.

We need to make some changes in the procedure for the translation. First:

$$\|t = u\|_\sigma \longmapsto \bigwedge_{1 \leq j \leq p} \|t\|^j_\sigma \equiv \|u\|^j_\sigma$$

Now we make the following changes in the (sub)procedure that translates terms of type index and field (we only need to modify the cases that deal with terms of type field). In the Base Case:

$$\begin{aligned}
\|0_{\text{field}}\|^j_\sigma &\longmapsto \mathtt{F} \qquad \text{for } 1 \leq j \leq p \\
\|1_{\text{field}}\|^1_\sigma &\longmapsto \mathtt{T} \\
\|1_{\text{field}}\|^j_\sigma &\longmapsto \mathtt{F} \qquad \text{for } 2 \leq j \leq p \\
\|a\|^j_\sigma &\longmapsto a^j
\end{aligned}$$

for $1 \leq j \leq p$. Now consider the Recursive Step. Addition is given as follows:

$$\|t+u\|_\sigma^j \longmapsto \bigvee_{j \leq i \leq p-1} \mathtt{MOD}_{p,i}(\{\|t\|_\sigma^k\}_{1 \leq k \leq p}, \{\|u\|_\sigma^k\}_{1 \leq k \leq p})$$

We now show how to translate products, and additive and multiplicative inverses:

$$\|t * u\|_\sigma^j \longmapsto \bigvee_{\substack{1 \leq i,k \leq p-1 \\ j \leq (ik \bmod p)}} (\|t\|_\sigma^i \wedge \neg\|t\|_\sigma^{i+1}) \wedge (\|u\|_\sigma^k \wedge \neg\|u\|_\sigma^{k+1})$$

$$\|-t\|_\sigma^j \longmapsto \bigvee_{\substack{1 \leq i \leq p-1 \\ j \leq p-i}} (\|t\|_\sigma^i \wedge \neg\|t\|_\sigma^{i+1})$$

$$\|t^{-1}\|_\sigma^j \longmapsto \bigvee_{\substack{1 \leq i,k \leq p-1 \\ j \leq k \wedge ik \equiv 1 \bmod p}} (\|t\|_\sigma^i \wedge \neg\|t\|_\sigma^{i+1})$$

We translate $\mathtt{e}(A,i,j)$ in the obvious way: $\|\mathtt{e}(A,i,j)\|_\sigma^k$ is $A_{ij}^k$; thus, with each matrix $A$ if size $m \times n$, we associate $m \cdot n \cdot p$ boolean variables $A_{ij}^k$. For constructed matrices, we do the following: $\|\mathtt{e}(\lambda ij\langle m,n,t\rangle, i, j)\|_\sigma^k \longmapsto \|t\|_\sigma^k$.

Finally, we deal with $\Sigma(A)$ as follows:

$$\|\Sigma(A)\|_\sigma^j \longmapsto \bigvee_{j \leq i \leq p-1} \mathtt{MOD}_{p,i}(\{A_{xy}^k\}_{1 \leq x \leq \sigma(\mathtt{r}(A)), 1 \leq y \leq \sigma(\mathtt{c}(A)), 1 \leq k \leq p})$$

We proceed in a similar way with constructed matrices. This ends the modification of the translation for fields $\mathbb{Z}_p$.

We will not present the details of the translation over the rationals since field operations over the integers (and we can encode the rationals as pairs of integers), have been already formalized using boolean formulas of size polynomial in the length of the encoding. See, for example, [Weg87, Theorems 1.2 and 1.3] for polysize circuits of depth $O(\log n)$ that compute the addition and multiplication of integers. Also, see [Pit00].

## 7.4 Translating theorems of LAP

In this section we are going to show how to translate theorems of LAP into families of boolean tautologies with quasi-poly-bounded Frege proofs. Again, we concentrate on the field $\mathbb{Z}_2$ (as before, when the underlying field is $\mathbb{Z}_p$, $p > 2$, or $\mathbb{Q}$, the translation is messier, but all the results still hold). The "quasi" prefix in poly-bounded comes from the fact that we require $\mathrm{NC}^2$ circuits to compute powers of matrices, and $\mathrm{NC}^2$ circuits correspond to boolean formulas of size $O(2^{\log^2 n})$.

Most of the work has been done already, when we translated LA into families of tautologies. In fact, we only need to show how to deal with $\mathtt{e}(\mathtt{P}(m, A), i, j)$ over $\mathbb{Z}_2$.

The first thing we do is show how to translate $\|\mathtt{e}(\mathtt{P}(m, A), i, j)\|_\sigma$ recursively:

**Base Case:** There are two cases:

**case 1** $\|m\|_\sigma = 0$. $\|\mathtt{e}(\mathtt{P}(m, A), i, j)\|_\sigma \longmapsto \mathtt{T}$ iff $\sigma(i) = \sigma(j)$.

**case 2** $\|m\|_\sigma = 1$. $\|\mathtt{e}(\mathtt{P}(m, A), i, j)\|_\sigma \longmapsto \|\mathtt{e}(A, i, j)\|_\sigma \longmapsto A_{ij}$.

**Recursive Step:** Assume $\|m\|_\sigma > 1$. There are two cases:

**case 1** $\|m\|_\sigma$ is even. Then $\|\mathtt{e}(\mathtt{P}(m, A), i, j)\|_\sigma$ is mapped to:

$$\mathtt{MOD}_{2,1}(\{\|\mathtt{e}(\mathtt{P}(\mathtt{div}(m, 2), A), i, k)\|_\sigma \wedge \|\mathtt{e}(\mathtt{P}(\mathtt{div}(m, 2), A), k, j)\|_\sigma\}_k)$$

where $k$ ranges in $\{1, \ldots, \max\{\|\mathtt{r}(A)\|_\sigma, \|\mathtt{c}(A)\|_\sigma\}\}$.

**case 2** $\|m\|_\sigma$ is odd. In this case, we effectively have to multiply three matrices; consider the $(i, j)$-th entry of the product $(AB)C$ (assume that all three are $n \times n$ matrices):

$$\sum_{k=1}^{n} (AB)_{ik} C_{kj} = \sum_{1 \le k \le n, 1 \le l \le n} ((A_{il} B_{lk}) C_{kj}) \tag{7.6}$$

If we now let $A$ and $B$ be $\mathtt{P}(\mathtt{div}(m-1, 2), A)$, and $C$ be $A$, and keep the right hand side of equation (7.6) in mind, we get the following translation:

$$\mathtt{MOD}_{2,1}(\{ (\|\mathtt{e}(\mathtt{P}(\mathtt{div}(m-1, 2), A), i, l)\|_\sigma \wedge \|\mathtt{e}(\mathtt{P}(\mathtt{div}(m-1, 2), A), l, k)\|_\sigma) \wedge \|\mathtt{e}(A, k, j)\|_\sigma \}_{l,k})$$

where $k$ and $l$ range in $\{1, \ldots, \max\{\|\mathtt{r}(A)\|_\sigma, \|\mathtt{c}(A)\|_\sigma\}\}$. Basically, the idea is that if $m$ is odd, then $m = 2n + 1$, and:

$$A^{\frac{m-1}{2}} A^{\frac{m-1}{2}} A = A^n A^n A = A^{2n+1} = A^m$$

We also have to show that axioms A34 and A35 (see Table 4.1 on page 45) translate to propositional tautologies with feasible quasi-polysize ($O(2^{\log^2 n})$) PK[2] derivations. This is easy to see for A34 (it follows from the base case of the above recursion). Showing it for A35:

$$\rightarrow \mathtt{P}(m + 1, A) = \mathtt{P}(m, A) * A$$

requires showing that:

$$\bigwedge_{i,j} \|\mathtt{e}(\mathtt{P}(m+1,A),i,j)\|_\sigma \equiv \|\mathtt{e}(\mathtt{P}(m,A)*A,i,j)\|_\sigma$$

and given the translation for $\mathtt{P}(m,A)$, this is just associativity of "$\wedge$".

# Chapter 8

# Proofs of the C-H Theorem

The main result of this chapter is a feasible proof of the Cayley-Hamilton Theorem. This result gives us a feasible proof of correctness of Berkowitz's algorithm, feasible proofs of hard matrix identities, and feasible proofs of the main principles of Matrix Algebra (specifically: axiomatic definition of the determinant, cofactor expansion formula, and multiplicativity of the determinant).

We present three feasible proofs of the C-H Theorem. The first, in Section 8.2.2, relies on translations from LAP with $\Pi_1^M$-Induction into a variant of the poly-time theory $\mathbf{V}_1^1$. The second and third, in Section 8.2.5 and Section 8.2.6, respectively, rely on translations from LAP with $\Pi_1^M$-Induction. We translate into families of Frege proofs with the permutation rule in one case, and introduction of propositional quantifiers in the second case. Permutation Frege is a fragment of Substitution Frege, which corresponds to reasoning with poly-time concepts. The fragment of Quantified Frege that we use is tree-like and all formulas only need one block of universal quantifiers, and this can be $p$-simulated by Extended Frege.

It seems that we provide the first feasible proof of correctness of the C-H Theorem. To support this claim we present in Section 8.1 the prototypical (infeasible) proofs of the C-H Theorem, given in algebra textbooks. They are both infeasible as they rely on the Lagrange formula for $\det(xI - A)$, which has $n!$ terms for an $n \times n$ matrix $A$.

In section 8.3 we give a feasible proof of correctness of the Gaussian Elimination Algorithm; a poly-time proof of a poly-time algorithm. This result is interesting in its own right because we do not know how to give a proof of correctness of Berkowitz's algorithm in its own complexity class. In other words, we do not know if we can prove the Cayley-Hamilton Theorem using $\mathrm{NC}^2$ concepts, rather than (feasible) poly-time concepts.

We use the proof of correctness of Gaussian Elimination to give a direct feasible proof (as opposed to an indirect proof via the feasible proof of the C-H Theorem) of $AB = I \rightarrow BA = I$ (Section 8.3.3), and a feasible proof of $\det(A) = 0 \rightarrow AB \neq I$ (Section 8.3.2). This last identity, together with a feasible proof of the Cayley-Hamilton Theorem, gives us a feasible proof of multiplicativity of determinant (see Section 6.4).

At this point, it is not known if there are poly-bounded Frege proofs, or even quasi-poly-bounded Frege proofs of hard matrix identities or of the Cayley-Hamilton Theorem. To repeat using the language of circuit complexity: we know that hard matrix identities, as well as the Cayley-Hamilton Theorem, have poly-bounded P/poly-Frege proofs, but it is not known if they have poly-bounded $\mathrm{NC}^i$-Frege proofs, for any $i$. Since Berkowitz's algorithm is an $\mathrm{NC}^2$ algorithm, it is tempting to conjecture that they all have $\mathrm{NC}^2$-Frege proofs.

## 8.1 Traditional proofs of the C-H Theorem

In this section we present *two* prototypical (infeasible) proofs of the Cayley-Hamilton Theorem that are given in one form or another in most Linear Algebra textbooks. Our (small) contribution is Claim 8.1.1 which is usually overlooked (and never proven) when Proof I is given. These proofs are infeasible because they rely on the Lagrange expansion of the determinant; the Lagrange expansion, for a $n \times n$ matrix, has $n!$ terms (i.e., it is the summation over all the permutations of $n$ elements: $\sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) a_{1\sigma(1)} a_{2\sigma(2)} \ldots a_{n\sigma(n)}$).

### 8.1.1 Infeasible proof of the C-H Theorem (I)

*Proof.* The most direct proof is using the Lagrange expansion on $\det(xI - A)$. We show that using the Lagrange expansion we can prove the axiomatic definition of the determinant, and hence, by defining $\mathrm{adj}(xI - A)$ as the matrix of cofactors, we can prove that:

$$(xI - A)\mathrm{adj}(xI - A) = \det(xI - A)I.$$

Let $\pi$ be the isomorphism that maps objects from $M(\mathbb{F}[x])$ to $(M(\mathbb{F}))[x]$, and replace $x$ by $A$ in the expression $(xI - A)\pi(\mathrm{adj}(xI - A)) = \pi(\det(xI - A)I)$. This gives us that $p_A(A) = \pi(\det(xI - A))|_{x=A} = 0$. □

Just for completeness we show that the map $\pi$, given in the above proof exists. This

map allows us to consider every matrix whose entries are polynomials over the ring $R$, as a polynomial whose coefficients are in $M_{n\times n}(R)$.

