CATEGORIES OF MATRIX FACTORIZATIONS AND DOLD-PUPPE EXTENSIONS OF FUNCTORS

Contents

1. Introduction/Goal of Project

These are notes for the project titled **Categories of Matrix Factorizations and Dold-Puppe Extensions of Functors** for the June 5th-9th Macaulay2 workshop at the University of Minnesota – Twin Cities. This project will have two parallel (but related) goals which are indicated in the title of the project: the first strand of this project is implementing the basic categorical structure on matrix factorizations, along with other tertiary commands for computing invariants of interest associated to matrix factorizations/$\mathbb{Z}/d\mathbb{Z}$-graded complexes. The second strand is to implement an algorithmic construction of what is often called the *Dold-Puppe* extension of a functor for certain functors in Macaulay2, with a side goal of laying the groundwork for computing in the so-called **simplicial category**.

I will give a slightly more detailed treatment to each strand of this project later, but I wanted to first convince those that may not be familiar with each strand of this project that these commands are useful to the broader mathematical community and have serious applications in the context of commutative algebra, algebraic geometry, and algebraic topology.

1.1. **Matrix Factorizations.**

Matrix factorizations are surprisingly ubiquitous objects, first introduced by Eisenbud [Eis80] while studying the structure of free resolutions over complete intersection rings. They are intimately related to the structure of maximal Cohen-Macaulay (MCM) modules over hypersurface rings, but also can be equipped with a categorical structure that ends up giving a beautifully concrete description of two much more (seemingly abstract) categories: the stable category of MCM modules over a hypersurface ring, and the singularity category of coherent sheaves over a hypersurface.

More recently, matrix factorizations and their various generalizations have become widely studied objects for their connections to mathematical physics, particularly connections to **Landau-Ginzburg (LG)** models. I am not an expert on this connection, but maybe other people in our group know more about this and have ideas for computations related to this aspect of matrix factorizations.

While there does already exist some functionality related to matrix factorizations in Macaulay2 (see the `CompleteIntersectionResolutions` package; related are the packages `ModuleDeformations` and `SuperLinearAlgebra`; also the work [Mas18]), the perspective taken in these packages does not actually implement functionality for basic categorical operations between matrix factorizations. These packages are more interested in implementing machinery used to *construct* matrix factorizations, and I suspect that the work done here will only enhance the utility of these preceding packages by providing methods to explicitly manipulate these objects with much more ease. Moreover, these other packages do not consider methods of constructing $\mathbb{Z}/d\mathbb{Z}$-graded matrix factorizations, a generalization of the classical notion that has connections to Ulrich modules over hypersurfaces and other interesting categories (see [Tri22], [LT21]; also [Hop21]).

1.2. **Dold-Puppe Extensions of Functors.**

The **Dold-Kan** correspondence is a well-known tool to algebraic topologists that still seems to be relatively underutilized in commutative algebra. It is an equivalence of categories:

$$\begin{tikzcd} \mathrm{Ch}_{\geq 0}(R) \arrow[rr, bend left, "\Gamma"] & & \text{Simplicial } R-\text{modules} \arrow[ll, bend left, "N"] \end{tikzcd}$$

where $\mathrm{Ch}_{\geq 0}(R)$ denotes the category of nonnegative graded chain complexes. One important consequence of this equivalence is that it gives a canonical method of extending *any* endofunctor on the category of R-modules to the category of (bounded) chain complexes in a way that preserves homotopies. This allows one to define derived functors of *non-additive* functors (for instance, how would you define the left derived functor $\mathbb{L}S^d$, where S^d denotes the dth symmetric power?), which is extremely important in algebraic topology for defining higher loop spaces and other "higher" topological data. This method of extending a functor is often known as the **Dold-Puppe extension** of a functor, named after the work [DP61] which develops the theory of derived nonlinear functors using the Dold-Kan correspondence (see also the work of Gillet-Soulé [GS87], which rigorously constructs K-theoretic operations preserving support by using the Dold-Kan correspondence)

Potentially the above topological application is not too compelling to an algebraist, but I can also give a more relevant recent application. It is often desirable to restrict oneself to subcategories of the derived category of R-modules, for example, the category of perfect complexes with finite length homology. Some readers may know that there is already an alternative notion of the symmetric power of a complex that is pretty easy to define, but it turns out that this definition *does not preserve homotopies* in arbitrary characteristic. This means that the "classically" defined symmetric power functor for complexes does not restrict to a well-defined endofunctor on the category of perfect complexes with finite length homology. This issue can be completely side-stepped if one instead uses the Dold-Puppe extension of the symmetric power functor instead.

This insight was recently used to resolve the **Total Rank Conjecture** [VW23] for arbitrary algebras over a field $\mathbf{k}$ (the case for $\mathbf{k}$ having odd characteristic was proved by Walker [Wal17]), and the main tool for getting around this issue of not having a well-defined restriction is resolved by instead using Dold-Puppe extensions of functors. The proofs of [VW23] use these Dold-Puppe extensions rather crudely, since in general these objects are extremely hard to compute by hand. One of the goals of this project is to combine the tools developed here along with the improved functionality of the `Complexes` package to develop a deeper understanding of the induced connecting homomorphisms between these Dold-Puppe extensions of functors, and potentially lead to a proof the the **Generalized Total Rank Conjecture** over rings of characteristic 2.

2. Mathematical Background: Matrix Factorizations

I'll give just a brief review of classical matrix factorizations and the more recent notion of $\mathbb{Z}/d\mathbb{Z}$-graded matrix factorizations (that is, factorizations with more factors). The purpose of this section is not to give comprehensive background, but to just give the most basic definitions to make sure we are all on the same page. Luckily, there are multiple people scheduled to be in this group that have background on matrix factorizations from a variety of different perspectives; my hope is that these people will also be able to tell us more about why they care about matrix factorizations and some of the mathematics related to matrix factorizations that they would like to see implemented.

