D-modules on stacks

Merrick Cai

February 14, 2024

Contents

1 Overview

These notes were made in preparation of a talk for GL Support Group seminar, on February 14, 2024, on the topic "D-modules on stacks." I'll forgo certain technical details, which hopefully will increase the readability, but if you really want to know, it's mostly in [GR17]. Another helpful exposition is certain parts in [DG13].

Acknowledgements

I would like to thank Dylan Pentland for very insightful viewpoints, Taeuk Nam for pretty much teaching me the entirety of §2, and a HUGE thanks to Wyatt Reeves who basically taught me everything else.

2 Stacks

In this section, we briefly review basic notions of stacks.

2.1 Motivation

Sheaves are the abstract machinery which allows us to understand how local data patches together into global data. Not everything can be understood globally; indeed, many things need to first be understood locally, and then we attempt to glue it together. A prime example of this idea is the existence of partitions of unity on smooth manifolds. What partitions of unity allow us to do is to construct the existence of say, a bilinear form, locally - and then patch it together into a global bilinear form. This is especially useful in many cases because it allows us to essentially reduce many construction problems to working on (an open set of) $\mathbb{R}^n$, instead of an abstract manifold. Another example is vector bundles - what characterizes vector bundles isn't its global structure, but rather its *local* structure, which always look the same. What then differentiates one vector bundle from another is how the local pieces are patched together.

While extremely powerful constructions such as partitions of unity don't exist everywhere, the basic notion *is* ubiquitous, in the form of sheaves and presheaves. A presheaf on a topological space X assigns to every open set a collection of objects associated to that open set, along with maps indicating how these restrict when we move from an open set U to an open set $V \subset U$ (satisfying some basic compatibility rules). For example, suppose we wanted to record the data of smooth functions on a manifold X. Then we could declare a (pre)sheaf $\mathcal{O}_X^{an}$ which records the data of smooth functions *locally*: on an open set U, $\mathcal{O}_X^{sm}(U)$ consists of (the set, group, or ring of) all smooth functions on U. A function $f \in \mathcal{O}_X^{sm}(U)$ then restricts to a function on $V \subset U$ in a canonical way: just by restricting the domain of f to V. Therefore, restriction maps $U \supset V$ on the (pre)sheaf $\mathcal{O}_X^{sm}$ simply restrict the domain of a function from U to V.

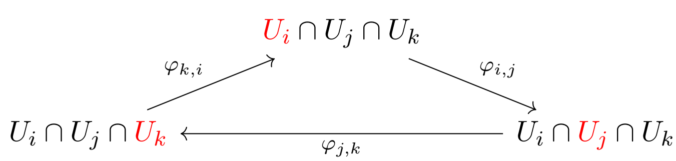
What separates sheaves from presheaves is the data of how to glue these functions together. In essence, a sheaf is a presheaf where the gluing is allowed. In the above example, $\mathcal{O}_X^{sm}$ has more than just the restriction maps: a function is determined entirely by its restrictions to an open cover. What this means is twofold: first, to check that two functions agree, it suffices to check that they agree on their restrictions to an open cover; and second, to specify a function, it suffices to specify functions on an open cover, such that they agree on overlaps (i.e., $f_i \in \mathcal{O}_X^{sm}(U_i)$ for $\{U_i\}_i$ covering U such that $f_i|_{U_i \cap U_j} = f_j|_{U_i \cap U_j}$). In particular, we know that *we can always glue together local functions into a global one*, given that there's nothing that causes the function to fail to actually be a function.

So a sheaf is basically the answer to: if we wanted to know about *every THING about a space*, where THING can be continuous functions, holomorphic functions, etc. - anything "clearly defined," then we define a sheaf which just assigns to each open set the THINGs defined over that set.

The goal of stacks is to copy the idea of sheaves, but to more difficult objects than functions. Just as how we wanted a sheaf to tell us about every THING on a space, we can now ask a stack to tell us every THING on a space where THING is no longer so clearly defined. Let us explain what "clearly defined" means. The key point is that functions are extremely rigid: there's no such thing as an "isomorphic" function. A function is simply determined by its values on every point. However, many things are only defined up to isomorphism. For example, to specify a (rank n) vector bundle $\mathcal{V}$ on a space X, it suffices to specify transition functions, i.e. isomorphisms

$$\varphi_{i,j} : \mathbb{C}^n \times (U_i \cap U_j) \xrightarrow{\sim} \mathbb{C}^n \times (U_i \cap U_j)$$

with $\varphi_{j,i} = \varphi_{i,j}^{-1}$. The crucial part is that these isomorphisms need to satisfy the compatibility relation called the triple intersection condition: on $U_i \cap U_j \cap U_k$, if we move in a circle



then we get the identity, i.e.

$$\varphi_{k,i} \circ \varphi_{j,k} \circ \varphi_{i,j} = \mathrm{id}_{U_i \cap U_j \cap U_k} .$$

The reason why this is necessary is because vector bundles are not defined uniquely. There are many different ways to construct isomorphic vector bundles. For example, in algebraic geometry, any two linearly equivalent divisors yield isomorphic line bundles, but notably *they are not actually the same sheaf*, as can be seen by looking at exactly which functions are defined over each open set. In many cases, there is no "one way" to specify an object - each object is defined only up to isomorphism. This leads us to the notion of a groupoid.

2.2 Groupoids

The basic idea is that sets are just a collection of objects, and each object is different from each other. For example, every continuous function on a space is distinct, because by definition they have to disagree on at least one point. What if objects were not so well-defined: like vector bundles? In this case, we can have two vector bundles which are not the same, but they *are* isomorphic. As in the construction of a vector bundle, we need to remember the local data *along with the isomorphisms* in order to check whether they can indeed glue together into a global vector bundle. This wasn't necessary for the case of functions because they were *literally* the same function. When we have isomorphic objects, then not only do we need to remember all of the objects, but also the isomorphisms between them (equivalently, this is when objects have automorphisms).

Definition 2.2.1. A **groupoid** is a category where every morphism is an isomorphism.

In essence, a groupoid is just remembering which objects are isomorphic, because it's no longer so clear as to just saying what vector bundle it is. The category of all groupoids is naturally a 2-category.

Definition 2.2.2. The 2-category Grpd consists of:

- the objects are groupoids,
- the 1-morphisms are functors (between groupoids),
- the 2-morphisms are natural transformations (of the functors).