**Example 8.1.1** Suppose that $n = 2$ and $R = \mathbb{Q}$. Consider:

$$\begin{pmatrix} 2x+5 & -x^2-7 \\ 3 & x^2+5x+1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix} x^2 + \begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix} x + \begin{pmatrix} 5 & -7 \\ 3 & 1 \end{pmatrix}$$

where the LHS is in $M_{2\times 2}(\mathbb{Q}[x])$ and the RHS is in $(M_{2\times 2}(\mathbb{Q}))[x]$.

**Claim 8.1.1** $M_{n\times n}(R[x]) \cong (M_{n\times n}(R))[x]$.

*Proof.* Let $\pi : M_{n\times n}(R[x]) \longrightarrow (M_{n\times n}(R))[x]$ be the natural mapping. We show first that $\pi$ is a ring homomorphism. Let $(\{a_{ij}^k\}) \in M_{n\times n}(R[x])$ where $\{a_{ij}^k\} \in R[x]$ is the $(i,j)$-th entry, and $a_{ij}^k$ is the coefficient of the $k$-th power of this polynomial. Then:

$$\begin{aligned} \pi((\{a_{ij}^k\}) + (\{b_{ij}^k\})) &= \pi((\{a_{ij}^k + b_{ij}^k\})) \\ &= \{(a_{ij}^k + b_{ij}^k)\} \\ &= \{(a_{ij}^k)\} + \{(b_{ij}^k)\} \\ &= \pi((\{a_{ij}^k\})) + \pi((\{b_{ij}^k\})) \end{aligned}$$

Now we want to show that $\pi((\{a_{ij}^k\}) \cdot (\{b_{ij}^k\})) = \pi((\{a_{ij}^k\})) \cdot \pi((\{b_{ij}^k\}))$.

$$\begin{aligned} \pi((\{a_{ij}^k\}) \cdot (\{b_{ij}^k\})) &= \pi((\Sigma_{l=1}^n \{a_{il}^k\} \cdot \{b_{lj}^k\})) \\ &= \pi((\Sigma_{l=1}^n \{\Sigma_{r+s=k} a_{il}^r b_{lj}^s\})) \\ &= \{\Sigma_{r+s=k} (\Sigma_{l=1}^n a_{il}^r b_{lj}^s)\} \\ &= \{\Sigma_{r+s=k} (a_{ij}^r) \cdot (b_{ij}^s)\} \\ &= \{(a_{ij}^k)\} \cdot \{(b_{ij}^k)\} \\ &= \pi((\{a_{ij}^k\})) \cdot \pi((\{b_{ij}^k\})) \end{aligned}$$

Thus $\pi$ is a ring homomorphism. Since $\pi$ is bijective, the claim follows. □

### 8.1.2 Infeasible proof of the C-H Theorem (II)

Another proof, more algebraic, considers algebraically closed fields. So let $A$ be a matrix, let $p_A(x) = \det(xI - A)$ (also computed via the Lagrange expansion), and let $\lambda_1, \lambda_2, \ldots, \lambda_n$

be the eigenvalues of $A$. As we are dealing with algebraically closed fields, we can find a matrix $P$ such that:

$$A = P^{-1}TP$$

where $T$ is an upper-triangular matrix with $\lambda_1, \lambda_2, \ldots, \lambda_n$ on the diagonal. We factor the characteristic polynomial as follows:

$$p_A(x) = (x - \lambda_1)(x - \lambda_2) \cdots (x - \lambda_n)$$

Then:

$$p_A(A) = p_A(P^{-1}TP) = P^{-1}p_A(T)P$$

and note that:

$$p_A(T) = (T - \lambda_1 I)(T - \lambda_2 I) \cdots (T - \lambda_n I)$$

Now, using induction on $k$, we can show that the first $k$ columns of $(T-\lambda_1 I) \cdots (T-\lambda_k I)$ are zero, $1 \leq k \leq n$. Thus, $p_A(T) = 0$, and therefore $p_A(A) = 0$. □

## 8.2 Feasible proofs of the C-H Theorem

In this section we present *three* feasible proofs of the Cayley-Hamilton Theorem. All three proofs rely on the same idea—they are the same proof expressed in slightly different ways to draw connections with different propositional proof systems.

The basic intuition behind these proofs is the following: if $p_A(A) \neq 0$, that is, if the C-H Theorem fails for $A$, then we can find *in polytime* a minor $A[i|j]$ of $A$ for which $p_{A[i|j]}(A[i|j]) \neq 0$, i.e., a minor for which the C-H Theorem fails already. Since the C-H Theorem does *not* fail for $1 \times 1$ matrices, after $n =$ (size of $A$) steps we get a contradiction. This idea can be expressed with universal quantifiers over variables of type matrix: if the C-H Theorem holds for all $n^2$ minors of $A$, it also holds for $A$.

Also note that we do not need multiplicative inverses for field elements to prove the C-H Theorem; that is, we do not need the function $^{-1}$ and we do not need axiom A21. Berkowitz's algorithm does not compute inverses of field elements, and we do not need to take inverses in the proofs of the C-H Theorem that we present below. Thus, the C-H Theorem holds for commutative rings. On the other hand, we *do* use inverses in our proof of the multiplicativity of the determinant[1] (see Section 6.4).

[1]It is an interesting question whether it is possible to prove the multiplicativity of the determinant for commutative rings.

Here are the outlines of the proofs:

- **Proof 1 with $\mathbf{V}^1_1$:** In Section 8.2.1 we show that LAP, with $\forall$ quantifiers for matrix variables (i.e., we allow $\Pi^M_1$ formulas—see Definition 8.2.1) and augmented by $\Pi^M_1$-Induction (we call the new theory $\forall$LAP), proves the C-H Theorem. In Section 8.2.2 we define the theory $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$, which is conservative over the poly-time theory $\mathbf{V}^1_1$, and in Section 8.2.3 we show that the theorems of $\forall$LAP can be interpreted in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$.

- **Proof 2 with Permutation Frege:** In Section 8.2.5 we use the machinery of LAP to show that the C-H Theorem has poly-bounded *uniform* families of Permutation Frege proofs. Since Permutation Frege is a fragment of Substitution Frege, which in turn can be $p$-simulated by Extended Frege, once again we have feasible proofs of the C-H Theorem.

- **Proof 3 with Quantified Frege:** In Section 8.2.6 we show that $\forall$LAP proofs translate into uniform, tree-like, poly-bounded families of Quantified Frege proofs, where all formulas either have no quantifiers at all, or a block of universal quantifiers. This fragment of Quantified Frege can be $p$-simulated by Extended Frege.

Since LAP proves the equivalence of the C-H Theorem, the axiomatic definition of the determinant, and the cofactor expansion (see Chapter 6), we obtain that these principles have feasible proofs as well.

Finally, in Section 8.3.2, we show that a feasible proof of the C-H Theorem implies that identity (6.16) has a feasible proof as well. From Section 6.4 we know that (6.16) is enough to prove in LAP the multiplicativity of the determinant. Therefore, we can conclude that the multiplicativity of det has feasible proofs also.

### 8.2.1 LAP augmented by $\Pi^M_1$-Induction: $\forall$LAP

**Definition 8.2.1** We define $\Pi^M_0$ to be the set of formulas over $\mathcal{L}_{\mathrm{LAP}}$ ("$M$" stands for matrix). We define $\Pi^M_1$ to be the set of formulas in $\Pi^M_0$ together with formulas of the form $(\forall A \leq n)\alpha$, where $\alpha \in \Pi^M_0$, and where $(\forall A \leq n)\alpha$ abbreviates:

$$(\forall A)((\mathtt{r}(A) \leq n \wedge \mathtt{c}(A) \leq n) \supset \alpha)$$

where $A$ is a matrix variable, *not* contained in the index term $n$.

**Definition 8.2.2** We define the proof system LK-∀LAP to be the same as PK-LAP, but where we allow $\Pi_1^M$ formulas (hence we use LK rather then PK). Thus we need two more rules for introducing a universal quantifier on the left and on the right of a sequent: see Table 8.1, where $T$ is any term of type matrix, and $n$ is any term of type index. Also,

$$\text{left} \quad \frac{\mathtt{r}(T) \leq n, \mathtt{c}(T) \leq n, \alpha(T), \Gamma \rightarrow \Delta}{(\forall X \leq n)\alpha(X), \Gamma \rightarrow \Delta} \qquad \text{right} \quad \frac{\mathtt{r}(A) \leq n, \mathtt{c}(A) \leq n, \Gamma \rightarrow \Delta, \alpha(A)}{\Gamma \rightarrow \Delta, (\forall X \leq n)\alpha(X)}$$

Table 8.1: ∀-introduction in LK-∀LAP

in ∀-introduction-right, $A$ is a variable of type matrix that does not occur in the lower sequent, **and** $\alpha$ is a $\Pi_0^M$ formula, because we just want a single matrix quantifier.

**Definition 8.2.3** The theory ∀LAP is the set of sequents with formulas in $\Pi_1^M$ which have LK-∀LAP derivations. In particular, ∀LAP has induction over $\Pi_1^M$ formulas, henceforth $\Pi_1^M$-IND.

$\Pi_1^M$-IND is what allows us to prove the C-H Theorem.

Note that instead of ∀LAP, we could have proceeded by allowing alternation of quantifiers (with ∃-introduction), and proving a Cut-Elimination Theorem.

**Theorem 8.2.1** ∀LAP proves the Cayley-Hamilton Theorem.

*Proof.* We prove that for all $n \times n$ matrices $A$, $p_A(A) = 0$, by induction on $n$. The **Basis Case** is trivial: if $A = (a_{11})$, then the char poly of $A$ is $x - a_{11}$. We use the following strong induction hypothesis: $(\forall A \leq n)p_A(A) = 0$. Thus, in our **Induction Step** we prove:

$$(\forall M \leq n)p_M(M) = 0 \rightarrow (\forall A \leq n+1)p_A(A) = 0 \tag{8.1}$$

So let $A$ be an $(n+1)\times(n+1)$ matrix, and assume that we have $(\forall M \leq n)p_M(M) = 0$. Then, by Corollary 6.1.1, we have that for all $1 \leq i < j \leq n-1$, $p_{(I_{ij}AI_{ij})} = p_A$.

Suppose now that the $i$-th row (column) of $p_A(A)$ is not zero. Then, the first row (column) of $I_{1i}p_A(A)I_{1i}$ is not zero. But:

$$I_{1i}p_A(A)I_{1i} = p_A(I_{1i}AI_{1i}) = p_{(I_{1i}AI_{1i})}(I_{1i}AI_{1i})$$

and the first row and column of $p_{(I_{1i}AI_{1i})}(I_{1i}AI_{1i})$ are zero by Lemma 8.2.1 below (letting $C = I_{1i}AI_{1i}$). Thus, contradiction; it follows that $p_A(A) = 0$. This argument can be clearly formalized in ∀LAP. □

**Lemma 8.2.1** LAP proves that if $p_{C[1|1]}(C[1|1]) = 0$, then the first row and the first column of $p_C(C)$ are zero[2].

*Proof.* We restate the Lemma using the usual notation of $A$ and $M = A[1|1]$. Thus, we want to show that LAP proves the following: if $p_M(M) = 0$, then the first row and the first column of $p_A(A)$ are zero. For clarity we let $p = p_A$ and $q = p_M$.

