Graphical Algebraic Geometry: From Ideals and Varieties to Quantum Calculi

Dichuan Gao
University of Oxford, UK

Razin A. Shaikh
University of Oxford, UK

Aleks Kissinger
University of Oxford, UK

Abstract

We introduce *Graphical Algebraic Geometry* (GAG), a family of diagrammatic languages extending the Graphical Linear Algebra programme. We construct several languages within this family and prove that they are universal and complete for the corresponding (co)span semantics of commutative algebras and affine varieties. This framework provides clear graphical representations of algebraic structures – such as polynomials, ideals, and varieties – enabling intuitive yet rigorous diagrammatic reasoning.

We showcase two practical viewpoints on GAG. First, we show that instances of counting constraint satisfaction problem (#CSP) are recast as rewrite problems of closed diagrams in GAG. This means that deciding rewritability in GAG is #P-hard, and GAG can be viewed as a complete and compositional rewrite system for networks of polynomial constraints. Second, we characterize the qudit ZH calculus, a diagrammatic language for quantum computation, as an extension of Graphical Algebraic Geometry. This establishes the correspondence that *Graphical Algebraic Geometry is to the ZH calculus what Graphical Linear Algebra is to the ZX calculus*. Using this construction, we show that computing amplitudes in qudit ZH requires only a constant number of queries to a GAG oracle.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting; Theory of computation → Categorical semantics; Theory of computation → Quantum computation theory

Keywords and phrases string diagrams, algebraic geometry, ZH calculus, constraint satisfaction

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.48

Related Version *Full Version*: `http://arxiv.org/abs/2605.13993`

Funding This work is supported by the Engineering and Physical Sciences Research Council grant number EP/Z002230/1, "(De)constructing quantum software (DeQS)".
Dichuan Gao: Supported by the Wolfson Harrison UK Research Council Quantum Foundation Scholarship.
Razin A. Shaikh: Supported by the Clarendon Fund Scholarship.

Acknowledgements We would like to thank Soichiro Fujii for pointing out initially the possibility of building the graphical language separately for the algebraic setting and for the geometric setting. We would like to thank Lia Yeh, Cole Comfort, Robert Booth, Nathan Corbyn, and Owen Lynch for many helpful discussions.

1 Introduction

Algebraic geometry and commutative algebra have been important influences in several fields in computer science, for example in robotics [1], computer vision [38], and in cryptography [28]. In the converse direction, computational techniques have become influential in algebraic geometry with the advent of computer algebra systems [23].

Of particular interest to us is the use of commutative algebra and geometry in two related areas of research. The first, and perhaps more obvious one, is that of Constraint Satisfaction Problems (CSP) [67]. A CSP instance where constraints are given as polynomials is a problem about an algebraic variety. In particular, in the case where the domain of the CSP is a finite field $\mathbb{F}_q$, the corresponding #CSP instance [43] is to count $\mathbb{F}_q$-rational points in a variety defined by polynomial constraints – a classical and central problem in 20th century algebraic geometry [40, 70]. More recently, cryptographic proof systems such as SNARK and STARK have relied heavily on algebraic techniques over finite fields, demonstrating the continuing relevance of this approach [7, 35].

The second area is quantum information theory. It has long been known that algebraic geometry can be used to design classical codes with good asymptotic behavior [63, 8, 34, 68, 42, 37, 69], and that via the CSS or stabilizer constructions these classical AG codes lift to quantum error correction (QEC) codes with equally good properties [2, 59, 58, 29, 45, 44, 65, 66]. Such codes also played a central role in the groundbreaking construction of a magic-state distillation protocol with constant overhead [71]. Aside from error correction, algebraic geometry has also served as a useful language for foundational areas of research such as quantum entanglement and quantum contextuality [19, 60, 55, 31, 27].

To systematically understand the role algebraic geometry can play in quantum information theory, we propose that graphical calculi would be useful. Graphical calculi are compositional diagrammatic languages based in category theory, equipped with formal "rewrite rules" with which one can manipulate diagrams rigorously as equational terms. For an in-depth exposition of graphical calculi and rewrite theory, see [9, 10, 11]. Such calculi are usually designed to represent some family of physical phenomena or computational processes. For instance, the ZX calculus and its related calculi have become prominent as languages for representing quantum computational processes [17, 18, 24, 50, 49, 61].

When considering the classical algebraic structure inherent in quantum computing, we often encounter fragments of the quantum calculi that are best characterized by languages in the family called Graphical Linear Algebra (GLA) [13, 15, 73]. Broadly speaking, GLA consists of diagrammatic languages that have denotational semantics built from classical algebraic structure. On the semantics side, the GLA story begins with a category of classical maps: for example, the linear maps between $\mathbb{F}_2$ spaces. It then applies the (co)span or the (co)relation construction to that category to obtain a dagger-category. For example, $\mathbb{F}_2$ linear maps give rise to spans of $\mathbb{F}_2$ linear maps, or to $\mathbb{F}_2$ linear subsets. On the syntactic side, it builds a simple diagrammatic language with generators that have clear semantic interpretations, and rewrite rules that characterize denotational equivalence.

Languages in the graphical linear algebra family are excellent at picking out fragments of quantum computation that correspond to some given notion of a "classical-relational backbone". For example, the CSS codes resulting from classical $\mathbb{F}_2$-linear codes correspond to the phase-free fragment of the ZX calculus [48], which is picked out by the language of interacting bialgebras [13], the language that kickstarted the GLA literature. Similarly, the language of interacting Hopf algebras [15, 73] picks out the phase-free fragment of the qudit ZX calculus, which models the linear-relational core of qudit computation. Expanding our algebraic structure from linear to affine algebra, the language of graphical affine algebra [12] picks out the phase-free ZX with a Pauli X gate, which is the affine-relational core of qubit computation [20].

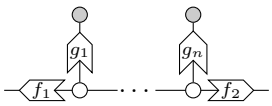
Of course, GLA has applications far beyond quantum computing. For instance, the language for linear relations over a field of rational functions $\mathbb{R}(x)$ models signal flow graphs and linear dynamical systems [14, 6, 26], while the language for affine relations models the fundamental structure of concurrency [12]. The strength of GLA lies in its ability to model a remarkably varied set of physical phenomena using the same formal structure.

However, existing versions of GLA are not yet sufficient to model the fragment of quantum computation with classical algebraic geometry as its "backbone"; nor can they reason about generic constraint satisfaction problems built on polynomial constraints. This is because commutative algebra involves a multiplication operator (an AND gate in the $\mathbb{F}_2$ case), which is essentially nonlinear (and non-affine for that matter). A graphical calculus that models classical algebraic geometry would need to supply such an operator, or equivalently, to supply the Toffoli gate.

In the realm of quantum computing, the ZH calculus [3, 4] or equivalently the ZX& calculus [20, 21] supplies such a gate for the $\mathbb{F}_2$ case, and the Galois-qudit ZH calculus [64, 30] serves as the appropriate version in the nonbinary case. The motivating question of this paper is therefore: can we extend GLA in a principled and simple way, to pick out the nonlinear "classical backbone" of the ZH calculus, corresponding to classical algebraic geometry? If so, can we provide a sound and complete set of rewrite rules for that extension?

The answer is yes. However, unlike in the linear case, in which algebra and geometry coincide, we must deal with a gap between commutative algebra and algebraic geometry, which is bridged by a family of theorems known as the Nullstellensatze [41, 32, 39]. Therefore we construct our extension of GLA in two steps.

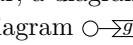
First, we develop a language GCA_k, which we call graphical commutative algebra (GCA). Diagrams in GCA_k are built from gates corresponding to copying, addition, scaling, multiplication, and the units of these operators – the standard generators featured in the theory of commutative algebras. Diagrams can be reduced to a pseudonormal form that looks like

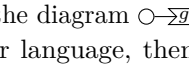


(1)

Denotational semantics are given as structured cospans of commutative algebras. In the pseudonormal form (1), the vertical wires denote a commutative algebra quotiented by the ideal $(g_1, \ldots, g_n)$, while f_1 and f_2 denote two algebra morphisms pointing into it. This captures the nonlinear *algebraic* backbone of quantum processes in ZH. Our first main result is a sound and complete rule set for GCA_k with respect to these cospan semantics.

Second, we develop languages GAG_k and GAG_q, which we refer to as Graphical Algebraic Geometry (GAG) over algebraically closed fields and finite fields, respectively. Diagrams in these are built from the same generators as diagrams in GCA_k, but they are subject to further rewrite rules that express the content of the Nullstellensatze. Denotational semantics are given respectively, as structured spans of affine varieties, and of $\mathbb{F}_q$-loci of varieties (or, equivalently, of finite sets). We still have the same pseudonormal form (1), but now the vertical wires with polynomials $g_1, \ldots, g_n$ denote an affine variety carved out by polynomial constraints, and f_1, f_2 are polynomial maps pointing out of it. This captures the nonlinear *geometric* backbone of quantum processes in ZH. Our second main result is that these languages are sound and complete with respect to their span semantics.

Our languages bear a clear relation to CSP instances with polynomial constraints. As will become clear, a diagram of the form 

just *is* the CSP instance with constraints $g = 0$, while the diagram 

is the #CSP problem of counting the number of points satisfying g. Our language, then, gives a compositional and complete rewrite system for networks of CSP instances with polynomial constraints. Through this view of GAG, we show that determining rewritability of an arbitrary pair of GAG diagrams is #P-hard. In the special case where the field is $\mathbb{F}_2$, this just reduces to a compositional language for SAT instances. This serves as an algebraic elucidation of two ideas already present in the literature: first,

that the ZX& calculus completely models "qubit multirelations" [21]; and second, that the ZH calculus can be used to model #SAT instances, and to reason about optimizations of #SAT algorithms [51, 52].

On the other hand, we justify our claim that *GAG is to ZH what GLA is to ZX* by characterizing the Galois-qudit ZH as an extension of GAG over finite fields. Indeed, we show that once we have access to the "classical backbone" carved out by GAG, any process in ZH can be built by consuming just a single basis state in the Fourier basis, up to a finite number of global scalars.

Outline of the paper

We begin in Section 2 by covering the requisite theoretical background on category theory and algebraic geometry. In Section 3, we discuss the Lawvere theory of commutative algebras, which serve as a building block (the "forward fragment") of our languages. We then construct the (co)span semantics categories of interest, namely, the structured cospan category of commutative algebras, and the structured span category of affine varieties, in Section 4. We prove our first main result in Section 5, where we construct the language GCA_k and prove its soundness and completeness for cospans of commutative algebras. The second main result is proven in Section 6, where we construct the languages GAG_k and GAG_q, and prove their soundness and completeness for spans of affine varieties and of rational loci, respectively. In Section 7, we show that closed diagrams in GAG are equivalent to instances of #CSP, and therefore rewriting in GAG is #P-hard. Finally, in Section 8 we show how the Galois-qudit ZH can be built as an extension of GAG.

Related Work

There have been several previous works which present diagrammatic languages of a similar flavor, although never from an explicitly algebraic-geometric angle. Lafont has presented an graphical calculus [54] of Boolean circuits which is equivalent to our "polynomial fragment" (Section 3), but modulo the additional rule that encodes the identity $x^2 = x$ for boolean variables. As will become clear in Section 6.2, this makes it a "forward fragment" of the language $\mathbb{L}_{\mathrm{Span}}$ constructed in section 6.2, for the finite field $\mathbb{F}_2$. Wilson and Zanasi have presented a diagrammatic language which formalizes differentiation of polynomials [72]. Underlying that language is a notion of polynomial circuits, which is again identical to our polynomial fragment in Section 3, defined over arbitrary semirings. Finally, the ZX&-calculus of Comfort [21] is the same language as the language GAG_q constructed in Section 6.2 for the field $\mathbb{F}_2$. However, the completeness proofs for the ZX&-calculus are tailored for that specific field, so in comparison, our approach serves to elucidate the algebraic nature of the language even in the case of $\mathbb{F}_2$.

2 Background

2.1 Diagrammatic Languages

In this section we present the theory of PROPs as it relates to the construction of our diagrammatic languages. For a more in-depth and theoretical discussion of PROPs, we recommend the excellent note by Lack [53].