2.3 From presheaves to prestacks

A presheaf can be written as a contravariant functor from the category of open sets of X, $\mathsf{Open}(X)$, to the category of sets Set. The category $\mathsf{Open}(X)$ is the category where the objects are open sets, and there is a single map from $V \to U$ iff $V \supset U$ (this includes the case $U = V$, whence the map is the identity map). This is just a fancy way of saying that a sheaf $\mathcal{F}$ assigns to every open set $U \subset X$, some set $\mathcal{F}(U)$ in Set, and every time we have a map $V \to U$ in $\mathsf{Open}(X)$ corresponding to $V \subset U$, then we have a corresponding map $\mathrm{res}_V^U : \mathcal{F}(U) \to \mathcal{F}(V)$.

Actually, we want to be a bit more general here. Let's work with the full category of schemes Sch; sometimes we will take this to mean the category of k-schemes, which we can denote by Sch_k. Then a presheaf is just a functor $\mathsf{Sch}^{op} \to \mathsf{Grpd}$. In other words,

$$\mathsf{PreSh}(\mathsf{Sch}) = \mathsf{Fun}(\mathsf{Sch}^{op}, \mathsf{Set}).$$

Definition 2.3.1. A **prestack** is a functor $\mathsf{Sch}^{op} \to \mathsf{Grpd}$.

In other words, a prestack $\mathfrak{X}$ just assigns to every scheme a groupoid, and every time we have a morphism of schemes $X \subset Y$, we have a functor of groupoids $\mathfrak{X}(Y) \to \mathfrak{X}(X)$, along with some compatibility conditions.

These compatibility conditions are actually quite important. Let's analyze it closely. Say we have a prestack $\mathfrak{X}$.

(0) On the level of objects, $\mathfrak{X}$ will send each scheme to a groupoid (remember, this is an entire category!).

(1) On the level of morphisms, for a map of schemes $A \to B$, $\mathfrak{X}$ will send this to a map of groupoids, i.e. a functor $\mathrm{res}^B_A : \mathfrak{X}(B) \to \mathfrak{X}(A)$.

(2) There is no nontrivial 2-category structure on our source Sch^{op}. So we sort of end here... **except** that item (1) doesn't actually make sense because although equality of morphisms makes sense in the 1-category Sch^{op}, it **doesn't** make sense in the target Grpd. Normally when we have a (contravariant) functor F of 1-categories, we expect $F(g \circ f) = F(f) \circ F(g)$, and the equality makes sense since we're living in a 1-category. Now, however, for morphisms of schemes $X \xrightarrow{f} Y \xrightarrow{g} Z$, we only know that $\mathfrak{X}(f) \circ \mathfrak{X}(g)$ is **naturally isomorphic** to $\mathfrak{X}(g \circ f)$. So additionally, the data of a prestack $\mathfrak{X}$ will include natural transformations $\alpha_{f,g} : \mathfrak{X}(f) \circ \mathfrak{X}(g) \xrightarrow{\sim} \mathfrak{X}(g \circ f)$. These natural isomorphisms must then play the "coherency" role in "making things functorial." What's the condition? Well, we have to go one step up: to check that these natural transformations are themselves compatible, we need to check that for every chain of maps

$$X \xrightarrow{f} Y \xrightarrow{g} Z \xrightarrow{h} W,$$

the induced diagram commutes:

$$\begin{array}{ccccc} & & \mathfrak{X}(f) \circ \mathfrak{X}(g) \circ \mathfrak{X}(h) & & \\ & \swarrow^{\alpha_{f,g} \times \mathrm{id}} & & \searrow^{\mathrm{id} \times \alpha_{g,h}} & \\ \mathfrak{X}(g \circ f) \circ \mathfrak{X}(h) & & & & \mathfrak{X}(f) \circ \mathfrak{X}(h \circ g) \\ & \searrow_{\alpha_{g \circ f, h}} & & \swarrow_{\alpha_{f, h \circ g}} & \\ & & \mathfrak{X}(h \circ g \circ f) & & \end{array}$$

By "commutes," we mean that the two compositions of natural transformations *should literally be equal*. This makes sense because they are 2-morphisms living in a 2-category; there's no higher structure, so we can genuinely say that they're equal.

The moral is basically that adding higher structure to the category means we can't say things are "equal," only "isomorphic," but these isomorphisms then need to be compatible on a higher level, which we check in this way until we reach the end of the structure (in which case we can genuinely say they're equal). However if we deal with infinity-categories then it never ends...

2.4 From sheaves to stacks

Mostly we've been working with the open affine subsets of a scheme. More generally, we want to work over some Grothendieck topology. There are several which are very popular:

$$\mathcal{S} = \mathsf{Zar} \subset \mathsf{\acute{E}t} \subset \mathsf{Sm} \subset \mathsf{fppf} \subset \mathsf{fpqc}.$$

They are: the Zariski site, the étale site, the smooth site, the fppf site, and the fpqc. The unifying idea is that to each object X, we require a collection of *covers*, each of which is a collection of maps (from arbitrary objects) to X, which are closed under certain operations. Basically this is a generalization of the data of an open cover in ordinary topology; people realized that it wasn't actually important that these things were *covers* in a literal sense, but moreso that they behaved in certain ways and obeyed certain rules.

In general, we always want to work with the category Sch of schemes; sometimes this will mean k-schemes for some field k. Endow it with any Grothendieck topology you want.

Definition 2.4.1. Choose a Grothendieck topology $\mathcal{S}$ on Sch^{op}. A **sheaf** $\mathcal{F}$ is a presheaf $\mathsf{Sch}_{\mathcal{S}}^{op} \to \mathsf{Set}$ satisfying 2 conditions: **uniqueness of gluing** and **existence of gluing**.

Let's spell this out carefully. Say we have a cover $U = \{U_i\}_{i \in I}$ of X. Make the notation $U_{ij} = U_i \cap U_j$, etc.

(1) **Uniqueness of gluing.** If we have two elements $x, y \in \mathcal{F}(X)$ such that $\operatorname{res}^X_{U_i} x = \operatorname{res}^X_{U_i} y \in \mathcal{F}(U_i)$ for all $i \in I$, then $x = y \in \mathcal{F}(X)$.

(2) **Existence of gluing.** Suppose we have $x_i \in \mathcal{F}(U_i)$ for all i such that $\operatorname{res}^{U_i}_{U_{ij}} x_i = \operatorname{res}^{U_j}_{U_{ij}} x_j \in \mathcal{F}(U_{ij})$ for all i, j. Then there exists $x \in \mathcal{F}(X)$ for which $\operatorname{res}^X_{U_i} x = x_i$ for all i.