2.1. Overview and Some Motivation.

Throughout this section S will denote a regular local ring or a (standard graded) polynomial ring over a field $\mathbf{k}$. All free S-modules are assumed to be finite rank.

Definition 2.1. A ($\mathbb{Z}/2\mathbb{Z}$-graded) **matrix factorization** of a ring element $f \in S$ is a factorization of the form

$$\begin{tikzcd} F_1 \arrow[r, bend left, "d_1"] & F_0 \arrow[l, bend left, "d_0"] \end{tikzcd}$$

where F_1 and F_0 are free S-modules of the same rank, and there are equalities

$$d_0 \circ d_1 = f \operatorname{id}_{F_1}, \quad d_1 \circ d_2 = f \operatorname{id}_{F_0}.$$

More generally, a $\mathbb{Z}/d\mathbb{Z}$-graded matrix factorization is a sequence of free S-modules of the same rank and maps

$$\begin{tikzcd} F_{d-1} \arrow[r, "d_{d-1}"] & F_{d-2} \arrow[r] & \cdots \arrow[r] & F_1 \arrow[r, "d_1"] & F_0 \arrow[llll, bend left, "d_0"] \end{tikzcd}$$

satisfying $d_1 \circ d_2 \circ \cdots \circ d_{d-1} \circ d_0 = f \operatorname{id}_{F_0}$.

Example 2.2. Let $S = \mathbf{k}[x, y, u, v]$ and $f := xy - uv$. Let $F_0 = F_1 = S^2$ and

$$d_0 := \begin{pmatrix} y & -u \\ -v & x \end{pmatrix},$$

$$d_1 := \begin{pmatrix} x & u \\ v & y \end{pmatrix}.$$

Then one can check that the data

$$\begin{tikzcd} F_1 \arrow[r, bend right, "d_1"'] & F_0 \arrow[l, bend right, "d_0"'] \end{tikzcd}$$

defines a $\mathbb{Z}/2\mathbb{Z}$-graded matrix factorization of f.

Example 2.3. Here is an interesting example of a larger matrix factorization: let $\mathbf{k} = \mathbb{C}$, $S := \mathbf{k}[x, y]$, and ω denote a primitive 3rd root of unity. Then a $\mathbb{Z}/3\mathbb{Z}$-graded matrix factorization of $f := x^3 + y^4$ is given by setting $F_0 = F_1 = F_2 = S^3$, and

$$d_1 = \begin{pmatrix} y & 0 & x \\ x & y^2 & 0 \\ 0 & x & y \end{pmatrix},$$
$$d_2 = \begin{pmatrix} y^2 & 0 & \omega x \\ \omega x & y & 0 \\ 0 & \omega x & y \end{pmatrix},$$
$$d_0 = \begin{pmatrix} y & 0 & \omega^2 x \\ \omega^2 x & y & 0 \\ 0 & \omega^2 x & y^2 \end{pmatrix}.$$

Then one can verify that any triple composition of the above differentials (in order) yields a factorization of multiplication by f.

An interesting fact (see Lemma 1.2.4 of [Tri22]) is that if one only assumes that *any* d-fold composition of the differentials of a $\mathbb{Z}/d\mathbb{Z}$-graded matrix factorization must also be scalar multiplication by the element $f \in S$. This tells us that factorizations have a natural **shift functor** $[i]$. Moreover, we can mimic many standard operations between complexes to obtain new matrix factorizations:

Definition 2.4 (Categorical Structure on Factorizations)**.** Let F and G be two $\mathbb{Z}/d\mathbb{Z}$-graded factorizations of a fixed ring element $f \in S$. A **morphism of matrix factorizations** $\varphi : F \to G$ is defined in an identical manner to that of complexes: it is a sequence of maps $\varphi_i : F_i \to G_i$ making the following diagram commute:

$$\begin{tikzcd} F_0 \arrow[r, "d_0^F"] \arrow[d, "\varphi_0"] & F_{d-1} \arrow[r, "d_{d-1}^F"] \arrow[d, "\varphi_{d-1}"] & F_{d-2} \arrow[r] \arrow[d, "\varphi_{d-2}"] & \cdots \arrow[r] & F_1 \arrow[r, "d_1^F"] \arrow[d, "\varphi_1"] & F_0 \arrow[d, "\varphi_0"] \\ G_0 \arrow[r, "d_0^G"'] & G_{d-1} \arrow[r, "d_{d-1}^G"'] & G_{d-2} \arrow[r] & \cdots \arrow[r] & G_1 \arrow[r, "d_1^G"'] & G_0 \end{tikzcd}$$

The class of all $\mathbb{Z}/d\mathbb{Z}$-graded factorizations of a fixed element $f \in S$ forms a category, often denoted $\mathrm{MF}_{\mathbb{Z}/d\mathbb{Z}}(f, S)$.

When $d = 2$, the category of $\mathbb{Z}/2\mathbb{Z}$-graded matrix factorizations is classically just referred to as matrix factorizations; this terminology arose in the seminal work of Eisenbud [Eis80]. One concrete application of matrix factorizations is that the minimal matrix factorizations provide an explicit description of all (non-free) maximal Cohen-Macaulay (MCM) modules over the hypersurface ring $R := S/(f)$:

Theorem 2.5. *There is an equivalence of categories*

$$\begin{tikzcd} \underline{\mathrm{MCM}}(R) \arrow[r, bend left, "\Theta"] & \mathrm{MF}(f, S) \arrow[l, bend left, "\Psi"] \end{tikzcd}$$

where $\underline{\text{MCM}}$ *denotes the stable category of MCM modules*[1].