The proof is by induction on the size of $M$. The **Basis Case** is when $M$ is a $1 \times 1$ matrix. Let $p_2, p_1, p_0$ be the coefficients of the char poly of $A$, and let $q_1, q_0$ be the coefficients of the char poly of $M$. By assumption $q_1 M + q_0 I = 0$. Note that $I$ is also a $1 \times 1$ matrix. From Berkowitz's algorithm we know that:

$$\begin{pmatrix} p_2 \\ p_1 \\ p_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -a_{11} & 1 \\ -RS & -a_{11} \end{pmatrix} \begin{pmatrix} q_1 \\ q_0 \end{pmatrix} = \begin{pmatrix} q_1 \\ -a_{11}q_1 + q_0 \\ -RSq_1 - a_{11}q_0 \end{pmatrix} \tag{8.2}$$

Note that:

$$A^2 = \begin{pmatrix} a_{11}^2 + RS & a_{11}R + RM \\ a_{11}S + MS & SR + M^2 \end{pmatrix}$$

We must now show that the first row and column of $p_A(A) = p_2 A^2 + p_1 A + p_0 I$ are zero. We just show that the $(1,2)$ entry is zero; the rest follow just as easily. From (8.2) we see that the $(1,2)$ entry of $p_A(A)$ is given by:

$$(a_{11}R + RM)q_1 + R(-a_{11}q_1 + q_0) + 0(-RSq_1 - a_{11}q_0) = R(Mq_1 + q_0) = 0$$

Note that it is actually possible, in the Basis Case, to show that $p_A(A) = 0$ (as this is *true*), not just the first row and column of $p_A(A)$. However, this seems infeasible to carry out in the Induction Step.

We prove the **Induction Step** with three claims. We indicate in Figure 8.1, which claim corresponds to which entries in the first row and column of $p_A(A)$.

We assume that $M$ is an $(n-1) \times (n-1)$ matrix, where $n - 1 \geq 1$. We let $p = p_A$ and $q = p_M$, that is, $p, q$ are the char polys of $A, M = A[1|1]$, respectively. Define

[2]The original hope was that from the assumption that $p_{C[1|1]}(C[1|1]) = 0$ it would be possible to show that $p_C(C) = 0$; unfortunately, this seems to be too weak an induction hypothesis. Hence we introduced the universal quantifiers over matrices to overcome this weakness, but then also the complexity of the proof jumped from quasi-polybounded Frege to polybounded Extended Frege.

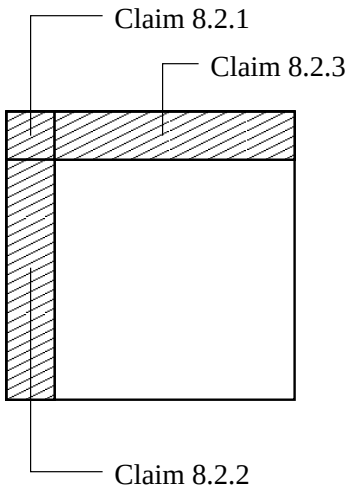



Figure 8.1: Shaded area of $p_A(A)$ is zero

$w_k, X_k, Y_k, Z_k$ as follows:

$$A = \begin{pmatrix} w_1 & X_1 \\ Y_1 & Z_1 \end{pmatrix} = \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix}$$

$$A^{k+1} = \begin{pmatrix} w_{k+1} & X_{k+1} \\ Y_{k+1} & Z_{k+1} \end{pmatrix} = \begin{pmatrix} w_k & X_k \\ Y_k & Z_k \end{pmatrix} \begin{pmatrix} a_{11} & R \\ S & M \end{pmatrix} \quad \text{for } k \geq 1$$

Note that $w_k, X_k, Y_k, Z_k$ cannot be defined in LAP as we cannot define new matrices recursively. However, all that we need in the following proof are entries of powers of $A$, which can be expressed in LAP. The entry $w_k$, and the submatrices $X_k, Y_k, Z_k$ are there to make the proof more human readable; for example, instead of $w_k$ we could write $\mathtt{e}(\mathtt{P}(k, A), 1, 1)$, or instead of $X_k$ we could write $\lambda ij\langle 1, n-1, \mathtt{e}(\mathtt{P}(k, A), 1, j+1)\rangle$, but then the proof would be difficult to read.

It is easy to see that LAP proves the following equations:

$$\begin{aligned} w_{k+1} &= a_{11}w_k + X_k S \\ X_{k+1} &= w_k R + X_k M \\ Y_{k+1} &= a_{11}Y_k + Z_k S \\ Z_{k+1} &= Y_k R + Z_k M \end{aligned} \tag{8.3}$$

As was mentioned above, we are going to prove that the first row and column consist of zeros with Claims 8.2.1, 8.2.2, and 8.2.3. Claim 8.2.3 follows from Claim 8.2.2 using the fact that $A$ and $A^t$ have the same char poly (the details are provided in the proof

of Claim 8.2.3). For the other two claims we are going to put $p_A(A)$ in a special form. Using Berkowitz's algorithm, it is easy to show in LAP that:

$$p(A) = (A - a_{11}I)q(A) - \sum_{k=1}^{n-1} q_k \sum_{i=0}^{k-1} (RM^iS)A \tag{8.4}$$

and thus, to show that the first column of $p(A)$ is zero, it is enough to show that the first columns of $(A - a_{11}I)q(A)$ and $\sum_{k=1}^{n-1} q_k \sum_{i=0}^{k-1}(RM^iS)A$ are the same. This is the strategy for proving Claims 8.2.1 and 8.2.2.

**Claim 8.2.1** The upper-left entry of $p(A)$ is zero.

*Proof.* Using (8.3) we obtain:

$$\begin{cases} w_0 = 1 \\ w_1 = a_{11} \\ w_{k+1} = a_{11}w_k + \sum_{i=0}^{k-1}(RM^iS)w_{k-1-i} \quad \text{for } k \geq 1 \end{cases} \tag{8.5}$$

The top left entry of $(A - a_{11}I)q(A)$ is given by

$$\sum_{k=1}^{n-1} q_k(w_{k+1} - a_{11}w_k) \tag{8.6}$$

(notice that we can ignore the term $k = 0$ since the top left entry of $A$ is the same as the top left entry of $a_{11}I$). We can compute $(w_{k+1} - a_{11}w_k)$ using the recursive definitions of $w_k$ (given by (8.5) above):

$$\begin{aligned} w_{k+1} - a_{11}w_k &= a_{11}w_k + \sum_{i=0}^{k-1}(RM^iS)w_{k-1-i} - a_{11}w_k \\ &= \sum_{i=0}^{k-1}(RM^iS)w_{k-1-i} \end{aligned}$$

Thus, (8.6) is equal to

$$\sum_{k=1}^{n-1} q_k \sum_{i=0}^{k-1}(RM^iS)w_{k-1-i}$$

This proves that the top left entry of $p(A)$ is zero (see equation (8.4) and the explanation below it). □

**Claim 8.2.2** The $(n-1) \times 1$ lower-left submatrix of $p(A)$ is zero.

*Proof.* Using (8.3) we obtain:

$$\begin{cases} Y_0 = 0 \\ Y_1 = S \\ Y_{k+1} = a_{11}Y_k + (M^k S) + \sum_{i=0}^{k-2}(RM^i S)Y_{k-1-i} \quad \text{for } k \geq 1 \end{cases} \tag{8.7}$$

The lower-left $(n-1) \times 1$ submatrix of $(A - a_{11}I)q(A)$ is given by

$$\sum_{k=0}^{n-1} q_k(Y_{k+1} - a_{11}Y_k)$$

and by (8.7) we have that for $k \geq 2$,

$$\begin{aligned} Y_{k+1} - a_{11}Y_k &= \left(a_{11}Y_k + M^k S + \sum_{i=0}^{k-2}(RM^i S)Y_{k-1-i}\right) - a_{11}Y_k \\ &= M^k S + \sum_{i=0}^{k-2}(RM^i S)Y_{k-1-i} \end{aligned}$$

Therefore:

$$\begin{aligned} \sum_{k=0}^{n-1} q_k(Y_{k+1} - a_{11}Y_k) &= q_0(Y_1 - a_{11}Y_0) + q_1(Y_2 - a_{11}Y_1) \\ &\quad + \sum_{k=2}^{n-1} q_k \left(M^k S + \sum_{i=0}^{k-2}(RM^i S)Y_{k-1-i}\right) \\ &= q(M)S + \sum_{k=2}^{n-1} q_k \sum_{i=0}^{k-2}(RM^i S)Y_{k-1-i} \end{aligned}$$

and by the IH, $\sum_{k=0}^{n-1} M^k S = q(M)S = 0$, and by definition $Y_0 = 0$, thus we can conclude that:

$$\sum_{k=0}^{n-1} q_k(Y_{k+1} - a_{11}Y_k) = \sum_{k=1}^{n-1} q_k \sum_{i=0}^{k-1}(RM^i S)Y_{k-1-i}$$

But the RHS of the above equation is equal to the $(n-1) \times 1$ lower-left submatrix of $\sum_{k=1}^{n-1} q_k \sum_{i=0}^{k-1}(RM^i S)A^{k-1-i}$, and hence the claim follows (once again, see equation (8.4) and the explanation below it). □

**Claim 8.2.3** The $1 \times (n-1)$ upper-right submatrix of $p(A)$ is zero.

*Proof.* To prove this claim we use Lemma 5.1.7 and Claim 8.2.2. The crucial observation is that the $(n-1)\times 1$ lower-left submatrix of $(A^t)^k$ is $X_k^t$. Now, we know by Lemma 5.1.7 that $p$ is also the char polynomial of $A^t$, so by Claim 8.2.2, we know that the $(n-1)\times 1$ lower-left submatrix of $p(A^t)$ is zero. Thus the $(n-1)\times 1$ lower-left submatrix of $(p(A))^t$ is zero, and therefore the $1\times(n-1)$ upper-right submatrix of $p(A)$ is zero, and hence the claim follows. □

This ends the proof of the Lemma 8.2.1. □

### 8.2.2 The theory $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$

The theory $\tilde{\mathbf{V}}^1$, defined in [Coo98, p.41], has the same theorems as $\mathbf{V}^1$, defined in [Coo98, Section 10]. Both theories correspond to $\mathbf{V}_1^1$ in [Kra95, Section 5.5]. All these theories, $\tilde{\mathbf{V}}^1, \mathbf{V}^1, \mathbf{V}_1^1$ are close variants of each other, and they all correspond to feasible (i.e., poly-time) reasoning. The theory $\tilde{\mathbf{V}}^1$ has been inspired by Domenico Zambella ([Zam96]). A good treatment of a $\tilde{\mathbf{V}}^1$-like theory can also be found in [CK01].

The theory $\tilde{\mathbf{V}}^1$ is a second order theory over the language $\mathcal{L}_A^2 = [0, 1, +, \cdot, ||; \in, \leq, =]$. There are number variables $x, y, z, \ldots$ and string variables $X, Y, Z, \ldots$. Here $|X|$ is intended to denote the length of the string $X$, and $X(t)$ abbreviates $t \in X$, where $t$ is a number term. Equating 1 with $\mathtt{T}$ and 0 with $\mathtt{F}$, as usual, we think of $X$ as a binary string:

$$X(0)X(1)X(2)\ldots X(n-1)$$

where $n = |X|$. See [Coo98, Section 9] for the formal syntax of the terms and formulas over $\mathcal{L}_A^2$.

The axioms of $\tilde{\mathbf{V}}^1$ are given in Table 8.2. We extend $\tilde{\mathbf{V}}^1$ by adding two function symbols to $\mathcal{L}_A^2$: $\Sigma$ and $\mathtt{P}$, which take strings to strings. The defining axioms of $\Sigma$ and $\mathtt{P}$ in LAP are A30–A35, so we add the translations of these axioms to $\tilde{\mathbf{V}}^1$. In Section 8.2.3 we will show that the translations of A30–A35 are $\Sigma_0^B$ formulas (see Definition 8.2.4). The resulting theory $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ is a conservative[3] extension of $\tilde{\mathbf{V}}^1$, and therefore $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ also corresponds to feasible (i.e., poly-time) reasoning[4]. Note that since $\Sigma$ and $\mathtt{P}$ take strings to strings, the sorts of $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ are still indices and strings.