This is alternatively captured by the following abstract notion. Consider U now as a single object (for example, take the disjoint unions of the U_i to form a single object). Then we have a commutative diagram

$$U \times_X U \rightrightarrows U \longrightarrow X.$$

From this, applying $\mathcal{F}$ we get

$$\begin{array}{ccccc} \mathcal{F}(X) & \longrightarrow & \mathcal{F}(U) & \rightrightarrows & \mathcal{F}(U \times_X U) \\ & \searrow & \uparrow & \nearrow & \\ & & \lim\left(\mathcal{F}(U) \rightrightarrows \mathcal{F}(U \times_X U)\right) & & \end{array}$$

as well as an induced map $\mathcal{F}(X)$ to the limit over the rest of the diagram. Now the first condition, uniqueness of gluing, is equivalent to this map being injective. The second condition, existence of gluing, is equivalent to this map being surjective. Thus $\mathcal{F}$ **is a sheaf iff this map is an isomorphism.**

Now let's move on to stacks. Once again,

Definition 2.4.2. A **stack** $\mathfrak{X}$ is a prestack $\mathsf{Sch}_S^{op} \to \mathsf{Grpd}$ satisfying the two conditions **uniqueness of gluing** and **existence of gluing**.

The only thing is, these conditions become a bit more involved. Let's spell it out concretely. Let $U = \{U_i\}_{i \in I}$ be a cover of X, and let $\mathfrak{X}$ be a prestack. For $\mathfrak{X}$ to be a stack, it needs to satisfy the following two conditions.

(1) **Uniqueness of gluing.** Suppose we have the data of $x, y \in \mathfrak{X}(X)$ (remember, $\mathfrak{X}(X)$ is a groupoid) and isomorphisms $\alpha_i : \operatorname{res}^X_{U_i} x \xrightarrow{\sim} \operatorname{res}^X_{U_i} y$ in (another groupoid) $\mathfrak{X}(U_i)$ for all i, such that $\operatorname{res}^{U_i}_{U_{ij}} \alpha_i = \operatorname{res}^{U_j}_{U_{ij}} \alpha_j$ for all i, j, as morphisms in the category $\mathfrak{X}(U_{ij})$ (this is a groupoid, hence an honest category, which is why we can ask that two morphisms are actually the same).

   Then there exists a unique isomorphism $\alpha : x \xrightarrow{\sim} y$ in $\mathfrak{X}(X)$ such that $\operatorname{res}^X_{U_i} \alpha = \alpha_i$ for all i.

(2) **Existence of gluing.** Suppose we have the data of $x_i \in \mathfrak{X}(U_i)$ for all i, along with isomorphisms $\beta_{ij} : \operatorname{res}^{U_i}_{U_{ij}} x_i \xrightarrow{\sim} \operatorname{res}^{U_j}_{U_{ij}} x_j$ in the groupoid $\mathfrak{X}(U_{ij})$, such that *they are compatible along triple intersections* in pretty much any way you restrict, i.e. the following diagram commutes:

$$
\begin{array}{ccccc}
 & \operatorname{res}^{U_{ij}}_{U_{ijk}} \circ \operatorname{res}^{U_i}_{U_{ij}} x_i & \xrightarrow{\operatorname{res}^{U_{ij}}_{U_{ijk}} \beta_{ij}} & \operatorname{res}^{U_{ij}}_{U_{ijk}} \circ \operatorname{res}^{U_j}_{U_{ij}} x_j & \\
 /\!\!/ & & & & \backslash\!\!\backslash \\
\operatorname{res}^{U_{ik}}_{U_{ijk}} \circ \operatorname{res}^{U_i}_{U_{ik}} x_i & & & & \operatorname{res}^{U_{jk}}_{U_{ijk}} \circ \operatorname{res}^{U_j}_{U_{jk}} x_k \\
 \searrow {\scriptstyle \operatorname{res}^{U_{ik}}_{U_{ijk}} \beta_{ik}} & & & & {\scriptstyle \operatorname{res}^{U_{jk}}_{U_{ijk}} \beta_{jk}} \swarrow \\
 & \operatorname{res}^{U_{ik}}_{U_{ijk}} \circ \operatorname{res}^{U_k}_{U_{ik}} x_k & = & \operatorname{res}^{U_{jk}}_{U_{ijk}} \circ \operatorname{res}^{U_k}_{U_{jk}} x_k &
\end{array}
$$

   Here, the double lines indicate canonical isomorphisms obtained from the data of $\mathfrak{X}$ as a prestack. Strictly speaking, we should actually expand the diagram to accommodate the isomorphisms encoded in these double lines, but we'll omit that for readability. The requirement that this diagram "commutes" is asking for an equality of morphisms either way you go around the diagram, which is fine since it just needs to be an equality of morphisms in the 1-category $\mathfrak{X}(U_{ijk})$.

   Assuming all this is satisfied, *then* there exists $x \in \mathfrak{X}(X)$ and isomorphisms $\gamma_i : \operatorname{res}^X_{U_i} x \xrightarrow{\sim} x_i$ for all i (remember that we cannot actually say that they're the same...) *such that* the γ_i are compatible (with each other, the restriction functors, the β_{ij}, etc...); in other words, for all i, j, the following diagram commutes:

$$\begin{array}{ccc}
\mathrm{res}^{U_i}_{U_{ij}} \circ \mathrm{res}^{X}_{U_i} x & \xrightarrow[\sim]{\mathrm{res}^{U_i}_{U_{ij}} \gamma_{ij}} & \mathrm{res}^{U_i}_{U_{ij}} x_i \\
\| & & \\
\mathrm{res}^{X}_{U_{ij}} x & & \sim \Big\downarrow \beta_{ij} \\
\| & & \\
\mathrm{res}^{U_j}_{U_{ij}} \circ \mathrm{res}^{X}_{U_j} x & \xrightarrow[\mathrm{res}^{U_i}_{U_{ij}} \gamma_j]{\sim} & \mathrm{res}^{U_j}_{U_{ij}} x_j
\end{array}$$

Ok, so we finally wrote down all of the conditions *concretely*. As you might imagine, this gets horrifyingly congested as the categories get more complicated. On the other hand, returning to the more abstract format, take U to be a single object again (for example, taking a disjoint union of the U_i)' we have a commutative diagram

$$U \times_X U \times_X U \Rrightarrow U \times_X U \rightrightarrows U \longrightarrow X$$

Applying $\mathfrak{X}$, we get

$$\begin{array}{ccccccc}
\mathfrak{X}(X) & \longrightarrow & \mathfrak{X}(U) & \rightrightarrows & \mathfrak{X}(U \times_X U) & \Rrightarrow & \mathfrak{X}(U \times_X U \times_X U) \\
 & \searrow & \nwarrow & & \uparrow & \nearrow & \\
 & & & \lim \Big(\mathfrak{X}(U) \rightrightarrows \mathfrak{X}(U \times_X U) \Rrightarrow \mathfrak{X}(U \times_X U \times_X U) \Big) & & &
\end{array}$$