The functor Ψ *is explicitly given on objects as follows: for any* ($\mathbb{Z}/2\mathbb{Z}$*-graded) matrix factorization* F*, define*

$$\Psi(F) := \operatorname{coker}(d_1).$$

The functor Θ *is explicitly given on objects as follows: given a MCM module* M*, it admits a free resolution of length* 1 *(by the Auslander-Buchsbaum formula) over* S*:*

$$F_1 \xrightarrow{d} F_0.$$

The map that multiplies by f *annihilates* M *and hence the scalar multiplication* $f : F \to F$ *induces a homotopy* $h : F_0 \to F_1$ *satisfying* $d \circ h = f\,\mathrm{id}_{F_1}$ *and* $h \circ d = f\,\mathrm{id}_{F_1}$*. The functor* Θ *is thus defined on objects as*

$$\Theta(M) := \qquad \begin{tikzcd} F_1 \arrow[r, bend left, "d"] & F_0 \arrow[l, bend left, "h"] \end{tikzcd}$$

2.2. **Some Constructions of Matrix Factorizations.** Before talking about examples, I wanted to say some words about tensor products between $\mathbb{Z}/d\mathbb{Z}$-graded factorizations for larger d:

Definition 2.6. Let F and G be $\mathbb{Z}/d\mathbb{Z}$-graded factorizations and let ω denote any primitive dth root of unity (if $d = 2$, then $\omega = -1$). The **tensor product** $F \otimes_S G$ is defined to be the $\mathbb{Z}/d\mathbb{Z}$-graded factorization with:

$$(F \otimes_S G)_t := \bigoplus_{r+s\equiv t \mod d} F_r \otimes_S G_s$$

$$= F_0 \otimes_S G_{i+d \mod d} \oplus F_1 \otimes_S G_{i+d-1 \mod d} \oplus \cdots \oplus F_r \otimes_S G_{i+d-r \mod d} \oplus \cdots \oplus F_i \otimes_S G_0,$$

for $i = 0, \ldots, d-1$, with differential:

$$d^{F\otimes G}(f_r \otimes g_s) := d^F(f_r) \otimes g_s + \omega^r f_r \otimes d^G(g_s), \quad \text{for } f_r \in F_r,\ g_s \in G_s.$$

Proposition 2.7. *If* F *and* G *are* $\mathbb{Z}/d\mathbb{Z}$*-graded factorizations of elements* f *and* g*, respectively, then the tensor product* $F \otimes_S G$ *is a* $\mathbb{Z}/d\mathbb{Z}$*-graded factorization of* $f + g$*.*

Remark 2.8. The **Hom** factorization is defined in an identical way to the tensor product, where we swap out tensors with Homs as so:

$$F_r \otimes G_s \longleftrightarrow \operatorname{Hom}_S(F_r, G_s), \quad \text{with differential}$$
$$d^{\operatorname{Hom}(F,G)}(\varphi) := d^F \circ \varphi - \omega^{|\varphi|}\phi \circ d^G,$$

where $|\cdot|$ denotes the degree.

Example 2.9 (**The Koszul Factorization**). Let $S = \mathbf{k}[x_1, \ldots, x_n]$ be any polynomial ring over a field and write

$$f = \sum_{i=1}^{n} x_i \cdot y_i,$$

where $y_i \in S$. Then for each $i = 1, \ldots, n$ there is the "trivial" matrix factorization

$$F^i : \qquad \begin{tikzcd} S \arrow[r, bend right, "y_i"'] & S \arrow[l, bend right, "x_i"'] \end{tikzcd}$$

[1]it is not important if you don't know exactly what this is, but informally this category takes all the MCM modules and quotients out by the "trivial" ones

The tensor product

$$\bigotimes_{i=1}^{n} F^i$$

is a matrix factorization of f, and is often referred to as the **Koszul factorization.**

Example 2.10. More generally, let $f = \sum_i f_i$ be any homogeneous polynomial of degree d, where each $f_i = c_i x_{i,1} \cdot x_{i,2} \cdots x_{i,d}$ is a monomial ($c_i \in \mathbf{k}$ is a constant). Each of the f_i's admits the "trivial" $\mathbb{Z}/d\mathbb{Z}$-graded factorization

$$F^i : \qquad \begin{tikzcd} S^1 \arrow[r, "c_i x_{i,1}"] & S^1 \arrow[r, "x_{i,2}"] & \cdots \arrow[r, "x_{i,d-1}"] & S^1 \arrow[lll, bend left, "x_{i,d}"] \end{tikzcd}$$

The tensor product $\bigotimes_i F^i$ is thus a $\mathbb{Z}/d\mathbb{Z}$-graded factorization of f, and by construction each of the matrices in this factorization are linear. Taking the cokernel of any of these maps (then tensoring with $S/(f)$) yields an Ulrich module over $S/(f)$; this is essentially the construction of Ulrich modules given by [HUB91].

Example 2.11. The following construction is due to recent work of Eisenbud-Schreyer and may be used to construct very small rank matrix factorizations, based on the work [ES22].

3. Mathematical Background: Dold-Puppe Extensions of Functors

This section is going to be a crash-course in simplicial R-modules and Dold-Puppe extensions of functors, along with a disillation of the algorithm of [SK10] (see also [Köc01]) that is more easily digestible to an algebraist. It is not necessary to fully understand the theoretical underpinnings of this material, so many things will be swept under the rug!

3.1. Overview and Motivation: The Dold Kan Correspondence.

In this section we recall details regarding the **Dold-Kan** correspondence. This correspondence gives a method for extending (not necessarily additive) functors defined at the level of R-modules to functors of complexes, in such a way that homotopy equivalences are preserved. The full details of this correspondence will be left as a sort of black box, but we will make explicit its essential properties.