[3]Here the idea is that if we add poly-time function symbols to $\tilde{\mathbf{V}}^1$, plus $\Sigma_1^B$ defining axioms for these new function symbols, then the resulting theory is still poly-time, and a conservative extension of $\tilde{\mathbf{V}}^1$. A source for this observation is [Bus86], where the analogous result is shown for $S_2^1$.

[4]Buss's RSUV isomorphism.

We form terms over $\mathcal{L}_A^2 \cup \{\Sigma, \mathtt{P}\}$ by forming terms over $\mathcal{L}_A^2$ (see [Coo98, p. 32]), and by adding three more cases:

- A string variable $X$ is a string term.
- If $T$ is a string term, then $\Sigma(T)$ is a string term. The intended meaning is the following:

$$\Sigma(T) = \begin{cases} |\Sigma(T)| = 1 \text{ and } T(0) = 1 & \text{if parity of } T \text{ is odd} \\ |\Sigma(T)| = 0 & \text{if parity of } T \text{ is even} \end{cases}$$

  Thus, the empty string corresponds to the case where the parity of $T$ is even, and a string consisting of a single 1 corresponds to the case where the parity of $T$ is odd. Therefore we need to add the following axiom: $|\Sigma(X)| \leq 1$.

- If $T$ is a string term, and $n$ is a number term, then $\mathtt{P}(T, n)$ is a string term. The intended meaning of $\mathtt{P}$ is the following: if $X_A$ is the string variable corresponding to the matrix $A$, then $\mathtt{P}(X_A, n) = X_{A^n}$. In the next section we explain the correspondence $A \leftrightarrow X_A$.

We define formulas over $\mathcal{L}_A^2 \cup \{\Sigma, \mathtt{P}\}$ as formulas over $\mathcal{L}_A^2$ (see [Coo98, p. 33]). That is, if $t, u$ are number terms, and $T$ is a string term, then $t \in T$ (abbreviated by $T(t)$), $t \leq u$, $t = u$ are atomic formulas. If $T, U$ are string terms, then $T = U$ is not a formula, however we can take it as abbreviation of:

$$|T| = |U| \wedge (\forall z \leq |T|)(T(z) \leftrightarrow U(z))$$

Note that the universe for a model for $\mathcal{L}_A^2 \cup \{\Sigma, \mathtt{P}\}$ consists of two non-empty sets $U_1$ and $U_2$, for number objects and string objects respectively. An element $\alpha \in U_2$ can be specified as a pair $(|\alpha|, S_\alpha)$, where $S_\alpha = \{u \in U_1 | u \in \alpha\}$.

**Definition 8.2.4** We define the following classes of formulas over $\mathcal{L}_A^2 \cup \{\Sigma, \mathtt{P}\}$:

- $\Sigma_0^B = \Pi_0^B$ is the set of formulas such that all number quantifiers are bounded, and there are no string quantifiers. (There may be free string variables.) Incidentally, note that all the theorems of LA, which do not require $\Sigma$ in their proofs, can be be translated to $\Sigma_0^B$ formulas and proven in $\mathbf{V}^0$—that is, they have $\mathrm{AC}^0$ proofs (see [Coo98, Section 10] for the definition of the theory $\mathbf{V}^0$).

- $\Sigma_1^B$ is the set of formulas $\exists X \leq tB$, where $B$ is a $\Sigma_0^B$ formula, together with all $\Sigma_0^B$ formulas. Here $\exists X \leq tB$ stands for $\exists X(|X| \leq t \wedge B)$, where the term $t$ does not involve $X$.

- $\Pi_1^B$ is the set of formulas $\forall X \leq tB$, where $B$ is a $\Pi_0^B$ formula, together with all $\Pi_0^B$ formulas. Here $\forall X \leq tB$ stands for $\forall X(|X| \leq t \supset B)$, where the term $t$ does not involve $X$.

| | |
|---|---|
| B1 | $x + 1 \neq 0$ |
| B2 | $x + 1 = y + 1 \supset x = y$ |
| B3 | $x + 0 = x$ |
| B4 | $x + (y + 1) = (x + y) + 1$ |
| B5 | $x \cdot 0 = 0$ |
| B6 | $x \cdot (y + 1) = (x \cdot y) + x$ |
| B7 | $x \leq x + y$ |
| B8 | $(x \leq y \wedge y \leq x) \supset x = y$ |
| B9 | $0 \leq x$ |
| B10 | $x \leq y \equiv x < y + 1$ |
| L | $X(y) \supset y < \|X\|$ |
| IND | $(X(0) \wedge \forall y < z(X(y) \supset X(y + 1))) \supset X(z)$ |
| $\Sigma_1^B$-IND | $(A(0) \wedge \forall x(A(x) \supset A(x + 1))) \supset A(y)$ |
| $\Sigma_0^B$-COMP | $\exists X \leq y \forall z < y(\|X\| = y \wedge (X(z) \equiv A(z))$ |

Table 8.2: The axioms of $\tilde{\mathbf{V}}^1$

**Lemma 8.2.2** The theory $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ has $\Pi_1^B$-IND as well. That is, for all formulas $A(x)$, where $A(x)$ is $\forall X \leq t(x)B(x, X)$ for some $B \in \Pi_0^B$, we have:

$$\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P}) \vdash (A(0) \wedge \forall x(A(x) \supset A(x + 1))) \supset A(y)$$

*Proof.* This is a well known result ($\mathbf{V}_1^1$ and $S_2^1$ prove $\Pi_1^B$-IND and $\Pi_1^b$-IND, respectively).

Suppose that we are given:

$$\forall X \leq t(0)B(0,X) \tag{8.8}$$

$$\wedge \quad (\forall X \leq t(x)B(x,X) \supset \forall X \leq t(x+1)B(x+1,X)) \tag{8.9}$$

we want to prove:

$$\forall X \leq t(y)B(y,X) \tag{8.10}$$

For the sake of contradiction, assume the negation of (8.10), and replace $y$ by $y-0$ to obtain:

$$\exists X \leq t(y-0)\neg B(y-0,X) \tag{8.11}$$

Taking the contrapositive of (8.9), and replacing $x$ by $y-(z+1)$, where $z$ is a *new* variable, we obtain:

$$\forall z(\exists X \leq t(y-z)\neg B(X,y-z) \supset \exists X \leq t(y-(z+1))\neg B(X,y-(z+1))) \tag{8.12}$$

Using $\Sigma_1^B$-IND on (8.11) and (8.12) we can conclude $\exists X \leq t(y-w)\neg B(y-w,X)$, for any $w$. Taking $w=y$ we get the negation of (8.8), and hence a contradiction. □

### 8.2.3 Interpreting $\forall$LA$\mathtt{P}$ in $\tilde{\mathbf{V}}^1(\Sigma,\mathtt{P})$

We are going to interpret the theorems of $\forall$LA$\mathtt{P}$ in the theory $\tilde{\mathbf{V}}^1(\Sigma,\mathtt{P})$, which is a conservative extension of the poly-time theory $\tilde{\mathbf{V}}^1$. This will show that the theorems of $\forall$LA$\mathtt{P}$ have feasible proofs.

To do this, we translate the sequents over $\mathcal{L}_{\mathrm{LA}\mathtt{P}}$ into formulas over the language of $\tilde{\mathbf{V}}^1(\Sigma,\mathtt{P})$. The details of this translation are given below. We let the underlying field be $\mathbb{Z}_2$, as this is the simplest case, but extending the result to more general fields is not difficult; it only requires a more complicated scheme for translating field elements.

This translation begs the following question: why not formalize Linear Algebra in $\tilde{\mathbf{V}}^1$ (or similar system) directly? The answer is that the advantage of LA$\mathtt{P}$ is that it is *field independent*, whereas field elements would have to be encoded explicitly in a theory like $\tilde{\mathbf{V}}^1$. Also, LA$\mathtt{P}$ is very natural for expressing concepts of Linear Algebra, and apparently weaker than $\tilde{\mathbf{V}}^1$ (it corresponds to the complexity class DET rather than poly-time).

Recall that $\Pi_0^M$ is the set of formulas over $\mathcal{L}_{\mathrm{LAP}}$ *without* quantifiers, and $\Pi_0^B$ is the set of formulas over $\mathcal{L}_A^2$, also *without* quantifiers, and that the underlying field is assumed to be $\mathbb{Z}_2$.

The translation $\|\cdot\| : \Pi_0^M \longmapsto \Pi_0^B$ preserves provability; that is, if $\alpha \in \mathrm{LAP}$, then $\|\alpha\| \in \tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$. (Recall that the theorems of LAP are sequents, so $\alpha$ is really the formula corresponding to the sequent). $\|\cdot\|$ preserves all logical connectives, as well as $=$ for all three sorts, and $\leq$ for indices. The fact that $\|\cdot\|$ preserves provability follows from the following two observations:

- If $\alpha$ is an axiom of LAP, then $\|\alpha\|$ can be proven in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$. At a high level, this can be seen from the fact that index axioms are very similar to the number axioms, the field axioms correspond to properties of boolean connectives, and the translations of the axioms for $\Sigma$ and $\mathtt{P}$ have been added to $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$.

- The rules of inference of LAP can be simulated easily in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$. For example, the LAP induction rule corresponds to $\Sigma_0^B$-IND which we have in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$.

Thus, we are just going to provide the details of the translations of the terms.

Terms of type index will be translated into number terms. Field elements and matrices will be translated into strings: given a field element $a$, we associate with it the string variable $X_a$, such that $|X_a| = 1$, and given a matrix variable $A$, we associate with it the string variable $X_A$, such that $|X_A| = r_A \cdot c_A$, where $r_A, c_A$ are $\|\mathtt{r}(A)\|, \|\mathtt{c}(A)\|$, respectively.

Here is how we translate (recursively) terms of type index into number terms:

$$\begin{aligned}
\|i\| &\longmapsto i \\
\|m+n\| &\longmapsto \|m\| + \|n\| \\
\|m-n\| &\longmapsto \|m\| - \|n\| \\
\|m*n\| &\longmapsto \|m\| \cdot \|n\| \\
\|\mathtt{div}(m,n)\| &\longmapsto \mathtt{div}(\|m\|, \|n\|) \\
\|\mathtt{rem}(m,n)\| &\longmapsto \mathtt{rem}(\|m\|, \|n\|)
\end{aligned}$$

Note that $\{-, \mathtt{div}, \mathtt{rem}\}$ are not function symbols in $\mathcal{L}_A^2$, but they can all be defined by

$\Sigma_0^B$ formulas. For example $z = \mathtt{div}(x, y)$ iff $\exists w \leq y(z \cdot y + w = x)$. Furthermore:

$$\begin{aligned} \|\mathtt{r}(A)\| &\longmapsto r_A \\ \|\mathtt{c}(A)\| &\longmapsto c_A \\ \|\mathtt{r}(\lambda ij\langle m, n, t\rangle)\| &\longmapsto \|m\| \\ \|\mathtt{c}(\lambda ij\langle m, n, t\rangle)\| &\longmapsto \|n\| \end{aligned}$$

where $r_A, c_A$ are two new number variables.