There's once again an induced map $\mathfrak{X}(X)$ to the limit over the rest of the diagram. The uniqueness of gluing implies that this map is fully faithful, while the existence of gluing implies that this map is essentially surjective. (Note that "injective" and "surjective" don't actually mean anything on the level of groupoids, which are 1-categories!) To summarize, $\mathfrak{X}$ **is a stack iff this map is an equivalence of categories.**

Remark 2.4.3. Note that sets are naturally groupoids where we have no non-identity morphisms; concretely, every set can be stupidly made into a groupoid by having the objects be the elements of this set, and the only morphisms being the identity maps on each element. Then a scheme, which itself is a sheaf (with extra conditions such as being locally affine), is a map $\mathsf{Sch}^{op} \to \mathsf{Set} \subset \mathsf{Grpd}$. Since there 2-category structure is completely trivial if we stay within Set, the rest of the coherence conditions are essentially just trivial (every time we ask for an isomorphism in the conditions to be a stack, it's actually just equality as an element of the set). This means that **schemes are always stacks**.

Remark 2.4.4. Since any scheme $X \in \mathsf{Sch}_{\mathcal{S}}$ (for any of the Grothendieck topologies we considered) is locally affine (i.e. has a cover by affine schemes; this is true by definition in the Zariski site, and all other sites we consider are finer), actually every sheaf and stack is entirely determined by its values on $\mathsf{Aff}^{op} \subset \mathsf{Sch}^{op}$, the category of affine schemes. But since $\mathsf{Aff}^{op} = \mathsf{CommRing}$, the category of commutative rings, we can just regard stacks and sheaves to be functors from $\mathsf{CommRing}$ instead of from Sch^{op}. (If we want to work over a base field k, then we just get $\mathsf{CommAlg}_k$, the category of commutative k-algebras.) This is not true for prestacks and presheaves - we need descent properties so that we can consider only the smaller subcategory of affine schemes.

2.5 Algebraic spaces

Algebraic spaces are the first generalization of schemes.

Definition 2.5.1. A map $f : \mathfrak{X} \to \mathfrak{Z}$ of prestacks is **schematic** (i.e., representable by a scheme) if for all maps $S \to \mathfrak{Z}$ where S is a scheme, then $\mathfrak{X} \times_{\mathfrak{Z}} S$ is also a scheme.

Basically, this means that the base change of the map $\mathfrak{X} \to \mathfrak{Z}$ by a scheme (mapping to $\mathfrak{Z}$) still gives us a scheme:

$$\begin{array}{ccc} \text{still a scheme} & \longrightarrow & S\text{, a scheme} \\ \downarrow & \lrcorner & \downarrow \\ \mathfrak{X} & \xrightarrow{\quad f \quad} & \mathfrak{Z} \end{array}$$

Proposition 2.5.2. *Let X be an étale sheaf, i.e. a functor $\mathsf{Sch}^{op}_{\text{Ét}} \to \mathsf{Set}$ satisfying the sheaf axioms. The following are equivalent:*

(i) The diagonal map $\Delta : X \to X \times X$ is schematic.

(ii) For all maps $S_1 \to X$, $S_2 \to X$ where S_1, S_2 are schemes, then $S_1 \times_X S_2$ is still a scheme.

(iii) Every map from a scheme $S \to X$ is schematic.

Proof. (i) $\iff$ (ii) is essentially tautological, since a map to $X \times X$ is just the data of giving two maps $S_1, S_2 \to X$.

(ii) $\iff$ (iii) is similarly tautological, since $S \to X$ is schematic iff for all maps from another scheme $S' \to X$, the fiber product $S \times_X S'$ is a scheme. □

Definition 2.5.3. An **algebraic space** is an étale sheaf X on Sch (i.e. a functor $\mathsf{Sch}^{op}_{\text{Ét}} \to \mathsf{Set}$ satisfying the sheaf axioms) which satisfies the following two conditions:

(1) Any of the equivalent conditions in Proposition 2.5.2.

(2) X admits an étale cover by a scheme C, i.e., for any map $S \to X$ from a scheme S, then the fiber product $C \times_X S$ is a scheme, and furthermore the base change $C \times_X S \to S$ is an étale cover of schemes.

$$\begin{array}{ccc} C \times_X S & \xrightarrow{\text{étale cover}} & S \\ \downarrow & \lrcorner & \downarrow \\ C & \longrightarrow & X \end{array}$$

Remark 2.5.4. In a similar vein, we can ask for all sorts of nice (i.e., preserved by base change) properties coming from schemes to generalize to schematic maps. The condition is pretty much that the base change of this schematic map by a map from a scheme gives a map of schemes with the desired property.

Example 2.5.5. The functor $\mathbb{A}^1/\mathbb{Z}$ is an étale an algebraic space. The scheme $\mathbb{A}^1$ sends a commutative ring R to its underlying set of elements, also denoted by R. This also has the structure of an abelian group, hence has a canonical $\mathbb{Z}$-action, and the resulting quotient is an algebraic space.

2.6 Artin stacks

Definition 2.6.1. An **Artin stack**, or an **algebraic stack**, is an étale stack $\mathfrak{X}$ on schemes (i.e. a functor $\mathsf{Sch}^{op}_{\acute{\mathsf{E}}\mathsf{t}} \to \mathsf{Grpd}$ satisfying the stack conditions) satisfying the following conditions:

(1) The diagonal map $\Delta : \mathfrak{X} \to \mathfrak{X} \times \mathfrak{X}$ is representable by an algebraic space.
(2) $\mathfrak{X}$ admits a smooth cover by a scheme.

Remark 2.6.2. Actually we only require $\mathfrak{X}$ to have a smooth cover by an algebraic space, but algebraic spaces have a smooth cover by a scheme (by definition), so we might as well go straight to the cover by a scheme.

The smooth cover part is a bit of a technical definition, so we'll omit the details for now.