Definition 3.1. The **simplicial category** Δ is the category whose objects are the finite, non-empty ordered sets $[n] := \{0 < 1 < \cdots < n\}$ for $n \geq 0$, with morphisms

$$\mathrm{Hom}_\Delta([n],[m]) := \{\text{nondecreasing set maps } [n] \to [m]\}.$$

Given a category $\mathcal{C}$, a **simplicial object** in $\mathcal{C}$ is a contravariant functor from Δ to $\mathcal{C}$. The class of all simplicial objects in $\mathcal{C}$ forms a category $\mathrm{s}\,\mathcal{C}$ with morphisms given by natural transformations.

Assume $\mathcal{C}$ is an idempotent complete, exact category; i.e., a full subcategory of an abelian category that is closed under extensions and summands. The example we have in mind occurs when $\mathcal{C}$ is the category of finitely generated and projective R-modules, for a commutative ring R. Then $\mathrm{s}\,\mathcal{C}$ is also an exact category, with the notion of exactness given component-wise. Let $\mathrm{Ch}_{\geq 0}(\mathcal{C})$ denote the category of non-negative chain complexes of objects in $\mathcal{C}$ with morphisms being chain maps; it is also an exact category with exactness defined degreewise.

Theorem 3.2 (Dold-Kan Correspondence, [Dol58], [Lur16, Theorem 1.2.3.7])**.** *For $\mathcal{C}$ as above, there is an equivalence of categories*

$$\begin{tikzcd} \mathrm{s}\,\mathcal{C} \arrow[r, bend left, "N"] \arrow[r, phantom, "\cong"] & \mathrm{Ch}_{\geq 0}(\mathcal{C}). \arrow[l, bend left, "\Gamma"] \end{tikzcd}$$

Moreover,

- N *converts simplicial homotopies to chain homotopies,*
- Γ *converts chain homotopies to simplical homotopies,*
- N *and* Γ *are exact functors, and*
- N *and* Γ *are natural for additive functors* $\mathcal{C} \to \mathcal{C}'$.

The functor N is the **normalization** functor, which associates to a simplicial object a chain complex whose degree n piece is the intersection of the kernels of the first n face maps. For a complex

$$P = (\cdots \xrightarrow{\partial_2} P_1 \xrightarrow{\partial_1} P_0 \to 0), \tag{3.1}$$

the degree n component of the simplicial object $\Gamma(P)$ is $\bigoplus_{[n]\twoheadrightarrow[k]} P_k$ where the direct sum is indexed by surjections in Δ with source $[n]$. For instance $\Gamma(P)_0 = P_0$, $\Gamma(P)_1 = P_0 \oplus P_1$, the degeneracy $\Gamma(P)_0 \to \Gamma(P)_1$ is the evident inclusion and the two face maps $\Gamma(P)_1 \to \Gamma(P)_0$ are $d_0(p_0, p_1) = p_0$ and $d_1(p_0, p_1) = p_0 + \partial_1(p_1)$.

Definition 3.3. Let $\mathcal{C}$ be an idempotent complete additive category. Given a covariant (not necessarily additive) endofunctor $G : \mathcal{C} \to \mathcal{C}$ define $G_* : \mathrm{s}\mathcal{C} \to \mathrm{s}\mathcal{C}$ be the functor given by composition: G_* sends a simplicial object $M : \Delta^{\mathrm{op}} \to \mathcal{C}$ to the simplicial object $G \circ M : \Delta^{\mathrm{op}} \to \mathcal{C}$. The **Dold-Puppe extension** of G is the functor $\widetilde{G} : \mathrm{Ch}_{\geq 0}(\mathcal{C}) \to \mathrm{Ch}_{\geq 0}(\mathcal{C})$ given by

$$\widetilde{G} := N \circ G_* \circ \Gamma.$$

Example 3.4. Even for classically defined functors on complexes, the Dold-Kan extension is in general a different construction. For example, consider the second symmetric power functor applied to a length 1 complex $C_1 \to C_0$. One can compute both versions of these functors as so:

$$S^2(C): \quad \begin{array}{c} \bigwedge^2 C_1 \\ \downarrow \\ C_1 \otimes C_0 \\ \downarrow \\ S^2(C_0) \end{array} \qquad\qquad \widetilde{S^2}(C): \quad \begin{array}{c} C_1 \otimes C_1 \\ \downarrow \\ C_1 \otimes C_0 \oplus S^2(C_1) \\ \downarrow \\ S^2(C_0) \end{array}$$

More generally, given a complex P as in (3.1) and functor G, the complex $\tilde{G}(P)$ has $\tilde{G}(P)_0 = G(P_0)$ and $\tilde{G}(P_1) = \ker(G(P_0 \oplus P_1) \xrightarrow{G(d_0)} G(P_0))$ and the differential $\tilde{G}(P)_1 \to \tilde{G}(P)_0$ is the restriction of $G(d_1)$.

Corollary 3.5. *The Dold-Puppe extension of any functor preserves chain homotopies.*

Definition 3.6. An endo-functor $G : \mathcal{C} \to \mathcal{C}$ is called *polynomial* if $G(0) = 0$ and, for some $d \geq 0$, the s-th cross effects functor $T_s(G) : \mathcal{C}^{\oplus s} \to \mathcal{B}$ is zero for all $s > d$ (see [EM54, §9]). The minimum value of d for which this holds is the *degree* of G.

A functor is additive if and only if it is polynomial of degree at most 1 [EM54, 9.11]. (A functor has degree 0 if and only if it is the zero functor.) An example of a non-additive polynomial functor is the second exterior power functor Λ^2_R defined on the category of finitely generated, projective R-modules (see below). We have a natural isomorphism

$$\Lambda^2_R(P \oplus P') \cong \Lambda^2_R(P) \oplus P \otimes_R P' \oplus \Lambda^2_R(P'), \tag{3.2}$$

which shows that the second cross effects functor T_2 associated to Λ^2_R is given by $T_2(P, P') = P \otimes_R P'$. Since T_2 is additive in each argument, we have $T_s = 0$ for all $s > 2$ [EM54, 9.9]. The same holds for the second symmetric power functor S^2_R and the second tensor power functor T^2_R; see below.