Translating $\mathrm{cond}(\beta, m, n)$ is a little bit more complicated. We do it as follows: say that $\mathrm{cond}(\beta, m, n)$ occurs in an atomic formula $\alpha$. Then, we split $\alpha$ into two copies. In $\alpha_1$, we replace $\mathrm{cond}(\beta, m, n)$ by $m$, and in $\alpha_2$, we replace $\mathrm{cond}(\beta, m, n)$ by $n$. Then, we let $\|\alpha\|$ be:

$$(\|\beta\| \wedge \|\alpha_1\|) \vee (\neg\|\beta\| \wedge \|\alpha_2\|)$$

We translate terms of type field as follows:

$$\begin{aligned} \|a\| &\longmapsto X_a(1) \\ \|t + u\| &\longmapsto \|t\| \oplus \|u\| \\ \|t * u\| &\longmapsto \|t\| \wedge \|u\| \\ \|-t\| &\longmapsto \|t\| \\ \|t^{-1}\| &\longmapsto \|t\| \end{aligned}$$

Since terms of type field are translated into formulas over $\mathcal{L}_A^2$, we can translate the field term $\mathrm{cond}(\beta, t, u)$ as follows:

$$(\|\beta\| \wedge \|t\|) \vee (\neg\|\beta\| \wedge \|u\|)$$

We associate the string variable $X_A$ with the matrix variable $A$, and we let $|X_A| = r_A \cdot c_A$. We use the standard pairing function,

$$\langle i, j\rangle := \min z \leq (2i + j)^2(2z = (i + j)(i + j + 1) + 2i),$$

to encode a matrix $A$ over $\mathbb{Z}_2$ as a string of 0s and 1s. See Figure 8.2. Thus $\|\mathtt{e}(A, m, n)\|$ is mapped to:

$$X_A(\langle\|m\| - 1, \|n\| - 1\rangle) \wedge (\|m\| < r_A) \wedge (\|n\| < c_A)$$

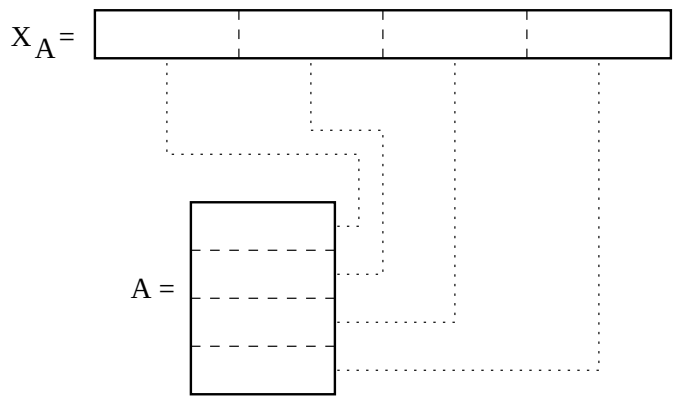



Figure 8.2: If $A$ is $4 \times 3$, then $|X_A| = 12$

We use $\Sigma_0^B$-COMP to translate constructed terms. So let $\lambda ij\langle m,n,t\rangle$ be a matrix term; we define $X_{\lambda ij\langle m,n,t\rangle}$ as follows:

$$X_{\lambda ij\langle m,n,t\rangle}(x) \equiv \exists i \leq \|m\| \exists j \leq \|n\| (x = \langle i-1, j-1\rangle \wedge (0 < i) \wedge (0 < j) \wedge \|t(i,j)\|)$$

we let $|X_{\lambda ij\langle m,n,t\rangle}| = \|m\| \cdot \|n\|$.

Since $\Sigma$ and $\mathtt{P}$ are function symbols in $\mathcal{L}_A^2 \cup \{\Sigma, \mathtt{P}\}$, we translate them as follows:

$$\|\Sigma(T)\| \longmapsto \Sigma(\|T\|)$$
$$\|\mathtt{P}(n,T)\| \longmapsto \mathtt{P}(\|n\|, \|T\|)$$

**Lemma 8.2.3** The axioms of LAP are translated into theorems of $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$.

**Theorem 8.2.2** The theorems of $\forall$LAP can be interpreted in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$.

*Proof.* We showed that $\Pi_0^M$ formulas can be translated into $\Pi_0^B$ formulas, and by Lemma 8.2.3, the axioms of LAP correspond to theorems of $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$.

We extend the translation $\|\cdot\| : \Pi_0^M \longrightarrow \Pi_0^B$ to $\Pi_1^M$ formulas in the obvious way:

$$\|(\forall A \leq n)\alpha\| \longmapsto (\forall X_A \leq (\|n\| \cdot \|n\|))\|\alpha\|$$

So now $\Pi_1^M$ formulas are translated into $\Pi_1^B$ formulas. Since $n$ does not contain the matrix variable $A$, $\|n\| \cdot \|n\|$ does not contain the string variable $X_A$.

This extension still preserves provability; that is, if $\alpha \in \forall$LAP, then $\|\alpha\| \in \tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$. This follows by Lemma 8.2.2, which states that $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ proves $\Pi_1^B$-IND. $\square$

**Corollary 8.2.1** The Cayley-Hamilton Theorem has a feasible proof.

*Proof.* By Theorem 8.2.1, ∀LAP proves the C-H Theorem. By Theorem 8.2.2, all the theorems of ∀LAP have feasible proofs. □

### 8.2.4 Summary of the feasible proof of the C-H Theorem

In this section we are going to summarize the elements of the feasible proof of the Cayley-Hamilton Theorem.

Let $p_A$ be the characteristic polynomial of $A$. The C-H Theorem states that $p_A(A)$ is the zero matrix. We compute the coefficients of $p_A$ using Berkowitz's algorithm which is explained in Section 4.2, and we use induction on the size of matrices to prove the C-H Theorem (Theorem 8.2.1).

The Basis Case is simple, and in the Induction Step we prove:

$$(\forall M \leq n)p_M(M) = 0 \rightarrow (\forall A \leq n+1)p_A(A) = 0$$

This argument can be formalized in ∀LAP. The last step of the proof is to show that ∀LAP can be interpreted in a poly-time theory (we choose $\tilde{\mathbf{V}}^1$); this is done in Section 8.2.2.

Thus, the main feat is proving the Induction Step. This is achieved with Lemma 8.2.1, and with Corollary 6.1.1.

In Lemma 8.2.1 we show that LAP proves that if $p_{C[1|1]}(C[1|1]) = 0$, then the first row and the first column of $p_C(C)$ are zero. Thus, if $M = A[1|1] =$ principal minor of $A$, then $p_M(M) = 0$ implies that the first row and column of $p_A(A)$ are zero. This is a long and technical proof, but it is basic and it can clearly be formalized in LAP.

We need more; we need to show that all of $p_A(A)$ is zero. To this end, we use Corollary 6.1.1, which states that LAP proves, using the C-H Theorem on $n \times n$ matrices, that $p_{I_{ij}AI_{ij}} = p_A$, where $A$ is an $(n+1) \times (n+1)$ matrix. The proof of this Corollary is most of Section 6.1.

So let $A$ be an $(n+1)\times(n+1)$ matrix, and assume that we have $(\forall M \leq n)p_M(M) = 0$. Then, by Corollary 6.1.1, we have that for all $1 \leq i < j \leq n-1$, $p_{(I_{ij}AI_{ij})} = p_A$.

Suppose now that the $i$-th row (column) of $p_A(A)$ is not zero. Then, the first row (column) of $I_{1i}p_A(A)I_{1i}$ is not zero. But:

$$I_{1i}p_A(A)I_{1i} = p_A(I_{1i}AI_{1i}) = p_{(I_{1i}AI_{1i})}(I_{1i}AI_{1i})$$

and the first row and column of $p_{(I_{1i}AI_{1i})}(I_{1i}AI_{1i})$ are zero by Lemma 8.2.1 (by letting $C = I_{1i}AI_{1i}$). Thus, contradiction; it follows that $p_A(A) = 0$.

### 8.2.5 Permutation Frege

In this section we are going to show that Permutation Frege can prove efficiently the C-H Theorem. Since Permutation Frege is a fragment of Substitution Frege (*Substitution Frege* is a Frege proof system with one more rule for replacing all the occurrences of a given variable in a formula by a formula), which in turn can be simulated efficiently by Extended Frege, we have another proof that the C-H Theorem has feasible derivations.

*Permutation Frege* is Frege augmented by the permutation rule. Let $S(\mathbf{x})$ be a propositional sequent whose variables are listed in $\mathbf{x}$. Let $\pi(\mathbf{x})$ denote a permutation of the variables (i.e., if $\mathbf{x} = a_1, a_2, \ldots, a_n$, then $\pi(\mathbf{x}) = a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)}$, where $\pi$ is a permutation in the usual sense). Then, the *permutation rule* is given in Table 8.3.

$$\frac{S(\mathbf{x})}{S(\pi(\mathbf{x}))}$$

Table 8.3: Permutation rule

Suppose that we want to prove the C-H Theorem for $A$, that is, we want to prove $\|p_A(A) = 0\|_\sigma$. We can simulate efficiently the proof of Theorem 8.2.1 with Permutation Frege as follows: in the induction step, we do not actually need the C-H Theorem on all $n \times n$ matrices to conclude that $p_A(A) = 0$, where $A$ is an $(n+1) \times (n+1)$ matrix. We need the C-H Theorem on the principal submatrices of permutations of $A$ (see the proof of Corollary 6.1.1).

We can simulate efficiently the Inductive Step of the proof of Theorem 8.2.1 with Permutation Frege because:

$$\|\alpha(I_{ij}AI_{ij})\|_\sigma = \pi(\|\alpha(A)\|_\sigma)$$

where $\pi = (ij)$. Note that the Permutation Frege proofs are *uniform* as they are obtained from the $\forall$LAP proof.

If we could show that Permutation Frege is weaker that substitution Frege, we could conclude that there are proofs of the Cayley-Hamilton Theorem of lesser complexity than poly-bounded Extended Frege.

### 8.2.6 Quantified Frege

In this section we show that poly-bounded Quantified Frege proves the Cayley-Hamilton Theorem. In fact, we do not even need the full power of Quantified Frege; we can restrict our derivations to be tree-like and use only universally quantified propositional formulas without alternation of quantifiers (i.e., $\Pi_1^q$ formulas, see Definition 8.2.6 below). Since Extended Frege can $p$-simulate such proofs (see [Kra95, Lemma 4.6.3]), we have yet another feasible proof of the C-H Theorem.

Quantified propositional calculus is formed from PK by introducing propositional quantifiers: $\forall x\alpha(x)$ and $\exists x\alpha(x)$, whose meaning is $\alpha(\mathtt{F}) \wedge \alpha(\mathtt{T})$ and $\alpha(\mathtt{F}) \vee \alpha(\mathtt{T})$, respectively. Thus, propositional quantifiers do not increase the expressibility of formulas, but allow them to be shortened.

We follow [Kra95, Definition 4.6.2.] to define Quantified propositional calculus.

**Definition 8.2.5** *Quantified propositional calculus*, denoted $G$, extends the system PK by allowing quantified propositional formulas in sequents and by adopting the quantifier rules in Table 8.4 and 8.5 (where $\beta$ is any formula, and with the restriction that the atom $p$ does not occur in the lower sequents of $\forall$ right and $\exists$ left).

$$\text{left:}\quad \frac{\alpha(\beta), \Gamma \to \Delta}{\forall x\alpha(x), \Gamma \to \Delta} \qquad \text{right:}\quad \frac{\Gamma \to \Delta, \alpha(p)}{\Gamma \to \Delta, \forall x\alpha(x)}$$

Table 8.4: $\forall$-introduction

$$\text{left:}\quad \frac{\alpha(p), \Gamma \to \Delta}{\exists x\alpha(x), \Gamma \to \Delta} \qquad \text{right:}\quad \frac{\Gamma \to \Delta, \alpha(\beta)}{\Gamma \to \Delta, \exists x\alpha(x)}$$

Table 8.5: $\exists$-introduction

**Definition 8.2.6** Let $\Pi_1^q$ be the set of propositional formulas without quantifiers, or whose prenex form is the following: $(\forall a_1 \ldots \forall a_n)\alpha$, where $\alpha$ is quantifier-free.

**Definition 8.2.7** We define $G_1$ to be a sub-system of $G$ where we only allow formulas in $\Pi_1^q$ in the derivations.

**Theorem 8.2.3** The theorems of ∀LAP translate into $\Pi_1^q$ tautologies with uniform, tree-like, poly-bounded $G_1$ proofs.

**Corollary 8.2.2** All the theorems of ∀LAP, and in particular the C-H Theorem, have uniform poly-bounded Extended Frege proofs.