Artin stacks are the type of stack that we can (sometimes) have geometric notions, coming from the smooth cover by a scheme. For example, there's a notion of a tangent space at a point. A point of an Artin stack $\mathfrak{X}$ over a field k is just a map $\operatorname{Spec} k \to \mathfrak{X}$. Let's recall the usual definition of tangent space of a k-scheme Y at a point $p \in Y(k)$. Then we have a natural map $\pi : k[\varepsilon]/\varepsilon^2 \to k$ which induces a map $Y(k[\varepsilon]/\varepsilon^2) \to Y(k)$. The preimage of p in the set $Y(k[\varepsilon]/\varepsilon^2)$ is defined to be the tangent space of Y at p; this has a natural structure of a k-vector space. We can adopt this notion for Artin stacks. For an Artin stack $\mathfrak{X}$, we again have a natural map (really, functor) $\pi : \mathfrak{X}(k[\varepsilon]/\varepsilon^2) \to \mathfrak{X}(k)$, as maps of groupoids. A point in $\mathfrak{X}$ is just an element p of $\mathfrak{X}(k)$, and the preimage of this functor at p is defined to be the tangent space at that point p; this is a groupoid. It will turn out this is even a vector space groupoid, because the objects carry a vector space structure and the automorphisms of each object are also a vector space. Actually what happens is that the objects of this preimage form some vector space C^0, and the automorphisms of the identity object form some vector space C^{-1}, so that we have a complex $C^{-1} \xrightarrow{d} C^0$ such that $\operatorname{Hom}(c_1, c_2) = \{f \in C^{-1} \mid df = c_1 - c_2\}$. This gives us a way to describe the tangent space as a vector space with automorphisms on each object, where the automorphisms are also a vector space.

Using the notion of tangent space, we can give the notion of dimension. If a point on an Artin stack is smooth (which is a more technical definition), then we can just define the dimension of the Artin stack to be the Euler characteristic of the complex $C^{-1} \to C^0$, or $\dim H^0 - \dim H^{-1}$.

A better way to define dimension is via relative dimension: we just need to know that dimension behaves well under base change. Then to an Artin stack $\mathfrak{X}$, we can take a smooth cover by a scheme $U \to \mathfrak{X}$. Then for some map from a scheme $Y \to \mathfrak{X}$ such that the base change $U \times_{\mathfrak{X}} Y$ is a scheme, then the relative dimension $\dim U \times_{\mathfrak{X}} Y - \dim Y$ is equal to the relative dimension $\dim U - \dim \mathfrak{X}$; therefore, we can define $\dim X \coloneqq \dim U - \dim U \times_{\mathfrak{X}} Y + \dim Y$ (all of which are schemes, hence have a notion of dimension).

2.7 BG

Let us briefly describe BG, the classifying stack of principal G-bundles. Let G be a smooth affine algebraic group.

Let's first describe $* = \operatorname{Spec} k$. This is the Artin stack for which a map from a k-scheme S, i.e. $S \to *$, corresponds to a principal-$\{1\}$ bundle on S. But that's just S itself.

Definition 2.7.1. We define BG to be the functor sending a scheme S to the groupoid of principal G-bundles on S.

Another way to describe BG is as the quotient stack $*/G$. It is an Artin stack, with smooth cover $* = \operatorname{Spec} k$ and dimension $\dim BG = -\dim G$. So for a scheme S, a map $S \to BG$ is just a principal G-bundle over S. This clarifies the strange idea of negative dimension: to a map $S \to BG$ we associated a principal G-bundle $\mathcal{G} \to S$, which gives rise to the Cartesian square

$$\begin{array}{ccc} \mathcal{G} & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow \\ S & \longrightarrow & BG \end{array}$$

But the map $\mathcal{G} \to S$ has relative dimension $\dim G$, so the map $* \to BG$ should also have relative dimension $\dim G$... which forces $\dim BG = \dim \operatorname{Spec} k - \dim G = -\dim G$.

Remark 2.7.2. What we mean by a map $S \to BG$ is a map of *functors*; we treat both S and BG as functors $\mathsf{Sch}^{op} \to \mathsf{Grpd}$ by considering every set as a groupoid with no non-identity isomorphisms (equivalently, every object has no automorphisms besides the identity). But by 2-Yoneda, $\operatorname{Hom}(S, BG) = BG(S)$, so we can canonically identify maps $S \to BG$ with principal G-bundles over S.

Example 2.7.3. There is a unique map $* \to BG$. This corresponds to the trivial principal G-bundle G over $* = \operatorname{Spec} k$.

Example 2.7.4. Let's compute the pullback of the maps $* \to BG$:

$$\begin{array}{ccc} ? & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow \\ * & \longrightarrow & BG \end{array}$$

Let's consider maps V to the pullback. We have

$$\begin{tikzcd} V \arrow[dr] \arrow[drr, bend left] \arrow[ddr, bend right] & & \\ & ? \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & * \arrow[d] \\ & * \arrow[r] & BG \end{tikzcd}$$

But the map $* \to BG$ corresponds to the trivial G-bundle on $*$; this pulls back to the trivial G bundle $V \times G$ on V. Thus the two composites $V \to * \to BG$ both correspond to giving a trivial G-bundle on V such that the projection maps to V agree. In other words, a map $V \to ?$ is giving the data of an automorphism of $V \times G$ as principal G-bundles. But this is just maps $V \to G$, giving an automorphism of G (as a G-torsor) over each fiber. Thus we have that the pullback is G itself:

$$\begin{array}{ccc} G & \longrightarrow & * \\ \big\downarrow & \lrcorner & \big\downarrow \\ * & \longrightarrow & BG \end{array}$$

3 Many abstract notions

From here on in, **all categories and functors are derived**. For example, $\mathsf{QCoh}(X)$ means the derived category of quasicoherent sheaves on X.

3.1 Why derived categories?

Many things only become apparent at the level of derived categories. For example, the cotangent complex $\mathbb{L}_X$ of a scheme X is the "correct" generalization of the cotangent sheaf (i.e. sheaf of differentials), which often doesn't satisfy certain desired properties if X isn't smooth - for example, exactness of certain sequences. However, the cotangent complex does just that. For example, for maps $X \xrightarrow{f} Y \to Z$, then we have the distinguished triangle

$$f^*\mathbb{L}_{Y/Z} \to \mathbb{L}_{X/Z} \to \mathbb{L}_{X/Y}$$

in $\mathsf{QCoh}(X)$. Furthermore, for a Cartesian square

$$\begin{array}{ccc} W & \xrightarrow{f} & X \\ \big\downarrow & \lrcorner & \big\downarrow \\ Z & \longrightarrow & Y \end{array}$$

we have the (quasi-isomorphism)

$$\mathbb{L}_{W/Z} \simeq f^*\mathbb{L}_{X/Y}.$$

Armed with this, let's see what this implies about BG. It's know that

$$\mathsf{QCoh}(X/G) \simeq \mathsf{QCoh}(X)^G,$$

i.e. the (derived) category of quasicoherent sheaves on the quotient stack X/G is equivalent to the (derived) category of G-equivariant quasicoherent sheaves on X. Here, $BG = */G$, so $X = \operatorname{Spec} k$. In particular, we get that

$$\mathsf{QCoh}(BG) \simeq \mathsf{Rep}\, G.$$

But what is the cotangent complex? We can use the Cartesian square

$$\begin{array}{ccc} G & \xrightarrow{f} & * \\ \downarrow & \lrcorner & \downarrow g \\ * & \longrightarrow & BG \end{array}$$

to deduce that

$$\mathbb{L}_{G/*} = f^* \mathbb{L}_{*/BG}.$$

But G is a smooth affine algebraic variety over $\operatorname{Spec} k$, so $\mathbb{L}_{G/*}$ is just the ordinary cotangent sheaf (i.e. sheaf of differentials) of G over $\operatorname{Spec} k$, which is trivializable due to the G-action. Concretely, $\mathbb{L}_{G/*}$ is the constant sheaf $\underline{\mathfrak{g}^*}$ on G.