The following is given by [DP61, 4.7 and Hilfssatz 4.23].

Proposition 3.7. *If G is a polynomial functor of degree d and $P \in \mathrm{Ch}_{\geq 0}(\mathcal{C})$ is concentrated in degrees $[0, n]$; i.e., $P = (\cdots \to 0 \to P_n \xrightarrow{d_n} \cdots \xrightarrow{d_2} P_1 \xrightarrow{d_1} P_0 \to 0)$, then the complex $\widetilde{G}(P)$ is concentrated in degrees $[0, dn]$. If G is additive, then there is a natural isomorphism*

$$\widetilde{G}(P) \cong G(P) := (\cdots \to 0 \to G(P_n) \xrightarrow{G(d_n)} \cdots \xrightarrow{G(d_2)} G(P_1) \xrightarrow{G(d_1)} G(P_0) \to 0).$$

We now focus on our primary case of interest: Let R be a commutative ring and $\mathcal{P}(R)$ the category of finitely generated and projective R-modules.

Definition 3.8. An endo-functor $G : \mathcal{P}(R) \to \mathcal{P}(R)$ *commutes with localization* if for each prime ideal $\mathfrak{p}$ of R there exists an endofunctor $G_{\mathfrak{p}} : \mathcal{P}(R_{\mathfrak{p}}) \to \mathcal{P}(R_{\mathfrak{p}})$ and a natural isomorphism joining the functors $P \mapsto G(P)_{\mathfrak{p}}$ and $P \mapsto G_{\mathfrak{p}}(P_{\mathfrak{p}})$.

Observe that if G commutes with localization then the naturality of the Dold-Kan correspondence gives a natural isomorphism

$$\widetilde{G_{\mathfrak{p}}}(P_{\mathfrak{p}}) \cong \widetilde{G}(P)_{\mathfrak{p}} \tag{3.3}$$

in $\mathrm{Ch}_{\geq 0}(\mathcal{P}(R_{\mathfrak{p}}))$ for each $P \in \mathrm{Ch}_{\geq 0}(\mathcal{P}(R))$.

Recall that the *support* of a complex P of R-modules is

$$\mathrm{Supp}_R(P) = \{\mathfrak{p} \in \mathrm{Spec}(R) \mid \ P_{\mathfrak{p}} \text{ is not exact}\}.$$

If P is a bounded below complex of projective R-modules, then $\mathfrak{p} \in \mathrm{Supp}_R(P)$ if and only if $P_{\mathfrak{p}}$ is not contractible as a complex of $R_{\mathfrak{p}}$-modules.

Proposition 3.9. *If $G : \mathcal{P}(R) \to \mathcal{P}(R)$ commutes with localization, then $\mathrm{Supp}_R(\widetilde{G}(P)) \subseteq \mathrm{Supp}_R(P)$ for all $P \in \mathrm{Ch}_{\geq 0}(\mathcal{P}(R))$.*

We write $\mathrm{Perf}_{\geq 0}(R)$ and $\mathrm{Perf}^{\mathsf{fl}}_{\geq 0}(R)$ for the full subcategories of $\mathrm{Perf}(R)$ and $\mathrm{Perf}^{\mathsf{fl}}(R)$ consisting of complexes with $P_j = 0$ for $j < 0$. By Proposition 3.7, given any polynomial endofunctor G of $\mathcal{P}(R)$, its Dold-Kan extension determines an endofunctor $\widetilde{G}$ of $\mathrm{Perf}_{\geq 0}(R)$ that preserves chain homotopies and hence quasi-isomorphisms. If G commutes with localization, then $\widetilde{G}$ restricts to an endofunctor of $\mathrm{Perf}^{\mathsf{fl}}_{\geq 0}(R)$.

Remark 3.10. The work of Tchernev-Weyman [TW04] provides an alternative method of extending polynomial functors to arbitrary complexes of free R-modules. Their construction is distinct from the Dold-Kan extension in general (see [TW04, Example 14.7]).

3.2. **An Algorithmic Construction.** Given an endofunctor $F : \mathcal{C} \to \mathcal{C}$ of a (well-behaved) category as defined in the previous section, computing the image of the Dold-Puppe extension is generally quite difficult computationally. The work of Köck-Satkuranath [SK10] gives an algorithmic method of constructing these Dold-Puppe extensions, and the steps of the algorithm may be enumerated as follows:

(1) First enumerate a set of matrices $A^{n,k}$ specified by integers n and k falling within a certain range.
(2) Second, use these matrices to compute the degeneracy maps σ_i applied to the image $\Gamma(C)$ by using the matrix $A^{n,k}$.
(3) Third, use the matrices $A^{n,k}$ and $A^{n-1,k}$ to compute the face maps d_i applied to the image $\Gamma(C)$.
(4) Compute a distinguished set of morphisms coming from the simplicial category known as **minimal honorable surjections**.
(5) Use these minimal honorable surjections to enumerate all (not necessarily minimal) honorable surjections.

$(n,k)=(1,1)$	0	1
$1,(1,1)$	2	2

$(n,k)=(2,1)$	0	1	2
$1,(1,2)$	2	0	1
$2,(2,1)$	0	1	2

$(n,k)=(2,2)$	0	1	2
$1,(1,1,1)$	2	2	2

(6) Use the data of the face/degeneracy maps along with the data of the honorable surjections to compute the terms and maps of the resulting complex $N \circ F \circ \Gamma(C)$, where F is the functor of interest.