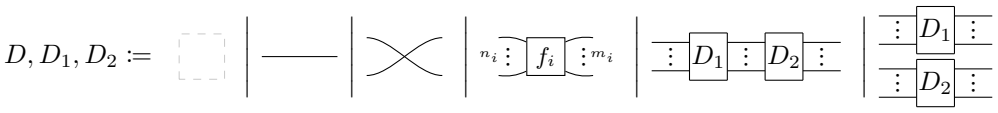


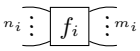
Figure 1 Recursive Definition of Diagrammatic Terms.

▶ **Definition 1** ([57]). *A Product-and-Permutation-Category (PROP) is a strict symmetric monoidal category $\mathcal{C}$ with $Ob(\mathcal{C}) \cong \mathbb{N}$, whose monoidal product on objects corresponds to addition of natural numbers.*

A PROP morphism from $\mathcal{C}$ to $\mathcal{D}$ is a strict symmetric monoidal functor such that $1_{\mathcal{C}} \mapsto 1_{\mathcal{D}}$, or in other words, the objects are fixed. A PROP isomorphism is a PROP morphism that is fully faithful.

When we speak of a **presentation** of a PROP, we are speaking about a particular data structure, which we describe here with terminology adopted from the graphical linear algebra literature [15].

▶ **Definition 2.** *A **signature** is a set $S = \{f_i : n_i \to m_i\}_{i \in \mathcal{I}}$ where each element has a name, a number of inputs, and a number of outputs, and is pictured as a node in a string diagram:*



*A **diagrammatic term over a signature** S is an inductive structure constructed as in Figure 1.*

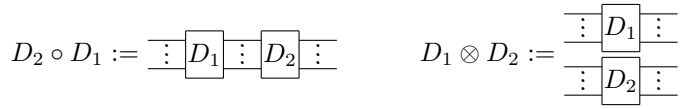
*A **diagram** over S is an equivalence class of diagrammatic terms, where the equivalence relation is the one generated by the coherence axioms of strict symmetric monoidal categories, under the standard interpretation of string diagrams as morphisms in strict symmetric monoidal categories.*

▶ **Definition 3** ([15]). *A symmetric monoidal theory (SMT) is a pair of sets (S, E), where*

- *The set S is a signature, whose elements are also referred to as the generators or the spiders;*
- *The set $E = \{D_j \sim D'_j\}_{j \in \mathcal{J}}$ is a set of pairs of diagrams over S, where each pair relates two diagrams with the same numbers of inputs and outputs. These pairs are called the relations or rewrite rules;*

▶ **Definition 4.** *For any symmetric monoidal theory (S, E), there is a corresponding PROP $\langle S \rangle / E$, where:*

- *Morphisms from n to m are equivalence classes of diagrams over S with n input wires and m output wires, where the equivalence relation is the smallest one containing the axioms E while satisfying the laws of transitivity and substitution;*
- *Composition is plugging together of wires, and monoidal product is side-by-side juxtaposition:*



▶ **Definition 5.** *If there is a PROP isomorphism $\langle S \rangle / E \cong \mathcal{C}$, we say (S, E) **presents** $\mathcal{C}$.*

Because a presentation, in this sense, lends itself to diagrammatic representation so naturally, we use **diagrammatic language** for $\mathcal{C}$ as a synonym for a presentation of $\mathcal{C}$.

Often we have a concrete PROP $\mathcal{C}$ in mind (for example, the PROP Mat_R of finite matrices over a ring R), and we want to find a presentation for it. It is useful, then, to think of this as constructing a "syntax category" $\langle S \rangle / E$, and a "semantics functor" $[-] : \langle S \rangle / E \to \mathcal{C}$. If $[-]$ is well-defined as a PROP morphism, we say that E is *sound* for $\mathcal{C}$. If it is full, we say that S is *universal* for $\mathcal{C}$. If it is faithful, we say that E is *complete* for $\mathcal{C}$. If it is all three at the same time, then it is a PROP isomorphism, and so we have a genuine presentation of $\mathcal{C}$.

The simplest useful examples of presentations of PROPs are Lawvere theories of well-behaved algebraic theories, which we now define.

▶ **Definition 6** (Lawvere Theory). *A Lawvere theory is a PROP $\mathbb{L}$ whose monoidal product is also its category-theoretical product, admitting a strictly identity-on-objects and product-preserving functor $\aleph_0 \to L$, where $\aleph_0$ is a skeleton of the category of finite sets and functions.*

Intuitively, a Lawvere theory is a PROP equipped with a cartesian structure (of copying and deleting).

A model of a Lawvere theory $\mathbb{L}$ is a finite product preserving functor $\mathbb{L} \to \mathsf{Set}$. The models of a fixed Lawvere theory form a category $\mathsf{Mod}(\mathbb{L})$.

▶ **Proposition 7** (Standard Fact on Lawvere Theories. See for example [16]). *Let T be a monad on Set that is finitary (i.e. it preserves filtered colimits). Then there is a Lawvere theory $\mathbb{L}_T$, defined by the equational theory of T, such that $\mathsf{Mod}(\mathbb{L}_T) \simeq \mathsf{Alg}(T)$ where $\mathsf{Alg}(T)$ denotes the Eilenberg-Moore category of T. Moreover, $\mathbb{L}_T$ is equivalent to the opposite of $\mathrm{Kl}\, T|_{fin}$, the Kleisli category of T restricted to finite objects.*

For instance, suppose we want to find a presentation for the category Mat_R of finite matrices over a ring R. Let T be the monad sending a set X to the free R-module generated by X. Then Mat_R is equivalent to the opposite of $\mathrm{Kl}\, T|_{fin}$. The Lawvere theory for finite-dimensional R-modules is therefore equivalent to Mat_R. But the Lawvere theory is also, in itself, a neatly presented PROP: the generators are the operations in the Lawvere theory

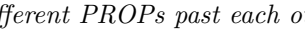
(copy, delete, add, zero, scaling by r) and the rewrite rules are the usual axioms of R-modules. This yields a diagrammatic presentation of the PROP Mat_R, as we wanted.

However, in the GLA literature, the target PROP is usually more complicated than the Kleisli category of an algebraic monad. In particular, it usually has a dagger structure. Building presentations of such PROPs requires us to be able to synthesize larger PROPs out of smaller ones. The simplest way to do this is via PROP sum.

▶ **Definition 8** ([73]). *A sum of PROPs $\mathcal{C} + \mathcal{D}$ is their coproduct in the category of PROPs.*

▶ **Proposition 9** ([73]). *If $\mathcal{C}$ has a presentation (S_1, E_1), and $\mathcal{D}$ has a presentation (S_2, E_2), then $\mathcal{C} + \mathcal{D}$ has presentation $(S_1 \sqcup S_2, E_1 \sqcup E_2)$*

In the sum of PROPs, morphisms coming from two different PROPs of course do not interact. To endow interactive behaviour on morphisms of $\mathcal{C}$ and $\mathcal{D}$, we need the notion of distributive laws of PROPs.

▶ **Definition 10.** *A distributive law is a rewrite rule that distributes morphisms from the two different PROPs past each other:*  *If E is a collection of such rewrites, then $\mathcal{C} +_E \mathcal{D}$ denotes the PROP presented by $(S_1 \sqcup S_2, E_1 \sqcup E_2 \sqcup E)$.*

Note, this is not the traditional way in which distributive laws and their resulting categories are defined, but in the case where $\mathcal{C}$ and $\mathcal{D}$ admit presentations (which is the only case we encounter in this paper), this is in fact equivalent to the traditional definition [53].

2.2 Algebraic Geometry

In reviewing the basic algebraic geometry tools required for our purposes, we follow standard texts [22, 39, 25].

We denote by k an arbitrary field, and $\overline{k}$ its algebraic closure. Likewise we denote by $\mathbb{F}_q$ a finite field of size $q = p^n$ for some prime p, and $\overline{\mathbb{F}}_q$ its algebraic closure. The category of finitely generated commutative algebras over k is denoted $\mathsf{CAlg}_k^{\mathsf{f.g.}}$.

2.2.1 Varieties and Polynomial Mappings

The basic objects we will be concerned with are affine varieties. Intuitively, we think of an affine variety as a place where certain polynomials evaluate to zero. If, say, a set S of polynomials all evaluate to zero at a certain place in space, then it's clear that any linear combination of polynomials in S also evaluate to zero. Thus, we think of affine varieties as being affiliated to *ideals* of polynomials, rather than just to sets of polynomials.

▶ **Definition 11.** *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal. The **zero-set** of I is $V(I) \coloneqq \{\mathbf{a} \in \overline{k}^n \mid \forall f \in I, f(\mathbf{a}) = 0\}$ An **affine variety** over k is a zero-set of any ideal of polynomials.*

Notice that a variety is a subset in a space over the *algebraic closure* of the base field. This is necessary, because in general the roots of polynomials exist only in the algebraic closure. However, often we need to reason about the restriction of a variety, either to the base field, or to some intermediate field. This is captured in the following notion:

▶ **Definition 12.** *For an intermediate field $k \subset k' \subset \overline{k}$, the **$k'$-rational locus** of a variety $V \subset \overline{k}^n$ is $V^{(k')} \coloneqq V \cap (k')^n$; i.e. the set of points in V whose coordinates lie entirely in k'. When we say "a rational locus over k" without naming an intermediate field, we mean the k-rational locus of a variety over k.*

We now define the category in which these geometric objects live:

▶ **Definition 13.** *Let $V \subset \overline{k}^n$ and $W \subset \overline{k}^m$ be affine varieties over k. A function $\phi : V \to W$ is a **k-polynomial mapping** if there exists a polynomial tuple $f = (f_1, \ldots, f_m) \in (k[x_1, \ldots, x_n])^m$ such that $\phi(a_1, \ldots, a_n) = f(a_1, \ldots, a_n)$ for all $(a_1, \ldots, a_n) \in V$.*

*The polynomial mappings from V to k itself form a commutative algebra over k, called the **coordinate ring** of V, and denoted $k[V]$.*

▶ **Definition 14** ([22])**.** *For each fixed field k, the polynomial mappings between affine varieties over k form a category AffVar_k. For each intermediate field $k \subset k' \subset \overline{k}$, there is a full subcategory $\mathsf{AffVar}_k^{(k')} \subset \mathsf{AffVar}_k$ consisting of affine varieties that are k'-rational.*

The following gives a basic rule regarding how the variety V, its set of vanishing polynomials, and its coordinate ring are related to each other, thus yielding a functorial correspondence:

▶ **Proposition 15** ([22])**.** *The coordinate ring of V can be computed as $k[V] \cong k[x_1, \ldots, x_n]/I_k(V)$ where $I_k(V)$ is the ideal of polynomials that vanish on V. The coordinate ring map $k[-]$ is a fully-faithful contravariant functor $\mathsf{AffVar}_k^{op} \to \mathsf{CAlg}_k^{\mathsf{f.g.}}$.*

2.2.2 The Nullstellensatze

We've seen how the coordinate ring functor $k[-]$ associates an algebraic object to each geometric shape (the variety). We will now show that this association is not an equivalence of categories, but very close to being one. The connection in the opposite direction is provided

by Hilbert's Nullstellensatz for algebraically closed fields, and its alternative formulation for finite fields. It's not surprising that we will use these in Section 6 to move from graphical commutative algebra to graphical algebraic geometry.