We also have the sequence of maps $* \xrightarrow{g} BG \to *$ which gives us the following distinguished triangle in $\mathsf{QCoh}(*) = \mathsf{Vec}_k$:

$$g^* \mathbb{L}_{BG/*} \to \mathbb{L}_{*/*} \to \mathbb{L}_{*/BG}.$$

But $\mathbb{L}_{*/*} = 0$, as this is the cotangent complex of $\operatorname{Spec} k$ over $\operatorname{Spec} k$; since $\operatorname{Spec} k$ is smooth and affine, this is the ordinary sheaf of differentials, which is zero. So we have that

$$\mathbb{L}_{*/BG} = g^* \mathbb{L}_{BG/*}[1].$$

Thus we have

$$\mathbb{L}_{G/*} \cong \underline{\mathfrak{g}^*} \cong f^* \mathbb{L}_{*/G} \cong f^* g^* \mathbb{L}_{BG/*}[1].$$

Since $f^* g^* = (g \circ f)^*$ is exact, this means that $\mathbb{L}_{BG/*}$ "only exists" in degree -1; so if we were to try to look at the cotangent "sheaf" of BG, we wouldn't see anything!

Explicitly, the equivalence $\mathsf{QCoh}(BG) \simeq \mathsf{Rep}$ identifies the cotangent complex $\mathbb{L}_{BG/*}$ with the coadjoint G-representation $\mathfrak{g}^*$, but shifted by $[-1]$.

The main takeaway from all this is that we do actually need the full power of derived categories to see the whole picture; otherwise, for example, the coadjoint representation in the (underived) category of G-representations wouldn't correspond to any quasicoherent sheaf on BG.

3.2 Finiteness conditions and IndCoh

In general, we want some sort of finiteness hypothesis, so that the objects we work with are more reasonable. For example, if X is a scheme which is not smooth, then there exist finitely generated quasicoherent sheaves $\mathcal{F}$ which don't have finite projective resolutions.

Example 3.2.1. Let $\operatorname{Spec} k[x]/x^2$. Then quasicoherent sheaves on X are just modules on $k[x]/x^2$. Since $k[x]/x^2$ is a local ring, projective modules are just free modules. One simple case is the module $M = k$, which does *not* have a finite free resolution; indeed, it has an infinite free resolution of the form

$$\cdots \xrightarrow{\cdot x} k[x]/x^2 \xrightarrow{\cdot x} k[x]/x^2 \xrightarrow{\cdot x} k[x]/x^2 \xrightarrow{\cdot x} k[x]/x^2 \to k.$$

There are many straightforward ways to see why k does not have a finite free resolution. One way is to compute that $\operatorname{Tor}_i^{k[x]/x^2}(k,k) = k$ for all $i \geq 0$, which implies that any free resolution of k must be infinite, else the Tor would become eventually zero. Another way is to just compute the Euler characteristic of a hypothetical finite free resolution. The free modules have even dimension, but k has odd dimension.

There's two main types of finiteness conditions that we can ask for on a quasicoherent sheaf (or more generally, a complex of quasicoherent sheaves).

(1) Define $\mathsf{Perf}(X)$ to be the complexes in $\mathsf{QCoh}(X)$ which are quasi-isomorphic to a bounded (i.e. finite) complex of finitely generated projectives.
(2) Define $\mathsf{Coh}(X)$ to be the complexes in $\mathsf{QCoh}(X)$ which are quasi-isomorphic to a bounded complex of finitely generated sheaves.

If X is smooth, then $\mathsf{Perf} = \mathsf{Coh}$, but in general, Coh is bigger. In certain unfortunate scenarios, Perf is not actually contained in Coh.

These "finiteness" conditions allow us to understand the simpler objects of a category.

Definition 3.2.2. An object c in a (filtered) category $\mathcal{C}$ is called **compact** if it commutes with all filtered colimits.

Compact objects are useful because they are the objects which are, as the name suggets, more "compact" or "finite." In the (underived) category Set, the compact objects are precisely the finite sets. In the (underived) category $R-\mathsf{mod}$, the compact objects are precisely the finitely presented R-modules. In the (unbounded, derived) category $\mathsf{QCoh}(X)$, the compact objects are precisely $\mathsf{Perf}(X)$.

Now, $\mathsf{Coh}(X)$ is (generally) bigger than $\mathsf{Perf}(X)$, but it's still a very reasonable and nice "finiteness" condition. What setting should we work in to consider $\mathsf{Coh}(X)$ as the compact objects? The answer is that we essentially need to construct it formally.

Definition 3.2.3. Let $\mathfrak{X}$ be a prestack (for example, a scheme). We construct the category $\mathsf{IndCoh}(\mathfrak{X})$ by "adjoining all (directed) colimits to $\mathsf{Coh}(\mathfrak{X})$." This gives us a category where $\mathsf{Coh}(\mathfrak{X})$ consists of the compact objects. More concretely, the objects are "formal colimits," denoted by $\widehat{\varinjlim} \mathcal{F}_i$ for $\mathcal{F}_i \in \mathsf{Coh}(\mathfrak{X})$, and the maps are given by

$$\mathrm{Hom}(\widehat{\varinjlim}_i \mathcal{F}_i, \widehat{\varinjlim}_j \mathcal{G}_j) = \varprojlim_i \mathrm{Hom}(\mathcal{F}_i, \widehat{\varinjlim} \mathcal{G}_j) = \varprojlim_i \varinjlim_j \mathrm{Hom}(\mathcal{F}_i, \mathcal{G}_j).$$

Note that $\mathsf{Coh}(\mathfrak{X})$ refers to the *derived* category, and so the Hom is the derived Hom!

Remark 3.2.4. The Ind construction is actually a more general categorical procedure, essentially given by adjoining all directed colimits. Thus, (in most cases) $\mathsf{IndCoh}(X) = \mathsf{Ind}(\mathsf{Coh}(X))$. Actually, this is not true in general, so the above definition is not correct, but we'll ignore that technicality and pretend our prestacks are nice.