*Proof.* Since Extended Frege can simulate tree-like $G_1$ proofs (see [Kra95, Lemma 4.6.3]), the result follows. □

*Proof.*(of Theorem 8.2.3) We concentrate on the case where the underlying field is $\mathbb{Z}_2$. First we show how to translate $\Pi_1^M$ formulas into families of $\Pi_1^q$ formulas:

$$\|(\forall A \leq n)\alpha\|_\sigma \longmapsto \forall A_{11}\forall A_{12}\ldots\forall A_{\|n\|_\sigma\|n\|_\sigma}\|\alpha\|_\sigma$$

Since $\alpha$ is a formula over $\mathcal{L}_{\text{LAP}}$, we proceed as before. If the underlying field were $\mathbb{Z}_p$, then instead of $\|n\|_\sigma^2$ propositional variables we would have $p\cdot\|n\|_\sigma^2$ propositional variables.

Now we show how to translate a PK-∀LAP derivation into a family of $G_1$ proofs. We only have to show what to do with the two new rules given in Definition 8.2.2. Consider ∀ introduction left, and suppose that:

$$\|\mathtt{r}(T)\leq n\|_\sigma, \|\mathtt{c}(T)\leq n\|_\sigma, \|\mathtt{r}(T)=\mathtt{c}(T)\|_\sigma, \|m\leq n\|_\sigma \quad \text{all map to} \quad \mathtt{T}$$

So we have to show how to derive the sequent $\|(\forall X\leq m)\alpha(X)\|_\sigma, \|\Gamma\|_\sigma \to \|\Delta\|_\sigma$ from the sequent $\|\alpha(T)\|_\sigma, \|\Gamma\|_\sigma \to \|\Delta\|_\sigma$. But it is easy to see that this can be done in $G_1$ with ∀ introduction left applied $\|m\|_\sigma^2$ times. The same can also be done with the second rule. □

## 8.3 Efficient Extended Frege proofs

In this section we give a feasible proof of $AB = I \to BA = I$ using Gaussian Elimination (Section 8.3.3). The fact that $AB = I \to BA = I$ has feasible proofs follows from the feasible proof of the C-H Theorem (see Chapter 6), but here we give a direct, more enlightening, proof.

We also give a feasible proof of the identity $\det(A) = 0 \to AB \neq I$ (Section 8.3.2). This, together with the feasible proof of the C-H Theorem given in the previous section, allows us to give a feasible proof of the multiplicativity of the determinant. To see why

$\det(A) = 0 \to AB \neq I$ and the C-H Theorem are sufficient to prove the multiplicativity of det, see Section 6.4.

We start by proving a correctness condition for Gaussian Elimination, and then we use this correctness condition to prove our two claims. Gaussian Elimination is the standard textbook approach to Linear Algebra. It is a simple poly-time algorithm, but it seems to be inherently sequential. Therefore, we can reason about it in poly-bounded Extended Frege (i.e., in P/poly-Frege), but it seems not possible to reason about it in poly-bounded $\mathrm{NC}^i$-Frege, for any $i$.

*Extended Frege* is a Frege proof system where we allow abbreviating formulas by new variables. See [Urq95] for details.

### 8.3.1 Gaussian Elimination algorithm

Let $e_{ij}$ be a matrix with zeros everywhere except in the $(i,j)$-th position, where it has a 1. A matrix $E$ is an *elementary matrix* if $E$ has one of the following three forms:

$$I + ae_{ij} \quad i \neq j \qquad \text{(type 1)}$$

$$I + e_{ij} + e_{ji} - e_{ii} - e_{jj} \qquad \text{(type 2)}$$

$$I + (c-1)e_{ii} \quad c \neq 0 \qquad \text{(type 3)}$$

Let $A$ be any matrix. If $E$ is an elementary matrix of type 1, then $EA$ is $A$ with the $i$-th row replaced by the sum of the $i$-th row of $A$ and $a$ times the $j$-th row of $A$. If $E$ is an elementary matrix of type 2, then $EA$ is $A$ with the $i$-th and $j$-th rows interchanged. If $E$ is an elementary matrix of type 3, then $EA$ is $A$ with the $i$-th row multiplied by $c \neq 0$.

Define the function `GE` (Gaussian Elimination) as follows:

$$\mathtt{GE}(A) = \{E_1, \ldots, E_k\}$$

where the $E_i$'s are elementary matrices, and the idea is that $E_k \cdots E_1 A$ is in row-echelon form. A matrix is in *row-echelon form* form if it satisfies the following three conditions:

1. If there is a non-zero row, the first non-zero entry of every row is 1. This entry is called a *pivot*.

2. The first non-zero entry of row $i+1$ is to the right of the first non-zero entry of row $i$.

3. The entries above a pivot are zero.

In short, a matrix is in row-echelon form if it looks as follows:

$$\begin{pmatrix} 1 & *\ldots* & 0 & *\ldots* & 0 & *\ldots* & 0 & \\ & & 1 & *\ldots* & 0 & *\ldots* & 0 & \\ & \ddots & & & 1 & *\ldots* & 0 & \\ & & 0 & & & & 1 & \ldots \\ & & & \ddots & & & \vdots & \ddots \end{pmatrix} \tag{8.13}$$

where the $*$'s indicate that any entry can be present.

It remains to explain how to compute $\texttt{GE}(A)$ given a matrix $A$. Each $E_{i+1}$ can be computed from $E_i \cdots E_1 A$.

The algorithm works in two stages. After the first stage the matrix satisfies conditions 1 and 2 of the row-echelon form. We start stage one as follows: find the first column which contains a non-zero entry. If it does not exist, then $A = 0$ and it is already in row-echelon form. Interchange rows using $E_1$ of type 2 moving the first row with the non-zero entry to the top. If the top row already has a non-zero entry then $E_1 = I$. Normalize this entry to 1 using $E_2$ of type 3. Then clear out the other entries in this column by a sequence of $E_i$'s of type 1. The resulting matrix is of the form:

$$\left(\begin{array}{c|c|c} 0 & 1 & B \\ \hline 0 & 0 & D \end{array}\right) = \left(\begin{array}{c|c|c} 0\ldots0 & 1 & *\ldots* \\ \hline 0\ldots0 & 0 & *\ldots* \\ \vdots & \vdots & \vdots \\ 0\ldots0 & 0 & *\ldots* \end{array}\right) \tag{8.14}$$

We now continue, performing the same operations on the smaller matrix $D$, until done, at which point the resulting matrix satisfies conditions 1 and 2. Now, in stage two, we clear the entries above the pivots using elementary matrices of type 1, and the matrix is in row-echelon form, as required.

**Definition 8.3.1** We define the *correctness* of GE as follows: If $\texttt{GE}(A) = \{E_1, \ldots, E_k\}$, then $E_k \ldots E_1 A$ is in row-echelon form.

**Theorem 8.3.1** The correctness of GE can be proven feasibly.

*Proof.* We first prove, by induction on the size of $A$, that $A$ can be put in a form that satisfies conditions 1 and 2 of row-echelon form. The **Basis Case** is easy (a $1 \times 1$ matrix). In the **Induction Step** assume that $D$ (see (8.14)) is in a form that satisfies 1 and 2,

and show that the entire matrix can be put in a form that satisfies 1 and 2. Then clear the entries above the pivots, and we are done. This argument, as presented here is quite informal; it can be formalized, however, in Buss' system $S^1_2$ or in some variant of $\mathbf{V}^1_1$. □

**Corollary 8.3.1** The correctness of GE can be proven with poly-bounded Extended Frege.

*Proof.* Since poly-bounded Extended Frege formalizes poly-time reasoning, we are done. We can also give a direct simulation by Extended Frege of the above proof of correctness; in fact, Extended Frege is ideal for formalizing Gaussian Elimination because of the sequential nature of the definitions of the elementary matrices that bring $A$ to row-echelon form. □

**Corollary 8.3.2** For any *square* matrix $A$, it can be proven with poly-bounded Extended Frege that if $\mathtt{GE}(A) = \{E_1, \ldots, E_k\}$, then $E_k \cdots E_1 A$ is either the identity matrix, or its bottom row is zero.

*Proof.* The proof is an application of the Pigeon-Hole Principle (PHP) to the row-echelon form of $A$, i.e. PHP applied to (8.13). □

### 8.3.2 Extended Frege proof of $\det(A) = 0 \to AB \neq I$

Recall that in Section 6.4 we needed the identity $\det(A) = 0 \to AB \neq I$ in the proof of the multiplicativity of the determinant from the C-H Theorem. If we could give an LAP proof of this identity, it would follow from the results in that section, that the equivalence of the C-H Theorem and the multiplicativity of the determinant can be shown in LAP. However, at this point we only have a feasible proof of this identity (given in this section), and therefore, all that we can state is that the equivalence of the C-H Theorem and the multiplicativity of the determinant can be proven feasibly.

However, we have, by Lemma 6.4.1, that LAP proves the C-H Theorem from the multiplicativity of the determinant.

**Theorem 8.3.2** Let $A, B$ be square matrices. The propositional tautologies expressing $\det(A) = 0 \to AB \neq I$ have poly-bounded Extended Frege proofs.

*Proof.* Let $E$ be an elementary matrix. Then, for any matrix $A$, we have:

$$\det(EA) = \det(E)\det(A)$$

This follows from the axiomatic definition of the determinant, which in turn follows from the C-H Theorem (see Section 6.1), and in this chapter we showed that the C-H Theorem can be proven with uniform poly-bounded Extended Frege proofs.

Using straightforward induction, we can prove that:

$$\det(E_k \cdots E_1 A) = \det(E_k) \cdots \det(E_1)\det(A)$$

Now suppose that $\det(A) = 0$, and let $\mathtt{GE}(A) = \{E_1, \ldots, E_k\}$ be the result of running the Gaussian Elimination algorithm on $A$. It follows, by Corollary 8.3.2, that the bottom row of $E_k \cdots E_1 A$ is zero, since $\det(I) = 1 \neq 0$ (see Corollary 5.2.2). Suppose that for some $B$, $AB = I$. Then $E_k \cdots E_1 AB = E_k \cdots E_1$ has the bottom row zero, which is a contradiction since $E_k \cdots E_1$ is invertible (the inverse of each $E_i$ can be computed easily). Thus, for all $B$, $AB \neq I$. □

Therefore we have proven feasibly that:

$$\det(A) \neq 0 \text{ iff } A \text{ is invertible}$$

the direction "$\Longleftarrow$" is the above result, the direction "$\Longrightarrow$" is the Cayley-Hamilton Theorem.

### 8.3.3 Extended Frege proof of $AB = I \rightarrow BA = I$

Now we show, using Gaussian Elimination, that $AB = I \rightarrow BA = I$ has a poly-bounded Extended Frege proofs. From this it follows that all the matrix identities in:

$$\mathrm{Th}(\mathrm{LA} \cup \{AB = I \rightarrow BA = I\})$$

have poly-bounded Extended Frege proofs.

We already knew all this from the feasible proof of the C-H Theorem (see Section 5.3), but the proof based on Gaussian Elimination is more direct.

**Theorem 8.3.3** The propositional tautologies expressing $AB = I \rightarrow BA = I$ have poly-bounded Extended Frege proofs.

*Proof.* Suppose that $AB = I$. Let $\texttt{GE}(A) = \{E_1, \ldots, E_k\}$. Then, by Corollary 8.3.2, $E_k \cdots E_1 A$ is either the identity matrix, or its bottom row is zero. Since $AB = I$, it follows that:

$$E_k \cdots E_1 AB = E_k \cdots E_1$$

so, if the bottom row of $E_k \cdots E_1 A$ is zero, then so is the bottom row of $E_k \cdots E_1$, which is not possible as $E_k \cdots E_1$ is invertible (where the inverse is $E_1^{-1} \cdots E_k^{-1}$, and the inverse of each elementary matrix is easy to compute). Thus $E_k \cdots E_1 A = I$.