▶ **Proposition 16** (Hilbert's Nullstellensatz)**.** *[39] Let k be an algebraically closed field, and $J \subset k[x_1, \ldots, x_n]$ an ideal. Then $I_k(V(J)) = \sqrt{J}$.*

We want to see this as an equivalence between a category of algebraic objects and one of geometric objects. So let us now translate this into the language of category theory:

▶ **Definition 17.** *A commutative algebra is **reduced** if it has no nonzero nilpotents. The full subcategory of $\mathsf{CAlg}_k^{\mathsf{f.g.}}$ consisting of reduced commutative algebras is denoted $\mathsf{CAlg}_k^{\mathsf{f.g.,\ red.}}$.*

▶ **Proposition 18** ([25])**.** *The functor $(-)_{red} : \mathsf{CAlg}_k^{\mathsf{f.g.}} \to \mathsf{CAlg}_k^{\mathsf{f.g.,\ red.}}$ sending $A \mapsto A/\operatorname{nil} A$ is the left-adjoint of the inclusion $\mathsf{CAlg}_k^{\mathsf{f.g.,\ red.}} \hookrightarrow \mathsf{CAlg}_k^{\mathsf{f.g.}}$.*

▶ **Proposition 19** (Hilbert's Nullstellensatz, Categorically)**.** *[25] If k is an algebraically closed field, then the coordinate ring functor $k[-]$ gives an anti-equivalence of categories*

$$k[-] : \mathsf{AffVar}_k^{op} \simeq \mathsf{CAlg}_k^{\mathsf{f.g.,\ red.}} \tag{2}$$

Denote its inverse functor (up to isomorphism) as $\operatorname{Spec} : \mathsf{CAlg}_k^{\mathsf{f.g.,\ red.}} \to \mathsf{AffVar}_k^{op}$

We now turn our attention to the case of finite fields. While affine varieties over algebraically closed fields are "rare" among the subsets of k^n, *any subset* of $\mathbb{F}_q^n$ is an affine variety over $\mathbb{F}_q$:

▶ **Proposition 20.** *The category $\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}$ of rational loci over $\mathbb{F}_q$ is equivalent to the category* FinSet *of finite sets and relations.*

Proof. It is well known [33, 56] that any subset of $\mathbb{F}_q^n$ is an affine variety, and any function between them is polynomial. ◀

However, our results are most easily read if we think in terms of polynomial constraints and morphisms, rather than subsets and functions. We retain the notation $\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}$ as a reminder of this "attitude".

▶ **Proposition 21** (Finite Field Nullstellensatz)**.** *[32] Let J be any ideal of n-variable polynomials over a finite field. Then $I_{\mathbb{F}_q}(V^{(\mathbb{F}_q)}(J)) = J + \Gamma_q^n$ where $\Gamma_q^n \coloneqq (x_1^q - x_1, \ldots, x_n^q - x_n)$.*

As with Hilbert's Nullstellensatz, we now translate the Finite Field Nullstellensatz into a statement about categorical equivalence:

▶ **Definition 22.** *A commutative algebra A over $\mathbb{F}_q$ is called **q-reduced**[1] if for every $f \in A$, $f^q = f$. The q-reduced algebras form a full subcategory $\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.,\ qred.}} \subset \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}}$.*

▶ **Definition 23.** *For a finite field $\mathbb{F}_q$ and a finitely generated commutative algebra A over it, define its q-radical as the ideal $\Gamma_q(A) \coloneqq (f^q - f : f \in A)$.*

▶ Remark 24. If $\{x_1, ..., x_n\}$ generate A over $\mathbb{F}_q$ then $\Gamma_q(A) = \Gamma_q^n = (x_1^q - x_1, \ldots, x_n^q - x_n)$.

[1] This is our terminology. In finite field error correction literature this is just called "reduced" [32], but we need to avoid confusion with the aforementioned more standard notion of reduction from algebra.

▶ **Proposition 25.** *The functor* $(-)_{qred} : \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}} \to \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.,\ qred.}}$ *mapping* $A \mapsto A/\Gamma_q(A)$ *is left-adjoint to the inclusion.*

▶ **Proposition 26** (Finite Field Nullstellensatz, Categorically)**.** *For a finite field* $\mathbb{F}_q$*, the coordinate ring functor* $\mathbb{F}_q[-]$ *restricted to the* $\mathbb{F}_q$*-rational loci defines an equivalence of categories*

$$\mathbb{F}_q[-] : \mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)^{op}} \simeq \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.,\ qred.}} \tag{3}$$

Denote its inverse functor as $\mathrm{Spec} : \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.,\ qred.}} \simeq \mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)^{op}}$.

Proof. Since $\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)} \simeq \mathsf{FinSet}$, this is really a nonbinary version of the Stone duality between finite boolean algebras and finite sets [46]. For a proof along this approach see [56]. For a more algebraic perspective see also [36]. ◀

▶ **Notation.** *The q-reduced versions of the free commutative algebras will be denoted* $\mathcal{R}_q^n := S_{\mathbb{F}_q}(n)/\Gamma_q^n$*, and the full subcategory of* $\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}}$ *consisting of such algebras will be denoted* $\mathsf{Poly}_{\mathbb{F}_q}^{\mathsf{qred.}}$*. Let* $F_q : \mathsf{Poly}_{\mathbb{F}_q}^{\mathsf{qred.}} \hookrightarrow \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}}$ *be its inclusion functor. This will used in Section 4 when we define the span-semantics target of graphical algebraic geometry.*

3 The Polynomial Fragment

Consider the free-forgetful adjunction $S_k \dashv U : \mathsf{CAlg}_k \leftrightharpoons \mathsf{Set}$. On finite sets n, $S_k(n)$ is just the polynomial algebra $k[x_1, \ldots, x_n]$. Then let $T_k = U \circ S_k$ be the associated monad.

▶ **Definition 27.** *Let* Poly_k *be the full subcategory of* $\mathsf{CAlg}_k^{\mathsf{f.g.}}$ *given by the image of* S_k *restricted to finite sets. In other words, let* Poly_k *be the Kleisli category of* T_k *restricted to finite sets, corresponding to the polynomial algebras* $k[x_1, \ldots, x_n]$*.*

▶ **Notation** (Sharps and Flats)**.** *It is worth being careful about the equivalence between* Poly_k *and* $\mathrm{Kl}\, T_k|_{fin}$*, the Kleisli category of* T_k *restricted to finite objects. If* $f \in (k[x_1, \ldots, x_n])^m$ *is a tuple of polynomials, denote by* $f^\#$ *its unique extension to an algebra morphism* $S_k(m) \to S_k(n)$*. This acts on a polynomial* $g(y_1, \ldots, y_m)$ *by substituting* y_j *with* $f_j(x_1, \ldots, x_n)$*. Notice the contravariance implicit in this: algebra morphisms point and compose in the opposite direction than polynomials. Conversely, if* $\varphi : S_k(m) \to S_k(n)$ *is an algebra morphism, denote by* $\varphi^\flat : m \to S_k(n)$ *the corresponding tuple of polynomials defined by* $\varphi_j^\flat = \varphi(x_j)$*.*

Recall from Section 2.1 that for any finitary monad T, there is an equivalence $\mathrm{Kl}\, T|_{fin} \simeq \mathbb{L}_T^{op}$, where $\mathbb{L}_T$ denotes the Lawvere theory corresponding to T. In our case, the Lawvere theory is that of commutative algebras over k, which we denote $\mathbb{L}_{CAlg_k}$. Then:

▶ **Proposition 28.** *The generators and rewrite rules depicted in Figure 2 form a presentation of the PROP* $\mathbb{L}_{CAlg_k} \simeq \mathsf{Poly}_k^{op}$*.*

Proof. This is the standard presentation of the Lawvere theory of commutative algebras over k, with the following interpretation for the diagrams: the white dots  and are read as copying and its counit deleting, the grey dots  and are read as adding and its unit zero, the yellow half-moons  and are read as multiplication and its unit one, and the "squeeze" gadgets  labelled by elements of k are read as the scaling operation by the field k on a k-commutative algebra. The rewrite rules of the Lawvere theory guarantee that each (finite variable) polynomial with coefficients in k can be expressed as a diagram in this language uniquely up to rewrites. ◀

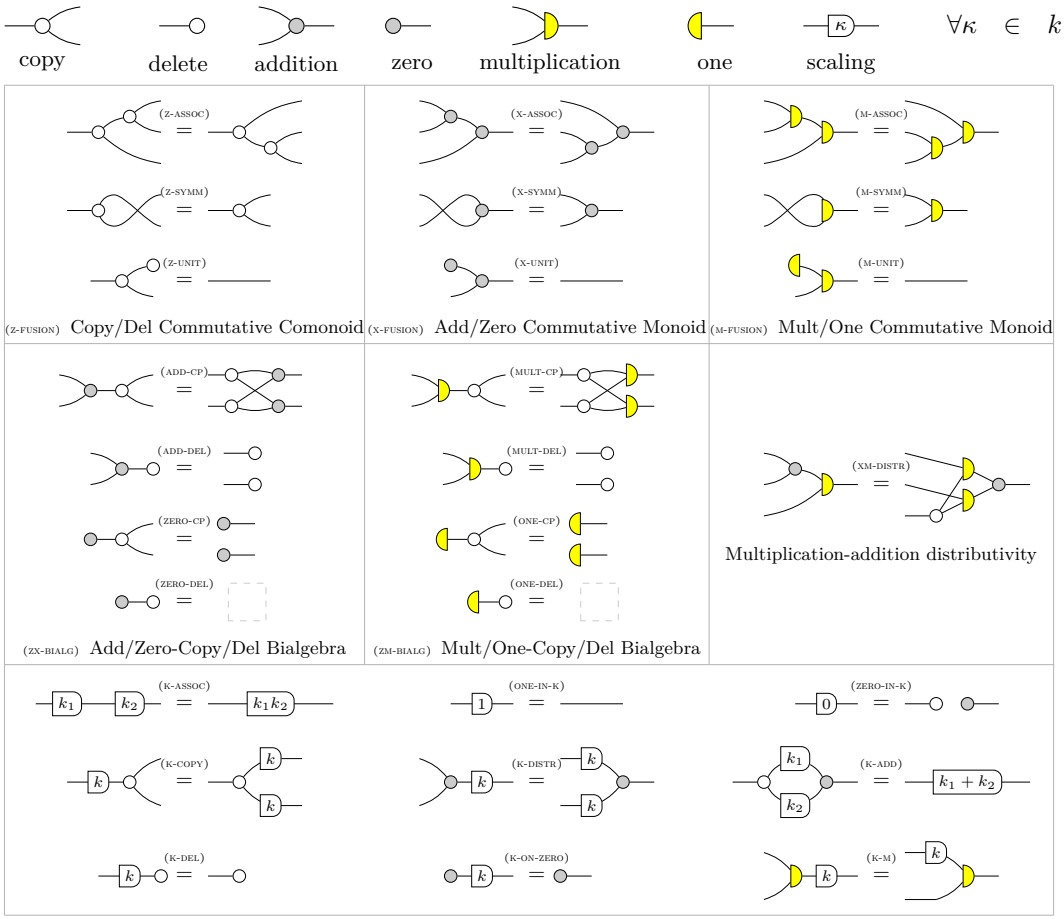
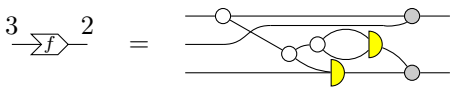


Figure 2 The Theory $\mathbb{L}_{CAlg_K}$ of Commutative Algebras over k.

▶ **Notation** (Scalable Notation). *We can now unambiguously introduce scalable notation for diagrams in $\mathbb{L}_{CAlg_k}$. For $n \in \mathbb{N}$ we define n-labelled wire and gates, as depicted in Figure 3.*

For every polynomial tuple $f \in (k[x_1, \ldots, x_n])^m$, we abbreviate the diagram (unique up to rewrites) that corresponds to $f^\#$ in $\mathbb{L}_{CAlg_k}$ as n —f— m.

For example, for the polynomial tuple $f = (x_1 + x_2, x_1x_3 + x_1^2) \in (k[x_1, x_2, x_3])^2$, we have



(4)

Notice that the contravariance between morphism composition and polynomial composition has been inherited by our diagrammatic language. Diagrammatically, everything is drawn in the direction of the Lawvere theory $\mathbb{L}_{CAlg_k}$. So n —f— m corresponds to a tuple of polynomials $(k[x_1, \ldots, x_n])^m$, which corresponds to a morphism $f^\# \in \mathsf{Poly}_k(m, n)$. The necessity for this will become clear in Section 6 when we construct the language for spans of affine varieties.