When X is a smooth scheme, then $\mathsf{Perf}(X) = \mathsf{Coh}(X)$, so $\mathsf{QCoh}(X) = \mathsf{IndCoh}(X)$. In general, however, there's a functor

$$\mathsf{IndCoh}(X) \xrightarrow{\Psi_X} \mathsf{QCoh}(X),$$
$$\widehat{\varinjlim} \mathcal{F}_i \mapsto \varinjlim \mathcal{F}_i$$

sending the "formal colimit" to the actual colimit.

Proposition 3.2.5. *We have*

$$\mathsf{IndCoh}(X) = \varprojlim_{\mathrm{Spec}\, A \to \mathfrak{X}} \mathsf{IndCoh}(\mathrm{Spec}\, A).$$

Example 3.2.6. For X a scheme, the above proposition is just saying that $\mathsf{IndCoh}(X) = \mathsf{QCoh}(X)$ are determined by their sections on affine open covers $\mathcal{U}_i \to X$, such that these maps satisfy certain conditions such as triple intersections, etc.

Example 3.2.7. Let $X = \mathrm{Spec}\, R$ where $R = k[x]/x^2$ and $\mathcal{F}$ be the quasicoherent sheaf corresponding to the R-module k, as in Example 3.2.1. We have the projective (hence, free) resolution of k by

$$(\cdots R \to R \to R \to R) \xrightarrow{\text{quasi-iso}} k.$$

In $\mathsf{QCoh}(X)$, these two are identified, and this complex is actually the colimit of

$$\mathcal{F}_i := 0 \to \underbrace{R \to R \to \cdots \to R}_{n\,\text{times}} \to 0.$$

However, in $\mathsf{IndCoh}(X)$, we do *not* have k being equal to the "colimit" of the $\mathcal{F}_i$. Let us compute why. If we indeed had $k = \widehat{\varinjlim} \mathcal{F}_i$, then we would have $H^0(\mathrm{Hom}(k, \widehat{\varinjlim} \mathcal{F}_i)) = H^0(\mathrm{Hom}(k,k)) = k$. When we compute the hom-space, however, we find that

$$\begin{aligned}\mathrm{Hom}_{\mathsf{IndCoh}(X)}(k, \widehat{\varinjlim} \mathcal{F}_i) &= \varinjlim_{\mathsf{Vec}_k} \mathrm{Hom}_{\mathsf{Coh}(X)}(k, \mathcal{F}_i), \\ &= \varinjlim_{\mathsf{Vec}_k} \left(\mathrm{Hom}_{\mathsf{QCoh}(X)} \left(\cdots \to R \to R \to R \to 0, \quad 0 \to \underbrace{R \to \cdots \to R}_{i\,\text{times}} \to 0 \right) \right).\end{aligned}$$

But every map looks like

$$\begin{array}{ccccccccccccc} \cdots & \xrightarrow{\cdot x} & R & \xrightarrow{\cdot x} & R & \xrightarrow{\cdot x} & R & \xrightarrow{\cdot x} & R & \xrightarrow{\cdot x} & R & \longrightarrow & 0 \\ & & \downarrow{\scriptstyle 0} & & \downarrow{\scriptstyle \cdot\lambda_4 x} & & \downarrow{\scriptstyle \cdot\lambda_3 x} & & \downarrow{\scriptstyle \cdot\lambda_3 x} & & \downarrow{\scriptstyle \cdot\lambda_1 x} & & \\ & & 0 & \longrightarrow & R & \xrightarrow{\cdot x} & R & \xrightarrow{\cdot x} & R & \xrightarrow{\cdot x} & R & \longrightarrow & 0 \end{array}$$

since the left-most commutative square implies that the downwards map $xR \to R$ must be zero; hence each downwards map must be multiplication by some scalar multiple of x (so the $\lambda_i \in k$). But then the H^0 of any particular map here is given by the kernel of $R \xrightarrow{\cdot\lambda x} R$ modulo the image of $R \xrightarrow{\cdot x} R$, hence is 0. It follows that $H^0(\mathrm{Hom}(k, \widehat{\varinjlim} \mathcal{F}_i)) = 0$, which contradicts the fact that $H^0(\mathrm{Hom}(k,k)) = k$, hence $\widehat{\varinjlim} \mathcal{F}_i \neq k$ in $\mathsf{IndCoh}(X)$.

Remark 3.2.8. The first step holds only in $\mathsf{IndCoh}(X)$ and not in $\mathsf{QCoh}(X)$, as $k \in \mathsf{Coh}(X)$ is compact in $\mathsf{IndCoh}(X)$, but $k \notin \mathsf{Perf}(X)$ hence is not compact in $\mathsf{QCoh}(X)$.

The assignment $\mathfrak{X} \mapsto \mathsf{IndCoh}(X)$ is functorial, which should not be a surprise given that to every map f of prestacks we have induced maps on Coh given by $f^!$. Let's denote this contravariant functor by

$$\mathsf{IndCoh}^! : \mathfrak{X} \mapsto \mathsf{IndCoh}(\mathsf{X}), \quad f \mapsto f^!_{\mathsf{IndCoh}}.$$

In fact, IndCoh gives rise to "pushforward" maps as well as "shriek pullback" maps. For a map of prestacks $f : \mathfrak{X} \to \mathfrak{Z}$, we get adjoint functors

$$\mathsf{IndCoh}(\mathfrak{X}) \underset{f^!_{\mathsf{IndCoh}}}{\overset{f^{\mathsf{IndCoh}}_*}{\rightleftarrows}} \mathsf{IndCoh}(\mathfrak{Z}).$$

Proposition 3.2.9 ([GR17, §3, Proposition 3.1.2])**.** *Let $f : \mathfrak{X} \to \mathfrak{Z}$ be a (inf-schematic nil-isomorphism) map of prestacks. Then:*

(1) The functor $f^!_{\mathsf{IndCoh}} : \mathsf{IndCoh}(\mathfrak{Z}) \to \mathsf{IndCoh}(\mathfrak{X})$ admits a left adjoint, denoted by f^{IndCoh}_*.

(2) The functor $f^!_{\mathsf{IndCoh}}$ is conservative.

(3) (Base change) Given a Cartesian diagram

$$\begin{array}{ccc} \mathfrak{X}' & \xrightarrow{\widehat{g}} & \mathfrak{X} \\ {\scriptstyle \widehat{f}}\downarrow & \lrcorner & \downarrow{\scriptstyle f} \\ \mathfrak{Z}' & \xrightarrow[g]{} & \mathfrak{Z} \end{array}$$

we have the natural isomorphism $\widehat{f}^{\mathsf{IndCoh}}_* \circ \widehat{g}^!_{\mathsf{IndCoh}} \xrightarrow{\sim} g^!_{\mathsf{IndCoh}} \circ f^{\mathsf{IndCoh}}_*$.