Now that we know that $A$ has a left inverse, and since we can show (in LA) that $AB = I \rightarrow A(BA - I) = I$, it follows that $BA = I$. $\square$

# Chapter 9

# Eight Open Problems

## 9.1 Can LA prove $AB = I \rightarrow BA = I$ ?

We want to separate LA and Th(LA $\cup \{AB = I \rightarrow BA = I\}$). In particular, this would show that $AB = I \rightarrow BA = I$ does not follow from the ring properties of the set of matrices.

This seems to be a much easier problem then separating Frege and Extended Frege. The most obvious approach would be model theoretic: design a model $\mathcal{M}$ of the theory LA such that $\mathcal{M} \not\models AB = I \rightarrow BA = I$.

In this section we present a different approach, due to Alasdair Urquhart [Urq00]. The idea is to show that if $AB = I \rightarrow BA = I$ can be shown in LA, then we could prove the Pigeonhole Principle (PHP) in bounded-depth Frege with mod 2 gates, which is believed to be impossible. Here we present the idea in ([Urq00]).

When translating general matrix identities (over the language $\mathcal{L}_{\text{LA}}$ into bounded-depth Frege (without mod 2 gates), we are faced with the difficulty that matrix multiplication cannot be translated efficiently into such a system, since unbounded parity gates are not available as a primitive connective (i.e., we do not have $\texttt{MOD}_{2,1}$ gates). However, matrix products can be expressed if we restrict ourselves to a special class of matrices.

Let us say that a $\{0,1\}$ matrix is a *partial permutation matrix* if each row and column contains at most one 1 (the terminology is taken from the monograph of Kim [Kim82] on Boolean matrix theory). If $A$ and $B$ are square partial permutation matrices, then the $(i,j)$-th entry of their product can be given as follows:

$$\bigvee_{k=1}^{n} (A_{ik} \wedge B_{kj})$$

In other words, in the special case of partial permutation matrices, the normal matrix product coincides with the Boolean matrix product (see [Kim82] for all the details).

It follows that in the special case of permutation matrices, $AB = I \rightarrow BA = I$ can be expressed efficiently in bounded-depth Frege. However:

**Lemma 9.1.1** The matrix identity $AB = I \rightarrow BA = I$ is hard for bounded-depth Frege, even when $A, B$ are partial permutation matrices.

*Proof.* Suppose that bounded-depth Frege proves $AB = I \rightarrow BA = I$ in the case where $A, B$ are partial permutation matrices. We are going to use this to show that bounded-depth Frege can give efficient proofs of the PHP, and hence derive our contradiction[1].

We shall take the PHP in the following form: If $f$ is an injective mapping on a finite set, then $f$ is surjective.

A square $\{0,1\}$ matrix can be considered as the incidence matrix of a relation on a finite set. Furthermore, the boolean product of matrices corresponds exactly to the relative product of two relations. That is to say, if $A, B$ are the incidence matrices corresponding to relations $R, S$, respectively, then their product $AB$ is the incidence matrix of their relative product $R|S$.

Let $R$ be a relation on a finite set that corresponds to a bijection. That is, $R$ satisfies the conditions:

1. $(Rxy \wedge Rxz) \supset y = z$
2. $\forall x \exists y Rxy$
3. $(Rxz \wedge Ryz) \supset x = y$

Hence, the incidence matrix $A$ corresponding to $R$ is a partial permutation matrix. Let $B$ be the incidence matrix of the converse of $R$; this is simply the transpose of $A$, $A^t$. Clearly, $B$ is also a partial permutation matrix.

By assumption we have that $R|S = Id$, hence $AB = I$. By our matrix implication we have that $BA = I$, so $S|R = Id$. This means that the domain of the converse of $R$, $S$, is the whole set, and hence $R$ is surjective.

All this can be easily formalized in bounded-depth Frege. However, the PHP requires exponential size bounded-depth Frege derivations. Therefore, $AB = I \rightarrow BA = I$,

[1] See [Pit92] where it is proven (in Chapter 3) that the proofs of the PHP in bounded-depth Frege require exponential size.

restricted to partial permutation matrices, also requires exponential size derivations in bounded-depth Frege. □

**Theorem 9.1.1** If PHP requires super polynomial proofs in bounded-depth Frege with mod 2 gates, then $\mathrm{LA} \nvdash AB = I \rightarrow BA = I$.

*Proof.* If PHP requires super polynomial proofs in bounded-depth Frege with mod 2 gates, then so does $AB = I \rightarrow BA = I$.

If $AB = I \rightarrow BA = I$ is not provable efficiently in bounded depth Frege with $\mathtt{MOD}_{2,1}$ gates, then, by Theorems 7.2.1 and 7.2.2, $AB = I \rightarrow BA = I$ is not provable in LA. □

## 9.2 Is $AB = I \rightarrow BA = I$ complete ?

To pose this question, we must propose some plausible definition of completeness. As we have seen, $AB = I \rightarrow BA = I$, is representative in some sense of a large class of universal matrix identities: from Theorem 3.2.1, we know that an efficient $C$-Frege proof of $AB = I \rightarrow BA = I$, where $C$ is a complexity class such that $\mathrm{NC}^1 \subseteq C$, would imply that many matrix identities have efficient $C$-Frege proofs.

The C-H Theorem states that for all $A$, $p_A(A) = 0$, that is, that the characteristic poly of $A$ is an annihilating poly of $A$. We know from Theorem 5.3.1 that the C-H Theorem implies, in $\mathrm{LA}\mathtt{P}$, hard matrix identities. Thus, $AB = I \rightarrow BA = I$ can be proven in:

$$\mathrm{Th}(\mathrm{LA}\mathtt{P} \cup \{p_A(A) = 0\})$$

where $p_A(A) = 0$ can be taken to be the annihilating poly of $A$, not necessarily the char poly. (In Section 5.3 we show that the C-H Theorem implies $AB = I \rightarrow BA = I$, but all we use about the C-H Theorem is that the char poly is an annihilating poly).

We can pose the question of the completeness of $AB = I \rightarrow BA = I$ as follows:

$$\mathrm{Th}(\mathrm{LA}\mathtt{P} \cup \{p_A(A) = 0\})|_{\mathcal{L}_{\mathrm{LA}}} \stackrel{?}{\subseteq} \mathrm{Th}(\mathrm{LA} \cup \{AB = I \rightarrow BA = I\})$$

That is, can we derive from $AB = I \rightarrow BA = I$ the same identities that we can derive from the existence of an annihilating poly?

## 9.3 Does $AB = I \rightarrow BA = I$ have NC$^2$-Frege proofs ?

The answer would be "Yes", if we could show that LAP proves $AB = I \rightarrow BA = I$.

We conjecture that $AB = I \rightarrow BA = I$ can be proven in LAP. To show that LAP proves $AB = I \rightarrow BA = I$, we only need to show that LAP proves that every matrix $A$ has a non-zero annihilating polynomial.

To see this, recall that in Section 5.3, we prove Theorem 5.3.1 which states that LAP shows that hard matrix identities follow from the Cayley-Hamilton Theorem. The only property of the characteristic poly that we use in the proof of this theorem, is that it is a non-zero annihilating polynomial.

We conjecture that LAP is strong enough to show that every matrix has a non-zero annihilating polynomial. What evidence do we have for this? The set of matrices $\{I, A, A^2, \ldots, A^{n^2}\}$ is a linearly *dependent* set, with a "high degree of redundancy"; by this we mean that already the set $\{I, A, \ldots, A^n\}$ is linearly dependent (by the C-H Theorem!). Therefore, it seems very plausible that we can compute, using matrix powering, a set of coefficients $c_0, c_1, c_2, \ldots, c_{n^2}$ (where at least one is non-zero), and show in LAP that $c_0 I + c_1 A + c_2 A^2 + \cdots + c_{n^2} A^{n^2} = 0$

If LAP proves $AB = I \rightarrow BA = I$, then it would follows that hard matrix identities have quasi-poly-bounded Frege proofs, i.e., NC$^2$-Frege proofs.

## 9.4 Can LAP prove $\det(A) = 0 \rightarrow AB \neq I$ ?

From Section 6.4 we know that if LAP can prove $\det(A) = 0 \rightarrow AB \neq I$, then it would follow that LAP proves the equivalence of the Cayley-Hamilton Theorem and the multiplicativity of the determinant (see Chapter 6 for more details).

In a sense this identity is the converse of the Cayley-Hamilton Theorem: we can look at the C-H Theorem as stating that if $\det(A) \neq 0$, then $A$ has an inverse. Therefore, showing that LAP can prove $\det(A) = 0 \rightarrow AB \neq I$ would show that LAP can prove the equivalence of the C-H Theorem and its converse—an interesting result.

## 9.5 Can LAP prove the C-H Theorem ?

It is difficult to conjecture the answer to this question. A "Yes" answer would imply that the C-H Theorem has quasi-poly-bounded Frege proofs (more precisely, NC$^2$-Frege

proofs); a major result. A "No" answer would prove the separation of Frege and Extended Frege; a very big result.

Still, it is important to note that matrix powering is the only major operation that we have to perform in the computation of the coefficients of the char poly. This is an $NC^2$ computation, therefore, it does not seem too far-fetched to assume that the correctness of this computation can be proven in $NC^2$-Frege. See Section 9.7 for some (slight) evidence towards an $NC^2$-Frege proof of the C-H Theorem. However, a uniform poly-bounded $NC^2$-Frege proof of the C-H Theorem, does not imply that LAP can prove the C-H Theorem.

Finally, an $NC^2$-Frege of the C-H Theorem, via the correctness of Berkowitz's algorithm, would be consistent with the meta-theorem which states that "the correctness of an algorithm should be provable within the complexity of the algorithm".

## 9.6 Feasible proofs based on Gaussian Elimination ?

In Section 8.3 we showed that we can give poly-bounded Extended Frege proofs of the identity $AB = I \rightarrow BA = I$, and of $\det(A) = 0 \rightarrow AB \neq I$, based on the Gaussian Elimination algorithm.

Is it possible to give feasible proofs based on Gaussian Elimination, of the other fundamental properties? That is, can we give a feasible proof of the axiomatic definition of the determinant based on Gaussian Elimination?

Such a proof would involve defining the determinant in terms of Gaussian Elimination (rather than in terms of Berkowitz's algorithms, as in this thesis). This definition of the determinant would be something like the product of the elements on the diagonal, when the matrix is reduced to row-echelon form. We would have to fix the procedure, as otherwise, to prove uniqueness of the result, we would have to have the axiomatic definition of the determinant.

## 9.7 How strong is Permutation Frege ?

It is easy to see that Permutation Frege is a restricted instance of Substitution Frege; what about the converse? That is, can Permutation Frege $p$-simulate Substitution Frege?

A "No" answer would show that the C-H Theorem can be proven in propositional proof systems strictly weaker than Extended Frege. The fact that the C-H Theorem can

be proven in uniform poly-bounded Permutation Frege (see Section 8.2.5) is the best indication we have of the possibility that the C-H Theorem can be proven in systems strictly weaker than Extended Frege. Perhaps the C-H Theorem can be proven in uniform $NC^2$-Frege after all? We conjecture $NC^2$-Frege since matrix powering, the main computation in the proof of the C-H Theorem, is an $NC^2$ operation.

In [Urq98] Urquhart discusses the permutation rule in the context of resolution systems (he calls the permutation rule, the symmetry rule). Urquhart proves exponential lower bounds on the size of resolution refutations using two forms of the symmetry rule, and discusses the relationship of symmetry rules to the extension rule (which allows the use of definitions in proofs to abbreviate formulas).

In [Ara95] and [Ara] Arai proves properties of cut-free Gentzen systems augmented by the permutation rule. Also, in [Ara], Arai shows that Frege $p$-simulates cut-free Gentzen with the renaming rule iff Frege $p$-simulates Extended Frege. The renaming rule permits arbitrary renaming of variables (so, unlike in the permutation rule, two distinct variables can be mapped to the same variable). We call this proof system *Renaming Frege*.