If using the honorable surjections to compute the induced differentials ends up being too difficult, an alternative approach is to only implement steps $(1)-(3)$ and then take the quotient by the image of the degeneracy maps, equipped with the differential given by the alternating sum of the face maps.

Alternatively, we can create a type `SimplicialModule` and implement the normalization functor in an efficient way. After specifying the simplicial objects using steps $(1)-(3)$, we could then just take the normalization to obtain the associated Dold-Puppe extension. We will see what seems most plausible as the project progresses.

3.3. **Constructing the Matrices Encoding Simplicial Structure.**

The first step for understanding the algorithmic construction is to construct a family of matrices that encodes the data of the face and degeneracy maps associated to the image of a given complex C under the Dold-Kan functor.

These matrices $A^{n,k}$ are specified by *two* pieces of data:

- an integer $n \geq 1$, specifying the degree $\Gamma(C)_n$, and
- an integer $1 \leq k \leq n$.

The columns of $A^{n,k}$ are indexed by the set $\{0,\dots,n-1\}$, and the rows of $A^{n,k}$ are indexed by all compositions of the integer $n+1$ into precisely $k+1$ parts, ordered lexicographically (where a composition is a tuple of *positive* integers adding up to $n+1$).

The entries of the matrices $A^{n,k}$ are elements of the set $\{0,1,2\}$, and the following flow chart specifies how to assign a value to each entry:

$$\begin{tikzcd}
\begin{array}{c}\text{Choose tuple } (\mu, j),\\ \text{where } \mu=(\mu_1,\dots,\mu_r) \text{ is}\\ \text{a composition, } j\in\{0,\dots,n-1\}.\end{array} \arrow[d, "\text{Initiate}"] & & \\
\begin{array}{c}\text{Is } \mu_1+\cdots+\mu_i=j+1\\ \text{for any } i\leq r?\end{array} \arrow[r, "\text{Yes}"] \arrow[d, "\text{No}"] & \text{Is } \mu_i = 1? \arrow[r, "\text{Yes}"] \arrow[d, "\text{No}"] & \text{Set } A_{\mu,j}=2. \\
\text{Set } A_{\mu,j} := 0. & \text{Set } A_{\mu,j}=1. &
\end{tikzcd}$$

3.4. **Computing the Degeneracy Maps.**

The degeneracy maps are easier to compute than the face maps, so we will first outline how to use the matrices of the previous section to compute the degeneracy map applied to each of the direct summands of $\Gamma(C)$.

$(n,k)=(3,1)$	0	1	2	3
$1,(1,3)$	2	0	0	1
$2,(2,2)$	0	1	0	1
$3,(3,1)$	0	0	1	2

$(n,k)=(3,2)$	0	1	2	3
$1,(1,1,2)$	2	2	0	1
$2,(1,2,1)$	2	0	1	2
$3,(2,1,1)$	0	1	2	2

$(n,k)=(3,3)$	0	1	2	3
$1,(1,1,1,1)$	2	2	2	2

$(n,k)=(4,1)$	0	1	2	3	4
$1,(1,4)$	2	0	0	0	1
$2,(2,3)$	0	1	0	0	1
$3,(3,2)$	0	0	1	0	1
$4,(4,1)$	0	0	0	1	2

$$
\begin{tikzcd}
\begin{array}{c}\text{Fix } n \geq 1,\ i \in \{0,\ldots,n-1\} \\ k \in \{0,\ldots,n\},\ \text{and} \\ \underline{c} = (c_1,\ldots,c_{\binom{n-1}{k}}) \in \Gamma(C)_{n-1,k}\end{array} \arrow[d, "\text{Initiate}"'] & \\
\text{Choose } 1 \leq \ell \leq \binom{n}{k} : \arrow[d] & \\
\text{Is } A^k_{\ell,i} = 0? \arrow[r, "\text{Yes}"'] \arrow[d, "\text{No}"] & \text{Set } \sigma_i(\underline{c})_\ell := c_{\nu(\ell)} \arrow[ul, "\ell \mapsto \ell+1"'] \\
\text{Set } \sigma_i(\underline{c})_\ell := 0. \arrow[uu, bend left, "\ell \mapsto \ell+1"] &
\end{tikzcd}
$$

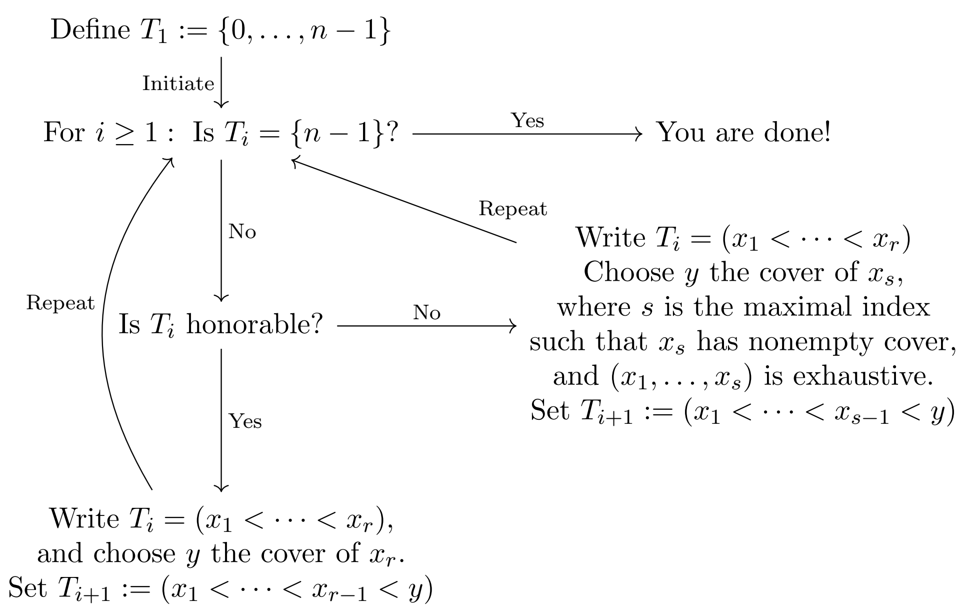
3.5. **Computing the Face Maps.** Now we will outline how to compute the face maps using the matrices computed previously. This is a more complicated process that involves multiple different cases, and the process is a little bit difficult to digest without examples, so I will work out a few examples in excruciating detail at the end.