▶ **Proposition 29.** *From the rewrite rules of $\mathbb{L}_{CAlg_k}$ it is possible to derive the rewrite rules about polynomials in the scalable notation, as depicted in Figure 4.*

Proof. See Appendix A in the full version of this paper. ◀

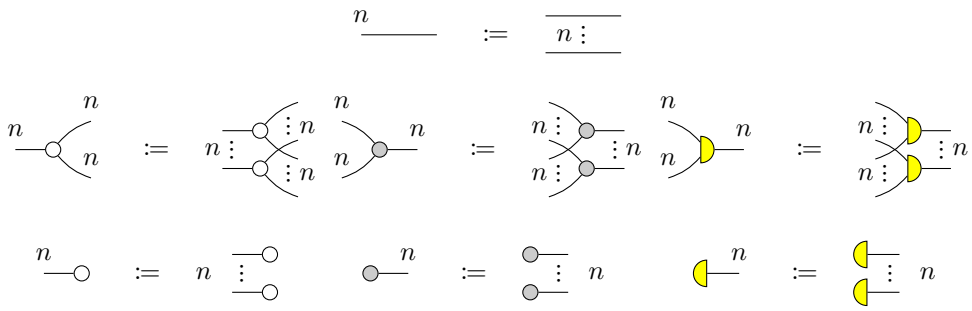


Figure 3 Scalable Notation for $\mathbb{L}_{CAlg_k}$.

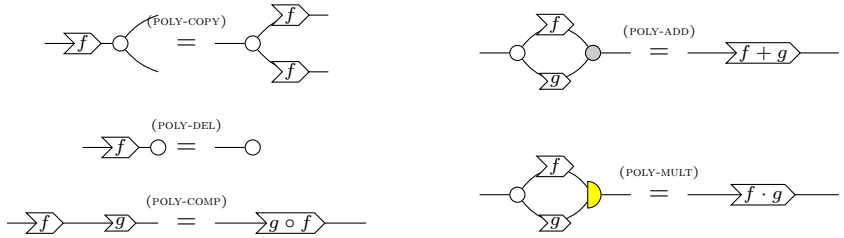
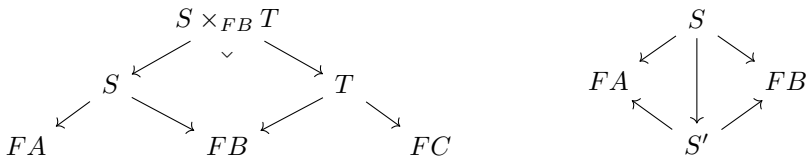


Figure 4 Rules for Scalable Notation.

4 (Co)span Semantics

In this section we describe the target semantics categories of graphical commutative algebra and graphical algebraic geometry. We begin by recalling the definition of structured (co)spans, which is of particular importance here:

▶ **Definition 30** ([5]). *Let $F : \mathcal{A} \to \mathcal{X}$ be a functor where $\mathcal{X}$ is a category with pullbacks. Define the bicategory of* ***structured spans*** *${}_F\mathbf{Sp}(\mathcal{X})$: its objects are those of $\mathcal{A}$; its 1-cells are pairs of $\mathcal{X}$ morphisms of the form $FA \leftarrow S \to FB$ with $A, B \in \mathcal{A}$, which compose via pullback (shown on the left), and its 2-cells are commuting diagrams (shown on the right):*



Quotienting the categories of 1-cells of ${}_F\mathbf{Sp}(\mathcal{X})$ by the 2-isomorphisms, we obtain a 1-category ${}_F\,\mathrm{Span}(\mathcal{X})$, whose objects are those of $\mathcal{A}$, and morphisms are isomorphism classes of spans $FA \leftarrow S \to FB$.

For a functor $F : \mathcal{A} \to \mathcal{X}$ where $\mathcal{X}$ has pushouts, the bicategory of ***structured cospans*** *${}_F\mathbf{Csp}(\mathcal{X})$ and the 1-category ${}_F\,\mathrm{Cosp}(\mathcal{X})$ are defined dually to the above, with 1-cells of the form $FA \to S \leftarrow FB$ and composition given by pushout instead of pullback.*

The following is an immediate well-known consequence of the dual nature of the definitions of structured cospans and spans:

▶ **Proposition 31.** *If $F : \mathcal{A} \to \mathcal{X}$ is a functor and $\mathcal{X}$ has pullbacks, then $\mathcal{X}^{op}$ has pushouts, and there is an equivalence ${}_F\,\mathrm{Span}(\mathcal{X}) \simeq {}_{F^{op}}\,\mathrm{Cosp}(\mathcal{X}^{op})$ where $F^{op} : \mathcal{A}^{op} \to \mathcal{X}^{op}$ is the dual functor of F.*

▶ **Definition 32.** *In the structured span construct, the category $\mathcal{A}$ embeds into ${}_F\,\mathrm{Span}(\mathcal{X})$ both covariantly by the graph embedding:*

$$G_{\mathrm{Span}}: \quad \left(A \xrightarrow{f} B\right) \quad \mapsto \quad \left(FA = FA \xrightarrow{Ff} FB\right) \tag{5}$$

and contravariantly by the cograph embedding:

$$G^{op}_{\mathrm{Span}}: \quad \left(A \xrightarrow{f} B\right) \quad \mapsto \quad \left(FB \xleftarrow{Ff} FA = FA\right) \tag{6}$$

Similarly in the structured cospan construct, the category $\mathcal{A}$ embeds into ${}_F\,\mathrm{Cosp}(\mathcal{X})$ covariantly by the graph embedding G_{Cosp} and contravariantly by the cograph embedding G^{op}_{Cosp}.

The following are not new results, but are recalled here as we shall need it immediately.

▶ **Proposition 33.** *For any field k, the category $\mathsf{CAlg}_k^{\mathsf{f.g.}}$ of finitely-generated commutative algebras over k has pushouts.*

Proof. To construct the pushout: suppose $A = k[x_1, \ldots, x_n]/I_A$, $B = k[y_1, \ldots, y_m]/I_B$, $C = k[z_1, \ldots, z_r]/I_C$ with maps $A \xleftarrow{\phi} C \xrightarrow{\psi} B$. Then

$$A \otimes_C B = \frac{k[x_1, \ldots, x_n, y_1, \ldots, y_m]}{I_A + I_B + \sum_{i=1}^{r}(\phi(z_i) - \psi(z_i))} \tag{7}$$

Where, by a slight abuse of notation, I_A, I_B denote the images of I_A, I_B via the embeddings $k[x_1, \ldots, x_n] \hookrightarrow k[x_1, \ldots, x_n, y_1, \ldots, y_m] \hookleftarrow k[y_1, \ldots, y_m]$ ◀

▶ **Corollary 34.** *If k is either algebraically closed or finite, the category $\mathsf{AffVar}_k^{(k)}$ has pullbacks.*

We are now ready to define the three target semantics categories we meet in this paper: one for graphical commutative algebra, and two for graphical algebraic geometry. As we shall see later in proposition 37, structuring the (co)spans with respect to the appropriate inclusion functors ensures that each of these semantics categories is a PROP.

▶ **Definition 35** (Semantics Category for Graphical Commutative Algebra)**.** *Consider the structured cospan category ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$, where F is the inclusion of the free commutative algebras into the finitely generated commutative algebras. The objects of ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$ are polynomial algebras $k[x_1, \ldots, x_n]$, and its morphisms are structured cospans*

$$k[x_1, \ldots, x_n] \xrightarrow{f} A \xleftarrow{g} k[y_1, \ldots, y_m] \tag{8}$$

where A is some finitely generated commutative algebra over k.

▶ **Definition 36** (Semantics Category of Graphical Algebraic Geometry)**.** *Consider the structured span categories ${}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$ and ${}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)})$, where G and G_q are the inclusions of the free spaces k^n, $\mathbb{F}_q^n$ in the categories of affine varieties. The objects are free spaces k^n and $\mathbb{F}_q^n$ respectively, and morphisms in ${}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$ are spans of polynomial maps*

$$k^n \leftarrow V \rightarrow k^m \quad , \quad \mathbb{F}_q^n \leftarrow W \rightarrow \mathbb{F}_q^m \tag{9}$$

respectively, where V is an affine variety over k and W is a rational locus over $\mathbb{F}_q$.

▶ **Proposition 37.** *The three semantics categories constructed above are PROPs.*

Proof. The objects of ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}\cdot})$ are the polynomial algebras $k[x_1,\ldots,x_n]$, the monoidal product on objects is $k[x_1,\ldots,x_n]\otimes k[y_1,\ldots,y_m]=k[x_1,\ldots,x_n,y_1,\ldots,y_m]$; and on morphisms by the tensor-product of algebra morphisms on both legs of the cospans.

The braidings are given by the graphs of

$$\sigma : k[x_1,\ldots,x_n,y_1,\ldots,y_m]\to k[y_1,\ldots,y_m,x_1,\ldots,x_n] \tag{10}$$

The proofs for the other two categories are similar. ◀

The semantics categories for GCA and GAG are related, and we now explain this relation. First, note that the structured (co)span construction is "functorial":

▶ **Proposition 38** ([5]). *Given $F:\mathcal{A}\to\mathcal{X}$ and $F':\mathcal{B}\to\mathcal{Y}$, with $\mathcal{X},\mathcal{Y}$ both having pullbacks, if there is a commutative square of functors* $\begin{array}{ccc}\mathcal{A} & \xrightarrow{F} & \mathcal{X}\\ {\scriptstyle\Phi}\downarrow & & \downarrow{\scriptstyle\Psi}\\ \mathcal{B} & \xrightarrow[F']{} & \mathcal{Y}\end{array}$ *and Ψ preserves pullbacks, then there is a functor* $\mathrm{Span}(\Psi,\Phi):{}_F\,\mathrm{Span}(\mathcal{X})\to{}_{F'}\,\mathrm{Span}(\mathcal{Y})$ *which sends the structured span $(FA_1\xleftarrow{f}X\xrightarrow{g}FA_2)$ to a structured span in ${}_F\,\mathrm{Span}(\mathcal{Y})$ by hitting everything with Ψ. The dual statement is true for structured cospans.*

▶ **Corollary 39.** *In particular, in the situation of proposition 38, if Φ,Ψ are both equivalences of categories, then the induced functor ${}_F\,\mathrm{Span}(\mathcal{X})\to{}_{F'}\,\mathrm{Span}(\mathcal{Y})$ is also an equivalence of categories.*

Now to explain how our three semantics categories are related: recall from proposition 25 from the background section that there are functors

$$\begin{aligned}(-)_{red} &: \mathsf{CAlg}_k^{\mathsf{f.g.}\cdot}\to\mathsf{CAlg}_k^{\mathsf{f.g.},\ \mathsf{red.}}\\ (-)_{qred} &: \mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}\cdot}\to\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.},\ \mathsf{qred.}}\end{aligned} \tag{11}$$

each of which are left-adjoint to the inclusion in the opposite direction, and therefore they preserve pushouts. Moreover, they commute with the inclusions of the polynomial algebras $F:\mathsf{Poly}_k\hookrightarrow\mathsf{CAlg}_k^{\mathsf{f.g.}\cdot}$ and $F_q:\mathsf{Poly}_{\mathbb{F}_q}^{\mathsf{qred.}}\hookrightarrow\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}\cdot}$. The functoriality of the structured cospan construction (proposition 38) implies there are functors

$$\begin{aligned}{}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}\cdot}) &\xrightarrow{(-)_{red}} {}_F\,\mathrm{Cosp}\left(\mathsf{CAlg}_k^{\mathsf{f.g.},\ \mathsf{red.}}\right)\\ {}_F\,\mathrm{Cosp}(\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}\cdot}) &\xrightarrow{(-)_{qred}} {}_{F_q}\,\mathrm{Cosp}\left(\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.},\ \mathsf{qred.}}\right)\end{aligned} \tag{12}$$

Now recall that if k is algebraically closed, then there is an anti-equivalence (19)

$$\mathrm{Spec}:\mathsf{CAlg}_k^{\mathsf{f.g.},\ \mathsf{red.}}\simeq\mathsf{AffVar}_k^{op} \tag{13}$$

and if $\mathbb{F}_q$ is finite of size q, then there is an anti-equivalence (26)

$$\mathrm{Spec}:\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.},\ \mathsf{qred.}}\simeq\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)^{op}} \tag{14}$$

Moreover, it is easy to check that these commute with the inclusions F,F_q,G,G_q in the obvious way. So using the duality between span and cospan (31) we obtain maps

$$\begin{aligned}{}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}\cdot}) &\xrightarrow{(-)_{red}} {}_F\,\mathrm{Cosp}\left(\mathsf{CAlg}_k^{\mathsf{f.g.},\ \mathsf{red.}}\right)\simeq{}_G\,\mathrm{Span}(\mathsf{AffVar}_k)\\ {}_F\,\mathrm{Cosp}(\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}\cdot}) &\xrightarrow{(-)_{qred}} {}_{F_q}\,\mathrm{Cosp}\left(\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.},\ \mathsf{qred.}}\right)\simeq{}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)})\end{aligned} \tag{15}$$

5 Graphical Commutative Algebra

We now work towards a diagrammatic presentation, which we refer to as Graphical Commutative Algebra (GCA), of the structured cospan category ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$. All lemmas, propositions, and theorems in this section are proven in Appendix B in the full version of this paper, unless otherwise specified.