## 9.8 Does $\forall$LAP capture polytime reasoning ?

In Section 8.2.3 we showed that $\forall$LAP can be interpreted in $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$, and from this it follows that all the theorems of $\forall$LAP have feasible proofs. The natural question to ask is the following: *can all the feasible theorems be proven in* $\forall$LAP ? That is, can $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ be interpreted in $\forall$LAP? We believe that the answer to this question is yes; however, the actual translation has many technical difficulties that we do not see how to overcome at the moment.

Here we present part of the translation, and we point out the major difficulties. To see that $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ can be interpreted in $\forall$LAP, we translate formulas over the language of $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ into formulas over the language of $\forall\mathrm{LAP}'$, where $\forall\mathrm{LAP}'$ is the same as $\forall$LAP, but we allow index quantifiers. To prove that $\forall\mathrm{LAP}'$ is a conservative extension of $\forall$LAP, we need to prove the completeness of the sequent calculus LK-$\forall$LAP (see Definition 8.2.2). This proof of completeness should follow the proof theoretic argument, with *anchored* proofs, but the complication is the $\lambda$-terms.

The translation from $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ into $\forall$LAP is straightforward: index terms go to to index terms directly ($\forall$LAP has index addition, multiplication, etc.). String variables go to matrix variables, in such a way that a string variable $X$ of length $|X|$, corresponds

to a row matrix $X$ of length $\mathtt{c}(X)$. In the translations, F is going to be represented by $0_{\text{field}}$, and T is going to be represented by a sum of $1_{\text{field}}$'s. We translate $X(i)$ into $\mathtt{e}(X, 1, i+1) \neq 0_{\text{field}}$. If $A$ is a formula over $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$, we denote the translated formula over $\forall\text{LA}\mathtt{P}'$ by $A^*$.

In the Table 8.2 (page 127), the axioms B1–B10 of $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$ are presented. These axioms either have counterparts in LA, or can be easily proven in LA. Axiom L, $X(i) \supset i < |X|$, translates into $\mathtt{e}(X, 1, i+1) \neq 0_{\text{field}} \supset i < \mathtt{c}(X)$. This can be easily proven in LA since out-of-bounds entries are zero by A28. The axiom IND can be simulated with the induction rule:

$$\frac{\mathtt{e}(X, 1, 0+1) \neq 0_{\text{field}} \quad \mathtt{e}(X, 1, i+1) \neq 0_{\text{field}} \supset \mathtt{e}(X, 1, i+1+1) \neq 0_{\text{field}}}{\mathtt{e}(X, 1, j) \neq 0_{\text{field}}}$$

The axiom $\Sigma_1^B$-IND, $(A(0) \wedge (A(x) \supset A(x+1)) \supset A(y)$, can be also presented as $\Pi_1^B$-IND (as $\Sigma_1^B$-IND and $\Pi_1^B$-IND are equivalent), and $\forall\text{LA}\mathtt{P}$ has $\Pi_1^M$-IND, which allows induction over formulas of the form $\forall X \leq nA$, where $A$ has no matrix quantifiers. The only thing is that $\Pi_1^M$-IND allows a single matrix quantifier, but this can be fixed in one of the following two ways: combine several matrices into one (each matrix corresponds to a block in a single big matrix), or extend $\Pi_1^M$-IND to allow a block of bounded universal matrix quantifiers.

Finally, the axiom $\Sigma_0^B$-COMP, $\exists X \leq i \forall j < i(|X| = i \wedge (X(j) \equiv A(j)))$, is the only one that gives some difficulties. What we want to do is construct a matrix $X$, of size $1 \times i$, such that $\mathtt{e}(X, 1, j+1) \neq 0_{\text{field}}$ iff $A^*(j)$ is true. Since we have a restriction on the $\alpha$'s that appear in $\text{cond}(t, u, \alpha)$ (they must be atomic formulas of type index; item 9 in the inductive definition of terms and formulas), we cannot simply define $X$ as $\lambda kl\langle 1, i, \text{cond}(1_{\text{field}}, 0_{\text{field}}, A^*)\rangle$.

We resort to the following trick. We translate a formula $A$ over $\tilde{\mathbf{V}}^1(\Sigma, \mathtt{P})$, into a term of type field, $t_A$, over LA, so that $A^*$ is true iff $t_A \neq 0$ is true.

Here is the translation: terms are translated as explained above, but we must show what to do with logical connectives, with the predicate symbols $=$ and $\leq$, and with bounded index quantifiers. We define $t_{\neg A}$ as $1_{\text{field}} - (t_A * t_A^{-1})$ (note that $0^{-1} = 0$), $t_{A_1 \wedge A_2}$ as $t_{A_1} *_{\text{field}} t_{A_2}$, and $t_{A_1 \vee A_2}$ as $t_{A_1} +_{\text{field}} t_{A_2}$. (Since below we add axioms that ensure that the field is of characteristic zero, no sum of $1_{\text{field}}$'s will result in $0_{\text{field}}$).

We translate $m = n$ into $\text{cond}(1_{\text{field}}, 0_{\text{field}}, m = n)$, and similarly for $\leq$. Finally, $t_{\exists i \leq nA}$ is $\Sigma\lambda kl\langle 1, n, t_A\rangle$, and $t_{\forall i \leq nA}$ is $1_{\text{field}} - (\Sigma\lambda kl\langle 1, n, t_{\neg A}\rangle)(\Sigma\lambda kl\langle 1, n, t_{\neg A}\rangle)^{-1}$.

Thus, given $\exists X \leq i \forall j < i(|X| = i \wedge (X(j) \equiv A(j)))$, we can define $X$ as follows: $\lambda kl\langle 1, i, t_A\rangle$. To ensure that a sum of $1_{\text{field}}$'s is never one (we do not want fields of finite characteristic), we add the following axiom schema to $\forall$LAP: $A^* \equiv (t_A \neq 0)$. This axiom schema not only assures that the underlying field does not have a finite characteristic, but it also asserts the correctness of the construction of the matrix variable $X$. However, we have to show that these axioms are feasible; for example, we have to show that their translations have uniform polybounded Extended Frege proofs.

It is really a characteristic zero version of $\forall$LAP that seems to capture polytime reasoning, rather than the original $\forall$LAP; can we do it without the characteristic zero axioms?

# Bibliography


[AB95] J.L. Alperin and Rowen B. Bell. *Groups and Representations*. Springer, 1995.

[Ara] Noriko H. Arai. Tractability of cut-free Gentzen type propositional calculus with permutation inference II.

[Ara95] Noriko H. Arai. Tractability of cut-free Gentzen type propositional calculus with permutation inference. 1995.

[Art91] Michael Artin. *Algebra*. Prentice-Hall, 1991.

[BBP94] M. Bonet, S. Buss, and T. Pitassi. Are there hard examples for frege systems? *Feasible Mathematics*, II:30–56, 1994.

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3):147–150, 1984.

[BF92] László Babai and Péter Frankl. Linear algebra methods in combinatorics. This book is unpublished, but it is possible to obtain a copy of the monograph (216 pages) by writing to the dept. of Computer Science at the University of Chicago, 1100 E. 58th Street, Chicago IL 60637-1504, USA, September 1992.

[BIK+92] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods. Exponential lower bounds for the pigeonhole principle. *Proc. of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, pages 200–220, 1992.

[BIP93] Paul Beame, Russell Impagliazzo, and Toniann Pitassi. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3(2):97–140, 1993.

[BM77] John Bell and Moshé Machover. *A Course in Mathematical Logic*. North Holland, 1977.

[BP96] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. *FOCS*, pages 274–282, 1996.

[BP98] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. *Bulletin of the EATACS, TR98-067*, 1998.

[BS90] Ravi B. Boppana and Michael Sipser. The complexity of finite functions. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 758–804. Elsevier Science Publishers B. V., 1990.

[Bus86] Samuel R. Buss. *Bounded Arithmetic*. Studies in proof theory. Napoli, 1986.

[Bus98] Samuel R. Buss. An introduction to proof theory. In Samuel R. Buss, editor, *Handbook of Proof Theory*, pages 1–78. North Holland, 1998.

[CK01] Stephen A. Cook and Antonina Kolokolova. A second-order system for polytime reasoning based on Grädel's theorem. 2001.

[CLR97] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1997.

[Coo75] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus. *Proc. 7th ACM Symposium on the Theory of Computation*, pages 83–97, 1975.

[Coo85] Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Computation*, 64(13):2–22, 1985.

[Coo98] Stephen A. Cook. Proof complexity and bounded arithmetic. Notes for CSC2429S, "Proof Complexity and Bounded Arithmetic", given at the Fields Institute (available on line at `www.cs.toronto.edu/~sacook`), 1998.

[Coo00a] Stephen A. Cook. Course notes. Notes for CSC438, 2000.

[Coo00b] Stephen A. Cook. The P versus NP problem. *Clay Mathematical Institute; The Millennium Prize Problem*, 2000.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *JSL*, 44:36–50, 1979.

[CS99] Stephen A. Cook and Michael Soltys. Boolean programs and quantified propositional proof systems. *Bulletin of the Section of Logic*, 28(3):119–129, 1999.

[CU93] Stephen A. Cook and Alasdair Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63(2):103–200, 1993.

[DF91] David S. Dummit and Richard M. Foote. *Abstract Algebra*. Prentice-Hall, 1991.

[Dow79] Martin Dowd. *Propositional Representations of Arithmetic Proofs*. PhD thesis, University of Toronto, 1979.

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1), 1984.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[Hal95] Paul R. Halmos. *Linear Algebra Problem Book*. The Mathematical Association of America, 1995.

[HS86] J. Roger Hindley and Jonathan P. Seldin. *Introduction to Combinators and $\lambda$-Calculus*. Cambridge University Press, 1986.

[Kab01] Valentine Kabanets. *Nonuniformly Hard Boolean Functions and Uniform Complexity Classes*. PhD thesis, University of Toronto, 2001.

[Kim82] Ki Hang Kim. *Boolean Matrix Theory and Applications*. Marcel Dekker, 1982.

[Kra95] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Cambridge, 1995.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MV97] Meena Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chicago Journal of Theoretical Computer Science*, 5, 1997.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pit92] Toniann Pitassi. *The Power of Weak Formal Systems.* PhD thesis, University of Toronto, 1992.

[Pit00] François Pitt. *A Quantifier-Free String Theory for ALOGTIME Reasoning.* PhD thesis, University of Toronto, 2000.

[Rut62] D. E. Rutheford. Inverses of boolean matrices. 1962.

[Sip97] Michael Sipser. *Introduction to the Theory of Computation.* PWS Publishing Company, 1997.

[SP95] Uwe Schöning and Randall Pruim. *Gems of Theoretical Computer Science.* Springer, 1995.

[Str83] Howard Straubing. A combinatorial proof of the Cayley-Hamilton Theorem. *Discrete Mathematics*, 43:273–279, 1983.

[Urq95] Alasdair Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1(4):425–467, 1995.

[Urq98] Alasdair Urquhart. The symmetry rule in propositional logic. *Discrete Applied Mathematics*, 96–97:177–193, 1998.

[Urq00] Alasdair Urquhart. Matrix identities and the pigeonhole principle. Private communication, October 2000.

[Val92] L.G. Valiant. Why is boolean complexity theory difficult? In M. Paterson, editor, *Boolean Function Complexity*, volume 169 of *London Mathematical Society Lecture Notes Series*, pages 84–94. Cambridge University Press, 1992.

[vzG91] Joachim von zur Gathen. Boolean circuits versus arithmetic circuits. *Information and Computation*, 91:142–154, 1991.

[vzG93] Joachim von zur Gathen. Parallel linear algebra. In John H. Reif, editor, *Synthesis of Parallel Algorithms*, pages 574–617. Morgan and Kaufman, 1993.

[vzGG99] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions.* Wiley-Teubner Series in Computer Science, 1987.

[Zam96] Domenico Zambella. Notes on polynomially bounded arithmetic. *JSL*, 61(3):942–966, 1996.

[Zha99] Fuzhen Zhang. *Matrix Theory*. Springer, 1999.

# Index