$(n,k)=(4,2)$	0	1	2	3	4
$1,(1,1,3)$	2	2	0	0	1
$2,(1,2,2)$	2	0	1	0	1
$3,(1,3,1)$	2	0	0	1	2
$4,(2,1,2)$	0	1	2	0	1
$5,(2,2,1)$	0	1	0	1	2
$6,(3,1,1)$	0	0	1	2	2

$(n,k)=(4,3)$	0	1	2	3	4
$1,(1,1,1,2)$	2	2	2	0	1
$2,(1,1,2,1)$	2	2	0	1	2
$3,(1,2,1,1)$	2	0	1	2	2
$4,(2,1,1,1)$	0	1	2	2	2

$(n,k)=(4,4)$	0	1	2	3	4
$1,(1,1,1,1,1)$	2	2	2	2	2

3.6. **Computing The Terms of $NF\Gamma(C)$.**



3.7. **Computing the Differentials of $NF\Gamma(C)$.**

4. Wish List of Functions/Types: Matrix Factorizations

In this section, we will enumerate a wish list of the different commands that would be useful to have for handling matrix factorizations and some of the constructions related to these objects. I will define the objects to be implemented and also give an outline for a potential method of implementing each command (with other relevant commands that might be useful for doing this).

4.1. **Basic Categorical Constructions.** Our goal here is to make all of the commands below compatible with the pre-existing `Complexes` package, and likewise we want some back-and-forth with Michael Brown's `DifferentialModules` material.

Command 1. The type `ZZdFactorization`: a matrix factorization that is a $\mathbb{Z}/d\mathbb{Z}$-graded. All of the below commands in this section should be with respect to these types.

Command 2. The type `ZZdFactorizationMap`: a morphism of $\mathbb{Z}/d\mathbb{Z}$-graded factorizations.

Command 3. The type `ToricFactorization`: this should be a thing.

Command 4. The shift command `[-]`: This shifts a $\mathbb{Z}/d\mathbb{Z}$-graded factorization.

Command 5. The type `ZZdbiFactorization`: A bifactorization is a collection of bigraded modules equipped with two differentials: with respect to each differential, the factorization should be $\mathbb{Z}/d\mathbb{Z}$-graded and $\mathbb{Z}/e\mathbb{Z}$-graded, respectively.

Command 6. The command `totalize`: this should totalize a $\mathbb{Z}/d\mathbb{Z}$-graded bifactorization and output the totalized factorization.

Command 7. The command `cone`: this takes as input a morphism of factorizations and outputs the cone of that morphism of factorizations.

Command 8. The command `tensorBifactor`:

Command 9. The command `tensor`: takes the tensor product of two $\mathbb{Z}/d\mathbb{Z}$-graded factorizations and outputs another $\mathbb{Z}/d\mathbb{Z}$-graded factorization.

Command 10. The command `halfTensor`: takes as inputs two $\mathbb{Z}/d\mathbb{Z}$-graded factorizations and outputs 1/2 of the above tensor product. Only applies to rings where 2 is a unit.

Command 11. The command `homBifactor`: takes as inputs two $\mathbb{Z}/d\mathbb{Z}$-graded factorizations and outputs the $\mathbb{Z}/d\mathbb{Z}$-graded Hom bifactorization.

Command 12. The command `Hom`: takes as input two $\mathbb{Z}/d\mathbb{Z}$-graded factorizations and outputs the $\mathbb{Z}/d\mathbb{Z}$-graded Hom factorization.

Command 13. The shift command: takes as input a $\mathbb{Z}/d\mathbb{Z}$-graded factorization and outputs the shifted factorization

$$P[i]$$

whose tth graded piece is $P_{i+t \mod d}$, with defining endomorphism $(-1)^i d^P$.

Command 14. The command `HH`: this should compute the homology of a $\mathbb{Z}/d\mathbb{Z}$-graded complex (ie, a factorization of 0). Before computing, M2 should check whether the defining endomorphism squares to 0.

Command 15. The command `eulerCharacteristic`: this should only apply to $\mathbb{Z}/2\mathbb{Z}$-graded complexes P, and computes the integer $\ell(H_0(P)) - \ell(H_1(P))$, where ℓ denotes the length (it outputs infinity if the length is infinite).

Command 16. The command `fold`: this takes two inputs: an integer d and a complex P. The output should be the "folded" $\mathbb{Z}/d\mathbb{Z}$-graded object induced by the complex P.

Command 17. The command `trivialFactorization`: takes two inputs: an integer d and a ring element f and outputs the trivial factorization.

Command 18. The command `koszulFactorization`: takes as input a ring element an outputs the Koszul factorization.

Command 19. The command `randomKoszulFactorization`: takes as inputs a degree and a ring, and outputs the Koszul factorization of a randomly chosen polynomial.

Command 20. The command `eulerFactorization`: computes the Koszul factorization on the decomposition induced by the Euler identity.

Command 21. The command `BGSFactorization`: takes as input a ring element and outputs the conjecturally minimal rank matrix factorization of f.