5.1 Defining the Calculus GCA_k

Our language GCA_k will be defined essentially as an amalgamation, modulo certain additional rewrite rules, of the Lawvere theory $\mathbb{L}_{CAlg_k}$ and its own opposite, so as to become a dagger-category:

▶ **Definition 40.** *Construct the PROP* $\mathsf{GCA}_k = \mathbb{L}_{CAlg_k} +_E \mathbb{L}_{CAlg_k}^{op}$ *where* E *is the set of rewrite rules displayed in Figure 5.*

We supply this PROP with an intended semantics functor $[-] : \mathsf{GCA}_k \to {}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$ *by mapping each generator in* $\mathbb{L}_{CAlg_k}$ *via the cograph embedding, and each generator in* $\mathbb{L}_{CAlg_k}^{op}$ *via the graph embedding.*

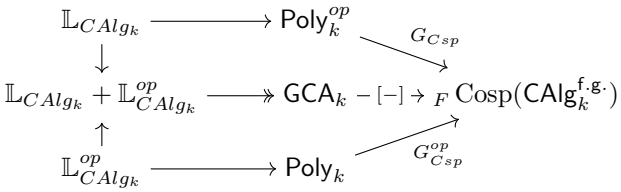
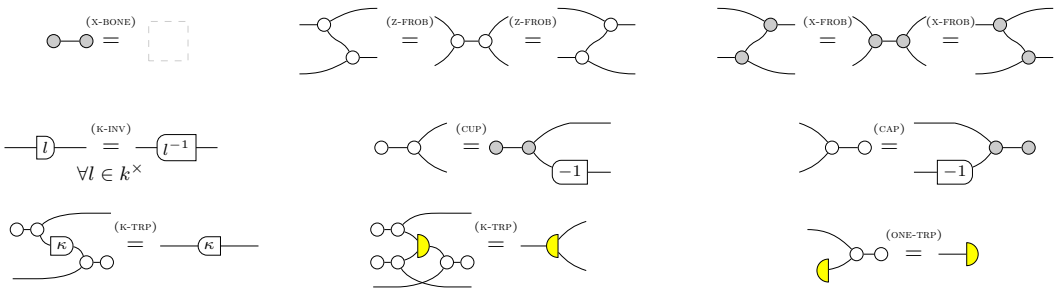


Figure 5 Additional Rewrite Rules for GCA_k.

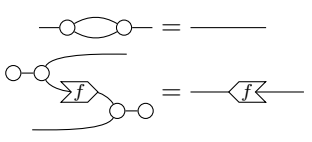


We must first prove that the functor $[-]$ is well-defined, which is equivalent to the following statement:

▶ **Proposition 41** (Cospan Soundness)**.** *Every rewrite rule in* E *is sound with respect to the intended semantics* $[-]$*.*

Once well-definedness has been established, we can derive the following useful rewrite rules from the given rules:

▶ **Lemma 42.** *The following are some useful derived rules of* GCA_k*:*



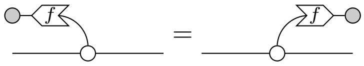
(special)

$\forall f \in (k[x_1, \ldots, x_n])^m$ (poly-trp)

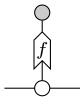
5.2 Universality and Completeness

To consider the nature of the (essential) image of $[-]$ in ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$, we first prove the following sequence of lemmas in the language GCA_k, which will allow us to introduce an essential piece of syntactic sugar for summarizing a diagram.

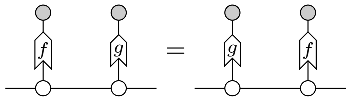
▶ **Lemma 43.** *For any $f \in k[x_1, \ldots, x_n]$, the following is derivable from the rewrite rules of GCA_k:*



(16)

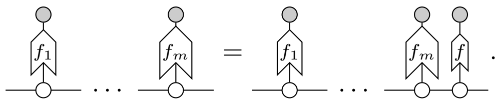
So we just speak of gadgets of the form 

▶ **Lemma 44.** *For any $f, g \in k[x_1, \ldots, x_n]$, the following is derivable from the rewrite rules of GCA_k:*



(17)

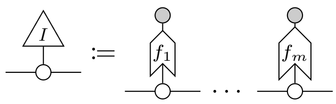
▶ **Lemma 45.** *Let $f_1, \ldots, f_m \in k[x_1, \ldots, x_n]$, and pick some f inside the ideal $(f_1, \ldots, f_m)$. Then the following is derivable from the rewrite rules of GCA_k:*



(18)

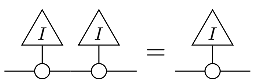
Lemmas 44 and 45 mean that we can coherently define the following gadget:

▶ **Definition 46.** *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal. Since $k[x_1, \ldots, x_n]$ is Noetherian [25], there exists a finite set of polynomials $f_1, \ldots, f_m \in k[x_1, \ldots, x_n]$ such that $I = (f_1, \ldots, f_m)$. Then define*



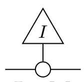
(19)

Note that neither the choice of the basis $f_1, \ldots, f_m$, nor the ordering of them, matters in the definition, since any two such choices give diagrams which are rewritable to each other per lemmas 44 and 45. For concreteness, we may always choose the reduced Grobner basis with respect to some chosen monomial ordering.

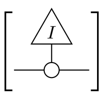
▶ **Corollary 47.** *Idempotence* 

is provable in GCA_k.

We now explain the semantics of these gadgets.

▶ **Proposition 48** (Quotient Gadgets)**.** *The semantics of the gadget* 

is (any finite presentation of) the finitely generated commutative algebra defined by I. More precisely, if $I \subset k[x_1, \ldots, x_n]$ is any ideal, then



$$= (S_k(n) \twoheadrightarrow S_k(n)/I \twoheadleftarrow S_k(n)) \tag{20}$$

where both legs of the cospan are the canonical projections π_I against I.

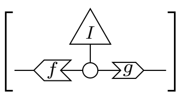
This is essentially what brings us from a language for Poly_k to a language that has full information about $\mathsf{CAlg}_k^{\mathsf{f.g.}}$, which forces us to introduce the following notation:

▶ **Notation.** *If $A = S_k(n)/I$, and $f \in (k[x_1, \ldots, x_n])^m$, denote by $f_I^\# : S_k(m) \to A$ the algebra morphism one gets by passing to the quotient:*

$$S_k(m) \xrightarrow{f^\#} S_k(n) \twoheadrightarrow S_k(n)/I = A \tag{21}$$

Or equivalently, the algebra morphism one gets by reducing f against I component-wise to obtain $f_I \in A^m$, and then taking its extension to an algebra morphism $f_I^\# : S_k(m) \to A$.

▶ **Proposition 49** (Cospan Universality)**.** *For any ideal $I \subset k[x_1, \ldots, x_n]$ and tuples of polynomials $f \in (k[x_1, \ldots, x_n])^m$ and $g \in (k[x_1, \ldots, x_n])^{m'}$,*

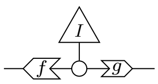


$$= \left(S_k(m) \xrightarrow{f_I^\#} A \xleftarrow{g_I^\#} S_k(m') \right) \tag{22}$$

where $A = S_k(n)/I$. Since every algebra morphism $S_k(m) \to A$ is equal to $f_I^\#$ for some f, $[-] : \mathsf{GCA}_k \to_F \mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$ is full.

It remains now to prove the completeness of the language GCA_k. We do this by reducing diagrams to a pseudo-normal form as follows:

▶ **Definition 50.** *A diagram D in GCA_k is said to be in **cospan form** if it is drawn as*



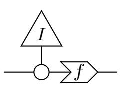
(23)

for some ideal I and some polynomial tuples f, g.

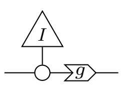
▶ **Proposition 51** (Cospan Normal Form)**.** *Every diagram in GCA_k can be rewritten into cospan form.*

Thus, to show completeness of GCA_k, it suffices to show that, for any two diagrams in cospan form, if they have the same semantics in ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$, then they can be rewritten into each other. We do this via the following lemmas:

▶ **Lemma 52.** *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal, and let $f, g \in (k[x_1, \ldots, x_n])^m$ be any two polynomial tuples such that $f - g \in I^m$. That is, let f, g be such that $f_I^\# = g_I^\#$. Then the following diagram rewrite is derivable in GCA_k:*

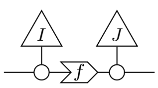


=

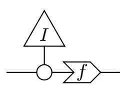


(24)

▶ **Lemma 53.** *Suppose $I \subset S_k(n)$ and $J \subset S_k(m)$ are ideals of polynomials and $f \in (S_k(n))^m$. Suppose $f_I^\# : S_k(m) \to S_k(n)/I$ satisfies $J \subset \ker f_I^\#$. Then the following rewrite holds:*



=

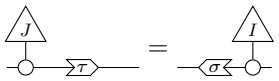


(25)

▶ **Lemma 54.** *Suppose $I \subset S_k(n)$ and $J \subset S_k(m)$ are ideals, $\alpha : S_k(m)/J \to S_k(n)/I$ is an isomorphism of algebras, and σ, τ are a pair of polynomial tuples such that they represent α, α^{-1} respectively, i.e. such that the following commutes:*

$$\begin{array}{ccccc} S_k(m) & \xrightarrow{\pi_J} & S_k(m)/J & \xleftarrow{\pi_J} & S_k(m) \\ {\scriptstyle \sigma^\#}\downarrow & & {\scriptstyle \alpha^{-1}}\uparrow \downarrow{\scriptstyle \alpha} & & \uparrow{\scriptstyle \tau^\#} \\ S_k(n) & \xrightarrow[\pi_I]{} & S_k(n)/I & \xleftarrow[\pi_I]{} & S_k(n) \end{array}$$

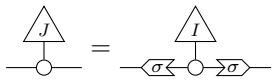
Then the following rewrite holds:



(26)

By composing $\multimap\!\!\triangleright\!\!\sigma\!\!\triangleright\!\!\multimap$ on the right to both sides of the equation in 54, we obtain:

▶ **Corollary 55.** *If $\alpha : S_k(m)/J \to S_k(n)/I$ is an isomorphism of algebras, and $\sigma \in (k[x_1, \ldots, x_n])^m$ represents α as in lemma 54, then*



(27)

Lemmas 52, 53, and 54 suffice to show that any two diagrams in cospan normal form which have the same semantics in ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$ can be rewritten into each other. And therefore:

▶ **Proposition 56.** *The functor $[-]$ is faithful. That is, GCA_k is complete as a language for ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$.*

We have now completed the demonstration of our first main theorem:

▶ **Theorem 57.** *The functor $[-] : \mathsf{GCA}_k \to {}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$ is an isomorphism of PROPs. Thus GCA_k is a sound, universal and complete language for ${}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}})$.*

Proof. Propositions 41, 49, and 56 together imply theorem 57. ◀

6 Graphical Algebraic Geometry

Our focus now turns from algebra to geometry. As mentioned in Section 2, this is where we will make heavy use of the Nullstellensatze to bridge the gap from algebra to geometry.

In this section k will always denote an algebraically closed field, and $\mathbb{F}_q$ will always denote a finite field of size q. All proofs are contained in Appendix C in the full version of this paper, unless otherwise specified.

Recall from equation 15 that as a consequence of the Nullstellensatze, there are two functors

$$\begin{aligned} &{}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}}) \xrightarrow{\mathrm{Spec}(-)_{red}} {}_G\,\mathrm{Span}(\mathsf{AffVar}_k) \\ &{}_F\,\mathrm{Cosp}(\mathsf{CAlg}_{\mathbb{F}_q}^{\mathsf{f.g.}}) \xrightarrow{\mathrm{Spec}(-)_{qred}} {}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}) \end{aligned} \tag{28}$$

We now seek diagrammatic presentations for ${}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$ and ${}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)})$. Since we already have a presentation

$$\mathsf{GCA}_k \cong_F \mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}}), \tag{29}$$

we seek to find GAG_k and GAG_q that fit into the following commutative diagrams:

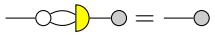
$$\begin{array}{ccc} \mathsf{GCA}_k & \xrightarrow{[-]} & {}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}}) \\ \vdots & & \downarrow{\scriptstyle \mathrm{Spec}(-)_{red}} \\ \mathsf{GAG}_k & \xrightarrow[[-]_k]{} & {}_G\,\mathrm{Span}(\mathsf{AffVar}_k) \end{array} \qquad \begin{array}{ccc} \mathsf{GCA}_k & \xrightarrow{[-]} & {}_F\,\mathrm{Cosp}(\mathsf{CAlg}_k^{\mathsf{f.g.}}) \\ \vdots & & \downarrow{\scriptstyle \mathrm{Spec}(-)_{qred}} \\ \mathsf{GAG}_q & \xrightarrow[[-]_q]{} & {}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}) \end{array}$$

such that $[-]_k$ and $[-]_q$ are both isomorphisms of PROPs.

6.1 GAG over Algebraically Closed Fields

For algebraically closed fields, Hilbert's Nullstellensatz tells us that affine varieties correspond to reduced coordinate rings. So we introduce the following rewrite rule (red), which intuitively should be thought of as "reducing" all rings. This intuition is made formal in lemma 60.

▶ **Definition 58.** *Let* GAG_k *be the quotient of* GCA_k *defined by the following additional rewrite rule:*



(red)

▶ **Proposition 59.** *The PROP morphism*

$$\mathsf{GCA}_k \xrightarrow{\mathrm{Spec}[-]_{red}}{}_G\,\mathrm{Span}(\mathsf{AffVar}_k) \tag{30}$$

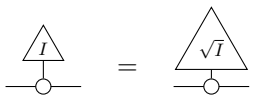
factors through GAG_k*. In other words, rule* (red) *is sound for affine varieties over an algebraically closed field.*

Thus we obtain a PROP morphism

$$\mathsf{GAG}_k \xrightarrow{[-]_k}{}_G\,\mathrm{Span}(\mathsf{AffVar}_k) \tag{31}$$

It remains to show that this morphism is faithful: in other words, that our rewrite rules are complete.

▶ **Lemma 60.** *Let* $I \subset k[x_1, \ldots, x_n]$ *be an ideal. The following is derivable from the rewrite rules of* GAG_k*:*



(32)

Leveraging Hilbert's Nullstellensatz, we see that lemma 60 alone is sufficient to prove completeness for GAG_k (See Appendix C in the full version of this paper for details in proof):

▶ **Proposition 61.** *The semantics map* $\mathsf{GAG}_k \xrightarrow{[-]_k}{}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$ *is faithful. In other words,* GAG_k *is complete for* ${}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$*.*

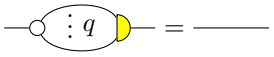
Fullness, on the other hand, is inherited from GCA_k. Thus:

▶ **Theorem 62.** *The semantics map* $\mathsf{GAG}_k \xrightarrow{[-]_k}{}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$ *is an isomorphism of PROPs;* GAG_k *is a universal, sound, and complete language for* ${}_G\,\mathrm{Span}(\mathsf{AffVar}_k)$*.*

6.2 GAG over Finite Fields

The recipe for obtaining a language for spans of rational loci of affine varieties over finite fields is almost identical to the one used for spans of affine varieties over closed fields. We begin by introducing a rewrite rule (qred) which intuitively q-reduces all rings:

▶ **Definition 63.** *Let* GAG_q *be the quotient of* GCA_k *by the following additional rewrite rule:*



(qred)

▶ **Proposition 64.** *The PROP morphism*

$$\mathsf{GCA}_k \xrightarrow{\mathrm{Spec}[-]_{qred}} {}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}) \tag{33}$$

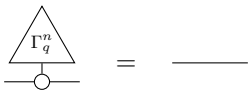
factors through GAG_q*. In other words, the rule qred is sound with respect to* $\mathbb{F}_q$*-rational loci.*

Thus we obtain a PROP morphism

$$\mathsf{GAG}_q \xrightarrow{[-]_q} {}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}) \tag{34}$$

Now to show that this morphism is faithful: again, we start by showing that the extra rule (qred) implies that "rings are q-reduced" in our language:

▶ **Lemma 65.** *The following is derivable from the rewrite rules of* GAG_q*:*



(35)

Together with the finite field Nullstellensatz, lemma 65 is sufficient for proving completeness (See Appendix C in the full version of this paper for details in proof):

▶ **Proposition 66.** *The semantics map* $[-]_q$ *is faithful. In other words, the rules of* GAG_q *are complete.*

▶ **Theorem 67.** *The semantics map* $[-]_q$ *is an isomorphism of PROPs.*

6.3 Matrix Semantics for GAG

Given a structured span of affine varieties over a field k (not necessarily algebraically closed) of the form $k^n \xleftarrow{\varphi} V \xrightarrow{\psi} k^m$ we can "repackage" the data of this structured span as an indexed family of affine varieties

$$V_{\mathbf{xy}} := \varphi^{-1}(\mathbf{x}) \cap \psi^{-1}(\mathbf{y}) \tag{36}$$

each of which is a subvariety of V.

In the particular case where $k = \mathbb{F}_q$ is a finite field, and V is an $\mathbb{F}_q$-rational locus, the family of affine varieties $V_{\mathbf{xy}}$ consists of finite sets, which we can view as entries in an $\mathbb{N}$-matrix:

▶ **Definition 68.** *Let* $M_q : {}_{G_q}\,\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)}) \to \mathrm{Mat}_{\mathbb{N}}$ *be the functor which sends a structured span of* $\mathbb{F}_q$*-rational loci*

$$\mathbb{F}_q^n \xleftarrow{\varphi} V \xrightarrow{\psi} \mathbb{F}_q^m \tag{37}$$

to the $q^n \times q^m$*-matrix* $M_q(\xleftarrow{\varphi}\ \xrightarrow{\psi})$ *with entries*

$$M_q(\xleftarrow{\varphi}\ \xrightarrow{\psi})_{\mathbf{xy}} = |\varphi^{-1}(\mathbf{x}) \cap \psi^{-1}(\mathbf{y})|. \tag{38}$$

The following is immediate (proofs are given in Appendix C.2 of the full version of this paper):

▶ **Proposition 69.** *The functor M_q is well-defined and fully faithful.*

Since $\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)} \simeq \mathsf{FinSet}$, this is just a restriction of the equivalence, well-known in folklore, between $\mathrm{Span}(\mathsf{FinSet}) \simeq \mathrm{Mat}_{\mathbb{N}}$ in disguise.

But since we already know that GAG_q is a sound, universal, and complete language for ${}_{G_q}\mathrm{Span}(\mathsf{AffVar}_{\mathbb{F}_q}^{(\mathbb{F}_q)})$ from Section 6.2, we have:

▶ **Corollary 70.** *The diagrammatic language GAG_q serves as a sound, universal, and complete language for a full subcategory of $\mathrm{Mat}_{\mathbb{N}}$ consisting of the objects q^n with $n \in \mathbb{N}$.*

We use $[\![-]\!]_q : \mathsf{GAG}_q \to \mathrm{Mat}_{\mathbb{N}}$ to denote these new semantics.

7 From GAG to Counting CSP

We now briefly illustrate the direct connection between GAG and CSP. Recall that a constraint satisfaction problem over a domain D and a constraint language Γ (typically a set of finite arity relations over D) is a pair $\langle X, C\rangle$, where:

- X is a set of variables $x_1, \ldots, x_n$,
- $C = \{C_1, \ldots, C_m\}$ is a set of constraints, each of which is a pair $C_j = \langle s_j, g_j\rangle$ where $s_j \subset X$ is the scope of the constraint, and $g_j \in \Gamma$ is the content of the constraint.

An assignment $\sigma : X \to D$ is said to satisfy the CSP if for every j, the tuple $(\sigma(x_i))_{i\in s_j}$ is in the relation g_j. The problem of determining whether there exists an assignment σ that satisfies $\langle X, C\rangle$ is called the decision problem $\mathsf{CSP}(\Gamma)$; while the problem of counting the number of assignments that satisfy $\langle X, C\rangle$ is called the counting problem $\#\mathsf{CSP}(\Gamma)$.

Now suppose that our domain of values is a finite field $\mathbb{F}_q$, and our constraint language Γ consists of polynomial maps $g : \mathbb{F}_q^n \to \mathbb{F}_q$, read as the requirement $g = 0$. Of course, every finite-arity relation over $\mathbb{F}_q$ can be expressed as an affine variety, and therefore our Γ is just the set of all relations over D. The finite field GAG gives an easy representation of instances of $\#\mathsf{CSP}(\Gamma)$ in the following way:

▶ **Proposition 71.** *Let $\langle X, C\rangle$ be a CSP instance over the domain $\mathbb{F}_q$, with set of variables $X = \{x_1, \ldots, x_n\}$, and polynomial constraints $g_1, \ldots, g_m \in k[x_1, \ldots, x_n]$. Let I be the ideal generated by $g_1, \ldots, g_m$. Then the span semantics of the ideal gadget is*

$$\overset{[-]_q}{\longmapsto} \quad \left(\mathbb{F}_q^n \hookleftarrow V(I)^{\mathbb{F}_q} \hookrightarrow \mathbb{F}_q^n\right) \tag{39}$$

where, recall, $V(I)^{\mathbb{F}_q} := \{\mathbf{x} \in \mathbb{F}_q^n : g_1(\mathbf{x}) = \cdots = g_j(\mathbf{x}) = 0\}$. In other words, it is the set of assignments satisfying $\langle X, C\rangle$.

The same diagram composed with no wires on either side has the following matrix semantic:

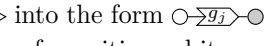
$$\overset{[\![-]\!]_q}{\longmapsto} \quad \left|V(I)^{(\mathbb{F}_q)}\right| \tag{40}$$

In other words, it is the number of assignments that satisfy $\langle X, C\rangle$.

▶ **Theorem 72.** *The problem of rewriting arbitrary GAG diagrams into each other is #P-hard.*

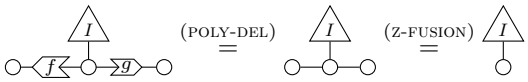
Proof. This is a direct consequence of equation (40) in proposition 71.

We can fix a family of polynomials $\{g_j\}_{j\in\mathbb{N}}$ such that each g_j is a polynomial of some number n_j of variables with exactly j roots in $\mathbb{F}_q^{n_j}$.

The problem #CSP(Γ) reduces to rewriting a given diagram  into the form  for some $j \in \mathbb{N}$. Since #CSP(Γ) is #P-complete [43], so is the problem of rewriting arbitrary GAG diagrams into each other. ◀

▶ Remark 73. In the case $q = 2$, polynomial constraints are equivalent to boolean constraints, and so #CSP(Γ) is #SAT.

▶ Remark 74. Conversely, since every diagram in GAG can be rewritten into cospan form, so every *closed* diagram (a diagram without input or output wires) can be rewritten into:



(41)

So determining the semantics of an arbitrary *closed* diagram in GAG is equivalent to #CSP(Γ).

8 From GAG to Qudit ZH Calculus

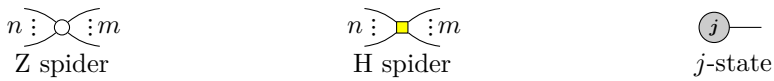
The ZH calculus is a graphical language for quantum computation, and it has a natural correspondence with the Toffoli-Hadamard gate set. In this section, we elucidate the connection between algebraic geometry and the ZH calculus. We review the Galois-Qudit ZH calculus introduced in [30] and show that the ZH calculus can be obtained as a fairly minimal extension of finite-field GAG: adding a single state and one scalar suffices to recover the ZH calculus in arbitrary dimension.