4.2. **More Advanced Constructions.**

Command 22. The command `morphFromCycle`: given a cycle of the Hom complex, there is an induced morphism of $\mathbb{Z}/d\mathbb{Z}$-graded factorizations. This should turn a cycle into a morphism.

Command 23. The command `sym`: Takes two inputs: an integer n and a $\mathbb{Z}/d\mathbb{Z}$-graded factorization P, and outputs the nth symmetric power of P (classically defined).

Command 24. The command `wedge`: Takes two inputs: an integer n and a $\mathbb{Z}/d\mathbb{Z}$-graded factorization P, and outputs the nth exterior power of P (classically defined).

Command 25. The command `factorizationToMCM` (and vice versa): takes as input a matrix factorization and

Command 26. The command `stableExt`: computes the stable Ext.

5. Wish List of Functions/Types: Dold-Puppe Extensions

5.1. **The Basic Building Block Functions.**

Command 27. The command `sHom`: takes as inputs two integers n and k, outputs all order preserving maps $[n] \to [k]$.

Command 28. The command `surj`: takes as inputs two integers n and k, outputs all order preserving surjections $[n] \to [k]$ ordered by a total order on the Boolean poset coming from [SK10].

Command 29. The commands `deltaBar` and `sigmaBar`: (defined in [SK10])

Command 30. The command `epiMonic`: produces the epimonic factorization of a set surjection.

Command 31. The command `morphToPtn`: given a morphism $\mu : [n] \to [k]$, outputs the partition

$$\{\mu^{-1}(0), \mu^{-1}(1), \ldots, \mu^{-1}(k)\}.$$

Command 32. The command `ptnToMorph`: reverse of above.

Command 33. The command `Sset`: takes as inputs integers n, k, and i: outputs surjections as in Notation 1.10 of the paper.

Command 34. The command `Stilde`: outputs $\widetilde{S}$ as in the Notation 1.10.

Command 35. The command `deltaOp`: Makes clear the bijection of Definition 4.1.

Command 36. The command `minimalHonorables`: Takes an input an integer n and outputs the minimal honorable tuples.

Command 37. The command `honorables`: Takes as input an integer n and outputs the honorable tuples of the disjoint union

5.2. More Advanced Constructions.

Command 38. The command `nonSurjectFactorization`: constructs the factorization of Lemma 1.8.

Command 39. The command `ithFace`:

Command 40. The command `ithDegen`:

Command 41. The command `symCrossEffect`: Takes as input an integer consisting of two lists: the first is a tuple $\alpha_0, \dots, \alpha_n$ of integers and the second is a list of free R-modules $F_0, \dots, F_n$. The output is the cross effect functor

$$\mathrm{cr}_{|\alpha|}(F_\alpha).$$

Command 42. The command `galeBoolean`: Takes as input an integer n and outputs the Boolean poset on n equipped with the Gale ordering.

Command 43. The command `wedge2sym2Connecting`: Constructs the connecting homomorphism $H_i(\bigwedge^2 P) \to H_{i-1}(S^2(P))$.

5.3. Bonus: The Tchernev-Weyman Construction.

References

[Dol58] Albrecht Dold, *Homology of symmetric products and other functors of complexes*, Annals of Mathematics (1958), 54–80.

[DP61] Albrecht Dold and Dieter Puppe, *Homologie nicht-additiver funktoren. anwendungen*, Annales de l'institut fourier, 1961, pp. 201–312.

[Eis80] David Eisenbud, *Homological algebra on a complete intersection, with an application to group representations*, Transactions of the American Mathematical Society **260** (1980), no. 1, 35–64.

[EM54] Samuel Eilenberg and Saunders MacLane, *On the groups h (π, n), ii: methods of computation*, Annals of Mathematics (1954), 49–139.

[ES22] David Eisenbud and Frank-Olaf Schreyer, *Hyperelliptic curves and ulrich sheaves on the complete intersection of two quadrics*, arXiv preprint arXiv:2212.07227 (2022).

[GS87] Henri Gillet and Christophe Soulé, *Intersection theory using adams operations*, Inventiones mathematicae **90** (1987), no. 2, 243–277.

[Hop21] Eric Hopkins, *N-fold matrix factorizations*, Ph.D. Thesis, 2021.

[HUB91] Jurgen Herzog, Bernd Ulrich, and Jörgen Backelin, *Linear maximal cohen-macaulay modules over strict complete intersections*, Journal of Pure and Applied Algebra **71** (1991), no. 2-3, 187–202.

[Köc01] Bernhard Köck, *Computing the homology of koszul complexes*, Transactions of the American Mathematical Society **353** (2001), no. 8, 3115–3147.

[LT21] Graham J Leuschke and Tim Tribone, *Branched covers and matrix factorizations*, arXiv preprint arXiv:2110.02435 (2021).

[Lur16] Jacob Lurie, *Higher algebra (2017)*, Available at http://www. math. harvard. edu/˜ lurie (2016).

[Mas18] Matthew Mastroeni, *Matrix factorizations and singularity categories in codimension two*, Proceedings of the American Mathematical Society **146** (2018), no. 11, 4605–4617.

[SK10] Ramesh Satkurunath and Bernhard Köck, *An algorithmic approach to dold-puppe complexes* (2010).

[Tri22] Tim Tribone, *Matrix factorizations with more than two factors*, Ph.D. Thesis, 2022.

[TW04] Alexandre Tchernev and Jerzy Weyman, *Free resolutions for polynomial functors*, Journal of Algebra **271** (2004), no. 1, 22–64.

[VW23] Keller VandeBogert and Mark E Walker, *The total rank conjecture in characteristic two*, arXiv preprint arXiv:2305.09771 (2023).

[Wal17] Mark E Walker, *Total betti numbers of modules of finite projective dimension*, Annals of Mathematics (2017), 641–646.