8.1 Qudit ZH Calculus

Here we give a brief overview of the qudit ZH Calculus as introduced in [30].

▶ **Definition 75.** *[47] A Galois-Qudit is a quantum system represented by a Hilbert space of dimension $q = p^t$ for some prime p and some natural number t, with a computational basis labelled as $|j\rangle$ with $j \in \mathbb{F}_q$.*

For a finite field $\mathbb{F}_q$, the **Galois-Qudit ZH Calculus** over $\mathbb{F}_q$ is the free[2] dagger-PROP generated by three families of generators:



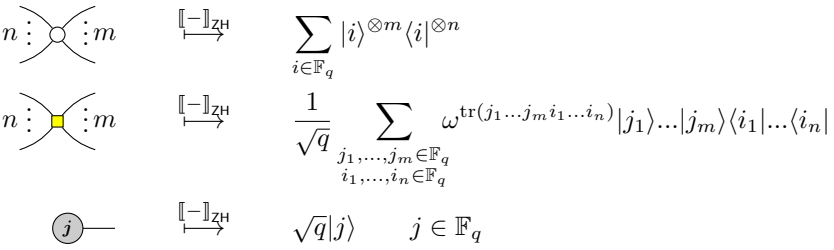
We will denote this PROP as ZH. It admits a semantics functor to the PROP $q^{\frac{*}{2}}\,\mathrm{Mat}_{\mathbb{Z}[\omega]}$:

$$\mathsf{ZH} \xrightarrow{[\![-]\!]_{\mathsf{ZH}}} q^{\frac{*}{2}}\,\mathrm{Mat}_{\mathbb{Z}[\omega]} \tag{42}$$

A morphism from n to m in $q^{\frac{*}{2}}\,\mathrm{Mat}_{\mathbb{Z}[\omega]}$ is a matrix of the form $q^{\frac{k}{2}}A$, where A is a $q^m \times q^n$ matrix with entries in the ring $\mathbb{Z}[\omega]$, ω is a primitive pth root of unity, and k is an integer.

[2] No complete set of rewrite rules is known for the qudit ZH. For the sake of full generality, we work with the free dagger-PROP generated by the ZH generators.

This functor is full [30] and is defined on the generators as follows:



where $\mathrm{tr} = \mathrm{tr}_{q/p}$ is the field trace function.

8.2 ZH Calculus as an Extension of GAG

We now extend the language GAG_q for finite field GAG to cover the entirety of the Galois-Qudit ZH Calculus over $\mathbb{F}_q$.

▶ **Definition 76.** *The dagger-PROP* $\mathsf{GAG}_q^{\mathsf{Fourier}}$ *is generated by* GAG_q *together with the following additional generators.*

Here, *is a state in the Fourier basis as we define below.*

▶ **Definition 77.** *We define the semantics functor* $[\![-]\!]_q^{\mathsf{Fourier}} : \mathsf{GAG}_q^{\mathsf{Fourier}} \to q^{\frac{*}{2}}\,\mathrm{Mat}_{\mathbb{Z}[\omega]}$. *For each generator of* GAG_q, *we set* $[\![-]\!]_q^{\mathsf{Fourier}}$ *to be the composition of* $[\![-]\!]_q$ *with the canonical inclusion* $\mathrm{Mat}_{\mathbb{N}} \hookrightarrow q^{\frac{*}{2}}\,\mathrm{Mat}_{\mathbb{Z}[\omega]}$. *For the additional generators, we define:*

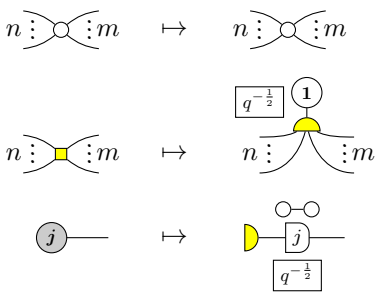
$$\left[\!\!\left[\text{(1)}\!\!- \right]\!\!\right]_q^{\mathsf{Fourier}} = \sum_{i\in\mathbb{F}_q} \omega^{\mathrm{tr}(i)}|i\rangle \qquad \left[\!\!\left[\boxed{q^{-\frac{1}{2}}} \right]\!\!\right]_q^{\mathsf{Fourier}} = q^{-\frac{1}{2}} \tag{43}$$

Hence, the following commutes:

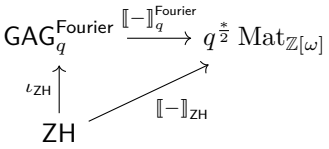
$$\begin{array}{ccc} \mathsf{GAG}_q^{\mathsf{Fourier}} & \xrightarrow{[\![-]\!]_q^{\mathsf{Fourier}}} & q^{\frac{*}{2}}\,\mathrm{Mat}_{\mathbb{Z}[\omega]} \\ \uparrow & & \uparrow \\ \mathsf{GAG}_q & \xrightarrow[[\![-]\!]_q]{} & \mathrm{Mat}_{\mathbb{N}} \end{array}$$

The dagger-PROP $\mathsf{GAG}_q^{\mathsf{Fourier}}$ covers the entirety of the Galois-Qudit ZH Calculus. To see this, we describe how its generators can be constructed in $\mathsf{GAG}_q^{\mathsf{Fourier}}$ and show that the semantics agree.

▶ **Definition 78.** *We have the functor* $\iota_{\mathsf{ZH}} : \mathsf{ZH} \to \mathsf{GAG}_q^{\mathsf{Fourier}}$ *defined on the generators as follows:*



▶ **Proposition 79.** *The functor ι_{ZH} is consistent with the semantics of the ZH Calculus. In other words, the following diagram commutes:*



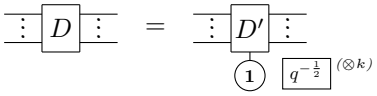
We will now show that by adding two rules to $\mathsf{GAG}_q^{\mathsf{Fourier}}$, we can simulate any amplitude computable in the ZH calculus using the rewrite rules of GAG and a constant overhead. Consider the following rewrite rules.

(44)

▶ **Proposition 80.** *The rewrite rules in equation* (44) *are sound with respect to the semantics* $[\![-]\!]_q^{\mathsf{Fourier}}$.

By working in the quotient $\mathsf{GAG}_q^{\mathsf{Fourier}}/_{(44)}$, we can reduce any diagram in $\mathsf{GAG}_q^{\mathsf{Fourier}}/_{(44)}$ to a diagram in GAG_q and a single $\boxed{1}$—, up to a finite scalar.

▶ **Proposition 81.** *For any diagram D in $\mathsf{GAG}_q^{\mathsf{Fourier}}/_{(44)}$, there exists $k \in \mathbb{N}$ and a diagram D' containing only the generators of GAG_q such that*



(45)

Proof. The idea to pull all the occurrences of $\boxed{1}$— to the bottom of the diagram, and then apply rule (Z1-COPY) repeatedly to merge all the occurrences of $\boxed{1}$— into one. The detailed proof is in Appendix E in the full version of this paper. ◀

This shows that just a single copy of the state in the Fourier basis is sufficient to extend the expressive power of finite-field GAG to cover the entirety of the Galois-Qudit ZH Calculus. Using this, we can compute amplitudes in the ZH calculus using finite-field GAG. Computing an amplitude in the ZH calculus corresponds to evaluating the semantics of a scalar diagram. We have the following result.

▶ **Theorem 82.** *Let D be a scalar diagram in the ZH calculus over $\mathbb{F}_q$. The scalar $[\![D]\!]_{\mathsf{ZH}}$ can be computed using exactly q queries to an oracle $\mathcal{O}$ that evaluates the semantics of scalar diagrams in GAG_q.*

9 Conclusion and Future Directions

We have introduced a family of diagrammatic languages, which we refer to collectively as graphical algebraic geometry (GAG). This family of languages extends the existing literature on graphical linear algebra (GLA) by introducing the nonlinear multiplication operator. We construct three examples of languages within this family, and show that they are sound, universal, and complete, for their respective semantics in (co)spans of commutative algebras and affine varieties.

We showed that closed diagrams in GAG correspond exactly to counting constraint satisfaction problems, thereby showing that rewriting GAG diagrams into each other is #P-hard. We show also that GAG forms the nonlinear classical "backbone" of the Galois-qudit ZH calculus. Indeed, we show that all quantum processes in the qudit ZH calculus can be represented by consuming a single Fourier basis state, up to a finite number of scalars, meaning that a GAG oracle would be able to simulate the ZH processes with a constant number of queries. These connections serve to illustrate the potential versatility of GAG.

There are several directions of future work. On the theoretic side, there is an open question of constructing graphical algebraic geometry over $\mathbb{R}$. Indeed, for a real-closed field, one would be able to express inequality constraints of the form $g(x) \geq 0$ using the generators of GAG; but finding a complete set of rewrite rules for such languages may prove difficult, and will involve the Positivstellensatze. Such a language would have exciting applications in the verification of hybrid systems, where the safety space is often given as a semialgebraic set in $\mathbb{R}^n$, and dynamics are given by polynomial mappings [62]. On the practical side, there are two obvious directions. The first is to explore the connection between GAG and CSP: in particular, are their existing computational techniques that can optimize GAG rewriting, or bound the complexity of rewriting between diagrams of a certain class? The second is to use the characterization of ZH as an extension of GAG to "grok" quantum algebraic geometry codes, in the same way graphical linear algebra has been used to "grok" CSS codes [48].

References

1 Amir Ali Ahmadi, Georgina Hall, Ameesh Makadia, and Vikas Sindhwani. Geometry of 3D Environments and Sum of Squares Polynomials, 2017. arXiv:1611.07369 [math]. doi:10.48550/arXiv.1611.07369.

2 A. Ashikhmin, S. Litsyn, and M. A. Tsfasman. Asymptotically Good Quantum Codes. *Physical Review A*, 63(3):032311, 2001. arXiv:quant-ph/0006061. doi:10.1103/PhysRevA.63.032311.

3 Miriam Backens and Aleks Kissinger. ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity. *Electronic Proceedings in Theoretical Computer Science*, 287:23–42, January 2019. doi:10.4204/EPTCS.287.2.

4 Miriam Backens, Aleks Kissinger, Hector Miller-Bakewell, John van de Wetering, and Sal Wolffs. Completeness of the ZH-calculus. *Compositionality*, 5:5, 2023. arXiv:2103.06610 [quant-ph]. doi:10.32408/compositionality-5-5.

5 John C. Baez and Kenny Courser. Structured Cospans, 2020. arXiv:1911.04630 [math]. doi:10.48550/arXiv.1911.04630.

6 John C. Baez and Jason Erbele. Categories in Control, 2015. arXiv:1405.6881 [math]. doi:10.48550/arXiv.1405.6881.

7 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity, 2018. Publication info: Preprint. MINOR revision. URL: https://eprint.iacr.org/2018/046.

8 Elwyn R. Berlekamp. *Algebraic coding theory*. McGraw-Hill series in systems science. McGraw-Hill, New York, 1st ed edition, 1968.

9 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String Diagram Rewrite Theory I: Rewriting with Frobenius Structure. *J. ACM*, 69(2):14:1–14:58, 2022. doi:10.1145/3502719.

10 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String diagram rewrite theory II: Rewriting with symmetric monoidal structure. *Mathematical Structures in Computer Science*, 32(4):511–541, 2022. doi:10.1017/S0960129522000317.

11 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. String Diagram Rewrite Theory III: Confluence with and without Frobenius, 2022. arXiv:2109.06049 [cs]. doi:10.48550/arXiv.2109.06049.

12 Filippo Bonchi, Robin Piedeleu, Pawel Sobocinski, and Fabio Zanasi. Graphical Affine Algebra. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, Vancouver, BC, Canada, 2019. IEEE. doi:10.1109/LICS.2019.8785877.

13 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. Interacting Bialgebras Are Frobenius. In *Foundations of Software Science and Computation Structures*, pages 351–365. Springer, Berlin, Heidelberg, 2014. ISSN: 1611-3349. doi:10.1007/978-3-642-54830-7-23.

14 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. Full Abstraction for Signal Flow Graphs. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '15, pages 515–526, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2676726.2676993.

15 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. Interacting Hopf Algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, 2017. arXiv:1403.7048 [cs]. doi:10.1016/j.jpaa.2016.06.002.

16 Eugenia Cheng. Distributive laws for Lawvere theories. *Compositionality*, 2:1, 2020. arXiv:1112.3076 [math]. doi:10.32408/compositionality-2-1.

17 Bob Coecke and Ross Duncan. Interacting Quantum Observables. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 298–310, Berlin, Heidelberg, 2008. Springer. doi:10.1007/978-3-540-70583-3_25.

18 Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 1 edition, 2017. doi:10.1017/9781316219317.

19 Valerie Coffman, Joydip Kundu, and William K. Wootters. Distributed Entanglement, 1999. arXiv:quant-ph/9907047. doi:10.1103/PhysRevA.61.052306.

20 Cole Comfort. Distributive Laws, Spans and the ZX-Calculus, 2021. arXiv:2102.04386 [math]. doi:10.48550/arXiv.2102.04386.

21 Cole Comfort. The ZX&-calculus: A complete graphical calculus for classical circuits using spiders. *Electronic Proceedings in Theoretical Computer Science*, 340:60–90, 2021. arXiv:2004.05287 [cs]. doi:10.4204/EPTCS.340.4.

22 David Cox, John Little, and Donal OSHEA. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer Nature, New York, NY, third edition edition, 2007. ISSN: 0172-6056. doi:10.1007/978-0-387-35651-8.

23 David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-16721-3.

24 Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery. *Quantum*, 4:218, 2020. doi:10.22331/q-2020-01-09-218.

25 David Eisenbud. *Commutative Algebra: with a View Toward Algebraic Geometry*. Graduate Texts in Mathematics, 150. Springer New York, New York, NY, 1st ed. 1995. edition, 1995. doi:10.1007/978-1-4612-5350-1.

26 Brendan Fong, Paolo Rapisarda, and Paweł Sobociński. A categorical approach to open and interconnected dynamical systems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 495–504, 2016. arXiv:1510.05076 [eess]. doi:10.1145/2933575.2934556.

27 Markus Frembs. An algebraic characterisation of Kochen-Specker contextuality, 2024. arXiv:2408.16764 [quant-ph]. doi:10.48550/arXiv.2408.16764.

28 Steven D. Galbraith. *The mathematics of public key cryptography*. Cambridge University Press, Cambridge ;, 1st ed. edition, 2012.

29 Carlos Galindo and Fernando Hernando. Quantum codes from affine variety codes and their subfield-subcodes. *Designs, Codes and Cryptography*, 76(1):89–100, 2015. arXiv:1403.4060 [cs]. doi:10.1007/s10623-014-0016-8.

30 Dichuan Gao. The Qudit ZH Calculus for Arbitrary Finite Fields: Universality and Application, 2024. arXiv:2406.02219 [quant-ph]. doi:10.48550/arXiv.2406.02219.

31 Masoud Gharahi. Classifying Entanglement by Algebraic Geometry. *International Journal of Quantum Information*, 22(03):2350047, 2024. arXiv:2408.12265 [quant-ph]. doi:10.1142/S0219749923500478.

32 Sudhir Ghorpade. A note on Nullstellensatz over finite fields. In Shrikrishna Dani, Surender Jain, Jugal Verma, and Meenakshi Wasadikar, editors, *Contemporary Mathematics*, volume 738, pages 23–32. American Mathematical Society, Providence, Rhode Island, 2019. doi:10.1090/conm/738/14876.

33 D. G. Glynn and J. W. P. Hirschfeld. On the classification of geometric codes by polynomial functions. *Designs, Codes and Cryptography*, 6(3):189–204, 1995. doi:10.1007/BF01388474.

34 V. D. Goppa. Codes on algebraic curves. *Dokl. Akad. Nauk SSSR*, 259(6):1289–1290, 1981.

35 Jens Groth. On the Size of Pairing-Based Non-interactive Arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 305–326, Berlin, Heidelberg, 2016. Springer. doi:10.1007/978-3-662-49896-5_11.

36 A. Grothendieck. *Eléments de géométrie algébrique*. Publications mathématiques ; no. 4, 8, 11, 17, 20, 24, 28, 32. Presses Universitaires de Frane, Paris, 1960.

37 V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999. doi:10.1109/18.782097.

38 Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UNITED KINGDOM, 2004. URL: http://ebookcentral.proquest.com/lib/oxford/detail.action?docID=256634.

39 Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1977. doi:10.1007/978-1-4757-3849-0.

40 Helmut Hasse. Zur Theorie der abstrakten elliptischen Funktionenkörper I. Die Struktur der Gruppe der Divisorenklassen endlicher Ordnung. *Journal für die reine und angewandte Mathematik*, 1936(175):55–62, 1936. doi:10.1515/crll.1936.175.55.

41 David Hilbert. Ueber die Theorie der algebraischen Formen. *Mathematische Annalen*, 36(4):473–534, December 1890. doi:10.1007/BF01208503.

42 T. Hoholdt and R. Pellikaan. On the decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 41(6):1589–1614, 1995. doi:10.1109/18.476214.

43 Mark Jerrum. Counting Constraint Satisfaction Problems. In Andrei Krokhin and Stanislav Zivny, editors, *Dagstuhl Follow-Ups*, pages 27 pages, 647864 bytes. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. Artwork Size: 27 pages, 647864 bytes. doi:10.4230/DFU.VOL7.15301.205.

44 Lingfei Jin. Quantum Stabilizer Codes from Maximal Curves. *IEEE Transactions on Information Theory*, 60(1):313–316, 2014. arXiv:1311.2705 [cs]. doi:10.1109/TIT.2013.2287694.

45 Lingfei Jin and Chaoping Xing. Euclidean and Hermitian Self-orthogonal Algebraic Geometry Codes and Their Application to Quantum Codes, 2013. arXiv:1308.3575 [cs]. doi:10.48550/arXiv.1308.3575.

46 P. T. Johnstone. *Stone spaces*. Cambridge studies in advanced mathematics ; v. 3. University Press, Cambridge, 1982.

47 Avanti Ketkar, Andreas Klappenecker, Santosh Kumar, and Pradeep Kiran Sarvepalli. Nonbinary stabilizer codes over finite fields, 2005. arXiv:quant-ph/0508070. doi:10.48550/arXiv.quant-ph/0508070.

48 Aleks Kissinger. Phase-free ZX diagrams are CSS codes (...or how to graphically grok the surface code), 2022. arXiv:2204.14038 [quant-ph]. doi:10.48550/arXiv.2204.14038.

49 Aleks Kissinger and John van de Wetering. Reducing T-count with the ZX-calculus. *Physical Review A*, 102(2):022406, 2020. doi:10.1103/PhysRevA.102.022406.

50 Alexander Kissinger and John van de Wetering. *Picturing Quantum Software*. Preprint, January 2024.

51 Tuomas Laakkonen, Konstantinos Meichanetzidis, and John van de Wetering. Picturing Counting Reductions with the ZH-Calculus. *Electronic Proceedings in Theoretical Computer Science*, 384:89–113, 2023. arXiv:2304.02524 [cs]. doi:10.4204/EPTCS.384.6.

52 Tuomas Laakkonen, Konstantinos Meichanetzidis, and John van de Wetering. A Graphical #SAT Algorithm for Formulae with Small Clause Density. *Electronic Proceedings in Theoretical Computer Science*, 406:137–161, 2024. arXiv:2212.08048 [cs]. doi:10.4204/EPTCS.406.7.

53 Stephen Lack. Composing PROPs. *Theory and Applications of Categories [electronic only]*, 13:147–163, 2004. URL: http://eudml.org/doc/124613.

54 Yves Lafont. Towards an algebraic theory of Boolean circuits. *Journal of Pure and Applied Algebra*, 184(2):257–310, 2003. doi:10.1016/S0022-4049(03)00069-0.

55 J. M. Landsberg and Laurent Manivel. On the ideals of secant varieties of Segre varieties, 2003. arXiv:math/0311388. doi:10.48550/arXiv.math/0311388.

56 Rudolf Lidl. *Finite fields*. Encyclopedia of mathematics and its applications ; volume 20. University Press, Cambridge, second edition. edition, 1997.

57 Saunders MacLane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71(1):40–106, 1965. Publisher: American Mathematical Society. URL: https://projecteuclid.org/journals/bulletin-of-the-american-mathematical-society/volume-71/issue-1/Categorical-algebra/bams/1183526392.full.

58 R. Matsumoto. Improvement of Ashikhmin-Litsyn-Tsfasman bound for quantum codes. *IEEE Transactions on Information Theory*, 48(7):2122–2124, 2002. doi:10.1109/TIT.2002.1013156.

59 Ryutaroh Matsumoto. Algebraic geometric construction of a quantum stabilizer code. *arXiv preprint*, 48(7):2122–2124, 2002. arXiv:0107129.

60 Akimasa Miyake. Classification of multipartite entangled states by multidimensional determinants. *Physical Review A*, 67(1):012108, 2003. arXiv:quant-ph/0206111. doi:10.1103/PhysRevA.67.012108.

61 Boldizsár Poór, Razin A. Shaikh, and Quanlong Wang. ZX-calculus is Complete for Finite-Dimensional Hilbert Spaces. *Electronic Proceedings in Theoretical Computer Science*, 426:127–158, August 2025. doi:10.4204/eptcs.426.5.

62 Stephen Prajna and Ali Jadbabaie. Safety Verification of Hybrid Systems Using Barrier Certificates. In Rajeev Alur and George J. Pappas, editors, *Hybrid Systems: Computation and Control*, pages 477–492, Berlin, Heidelberg, 2004. Springer. doi:10.1007/978-3-540-24743-2_32.

63 I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Society for Industrial and Applied Mathematics. Journal of the Society of Industrial and Applied Mathematics*, 8(2):5, June 1960. Num Pages: 5 Place: Philadelphia, United States Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/0108018.

64 Patrick Roy, John van de Wetering, and Lia Yeh. The Qudit ZH-Calculus: Generalised Toffoli+Hadamard and Universality. *Electronic Proceedings in Theoretical Computer Science*, 384:142–170, 2023. arXiv:2307.10095 [quant-ph]. doi:10.4204/EPTCS.384.9.

65 Pradeep Kiran Sarvepalli and Andreas Klappenecker. Nonbinary Quantum Codes from Hermitian Curves. In Marc P. C. Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 136–143, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11617983_13.

66 Shaska. Quantum codes from algebraic curves with automorphisms. *Condensed Matter Physics*, 11(2):383, 2008. doi:10.5488/CMP.11.2.383.

67 Edward Tsang. *Foundations of constraint satisfaction*. Computation in Cognitive Science. Academic Press Limited, London, England ;, 1993.

68 M. A. Tsfasman, S. G. Vlăduts, and Th. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, January 1982. doi:10.1002/mana.19821090103.

69 S. G. Vlâdut and V. G. Drinfel'd. Number of points of an algebraic curve. *Functional Analysis and Its Applications*, 17(1):53–54, January 1983. Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 1 Publisher: Kluwer Academic Publishers-Plenum Publishers. doi:10.1007/BF01083182.

70 André Weil. Numbers of solutions of equations in finite fields. *Bulletin (new series) of the American Mathematical Society*, 55(5):497–508, 1949. doi:10.1090/S0002-9904-1949-09219-4.

71 Adam Wills, Min-Hsiu Hsieh, and Hayata Yamasaki. Constant-Overhead Magic State Distillation, 2024. arXiv:2408.07764 [quant-ph]. doi:10.48550/arXiv.2408.07764.

72 Paul Wilson and Fabio Zanasi. An axiomatic approach to differentiation of polynomial circuits. *Journal of Logical and Algebraic Methods in Programming*, 135:100892, 2023. doi:10.1016/j.jlamp.2023.100892.

73 Fabio Zanasi. Interacting Hopf Algebras: the theory of linear systems, 2018. arXiv:1805.03032 [math]. doi:10.48550/arXiv.1805.03032.