3.3 Ind-schemes

The essence of ind-schemes is that they're just an infinite union of schemes.

Definition 3.3.1. An **ind-scheme** is, roughly speaking, a (filtered) colimit of closed embeddings of schemes.

Example 3.3.2. Define $\mathbb{A}^\infty := \varinjlim_n \operatorname{Spec} k[x_1, \dots, x_n]$ with the usual closed embeddings. Note that this is **not** the same as $\operatorname{Spec} k[x_1, x_2, \dots]$.

Example 3.3.3. Another famous example is $\mathbb{P}^\infty$, defined to be the colimit of the closed embeddings $\mathbb{P}^1 \hookrightarrow \mathbb{P}^2 \hookrightarrow \mathbb{P}^3 \hookrightarrow \cdots$.

Example 3.3.4. The affine Grassmannian of an algebraic group is also an ind-scheme.

The most important example for us today:

Example 3.3.5. Consider the closed embeddings $\operatorname{Spec} k[t]/t^n \hookrightarrow \operatorname{Spec} k[t]/t^{n+1}$, corresponding to the surjections $k[t]/t^{n+1} \twoheadrightarrow k[t]/t^n$. The colimit $\widehat{\mathbf{0}} := \varinjlim_n \operatorname{Spec} k[t]/t^n$ is an ind-scheme; however, it is *not* a scheme in the usual sense, and in particular is it *not* the same as $\operatorname{Spec} k[[t]]$ (which is the formal completion of the point $0 \in \mathbb{A}^1$). Note that the ind-scheme constructed here has a single closed point, plus some "fuzz," while $\operatorname{Spec} k[[t]]$ has two points - a closed point and a generic point.

Maps from $\operatorname{Spec} R$ to $\widehat{\mathbf{0}}$ are exactly maps which factor through some $\operatorname{Spec} k[t]/t^n$, that is, nilpotent elements. So $\widehat{\mathbf{0}}$ as a functor of points sends $\operatorname{Spec} R$ to the nilpotent elements of R.

3.4 deRham prestack

We'll study an extremely important prestack, called the deRham prestack.

Definition 3.4.1. Let $\mathfrak{X}$ be a prestack (for example, a scheme). We define the **deRham prestack** $\mathfrak{X}_{dR}$ of $\mathfrak{X}$ by its functor of points: for every scheme S,

$$\operatorname{Hom}(S, \mathfrak{X}_{dR}) := \operatorname{Hom}(S^{\text{red}}, \mathfrak{X}).$$

The assignment $\mathfrak{X} \mapsto \mathfrak{X}_{dR}$ is functorial, and this functor commutes with both limits and colimits. Call this functor dR. As a result, to every map of prestacks $f : \mathfrak{X} \to \mathfrak{Z}$ we have an induced map of deRham prestacks $f_{dR} : \mathfrak{X}_{dR} \to \mathfrak{Z}_{dR}$. We will revisit this in §4.1 in defining D-modules on prestacks.

For every prestack $\mathfrak{X}$ there is a canonical projection

$$p_{\mathfrak{X}} : \mathfrak{X} \to \mathfrak{X}_{dR},$$

corresponding to the pre-composition with the map $S^{\text{red}} \to S$ for any test scheme S; concretely, we have

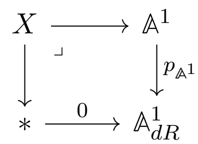
$$(S \to \mathfrak{X}) \mapsto (S^{\text{red}} \to S \to \mathfrak{X}) \leftrightarrow (S \to \mathfrak{X}_{dR}).$$

If $\mathfrak{X} = \operatorname{Spec} A$ is an affine scheme and $S = \operatorname{Spec} R$ is affine, this is just the canonical post-composition with the map $R \twoheadrightarrow R^{\text{red}} = R/\operatorname{Nil}(R)$: we have

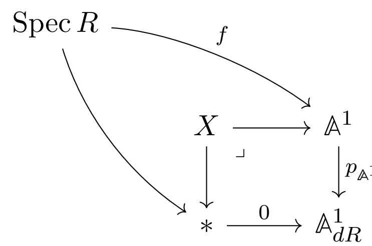
$$(A \to R) \mapsto (A \to R \twoheadrightarrow R^{\text{red}}).$$

We should think of $\mathfrak{X}_{dR}$ as "pinching the infinitesimal neighborhoods around every single point," so that it kills any "fuzz" (corresponding to nilpotents) around every point simultaneously. This can be weird to think about - for example, for a reduced scheme X, there doesn't appear to be much fuzz at all. But to construct X_{dR}, we want to kill even the infinitesimal fuzz around the points in X so that we never see nilpotents. As an example, let's see what happens locally around a point in $\mathbb{A}^1$.

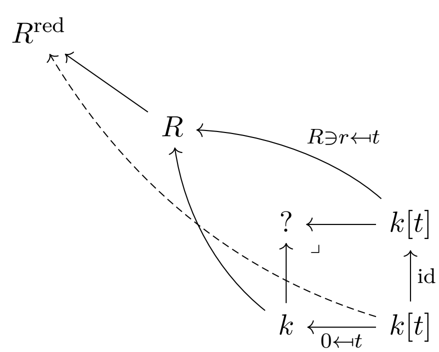
Example 3.4.2. Let's compute the pullback of the two maps $* \xrightarrow{0} \mathbb{A}^1_{dR}$ and $p_{\mathbb{A}^1} : \mathbb{A}^1 \to \mathbb{A}^1_{dR}$. Call the pullback X. Then we have the Cartesian square



Now it suffices to check what a map $\operatorname{Spec} R \to X$ is for every affine k-scheme $\operatorname{Spec} R$; this gives us the diagram



Dualizing this, we have the diagram



This diagram essentially encodes the data of the map $\operatorname{Spec} R \to \mathbb{A}^1$, which is just a map $k[t] \to R$, satisfying certain conditions. The condition for this diagram to commute is just that the composite $\operatorname{Spec} R \to \mathbb{A}^1 \to \mathbb{A}^1_{dR}$ agrees with the composite $\operatorname{Spec} R \to \operatorname{Spec} k \to \mathbb{A}^1_{dR}$; in rings, this is just that the image of $t \mapsto r \in R$ is such that the image of r under the map $R \twoheadrightarrow R^{\text{red}}$ agrees with the eventual image of t going the other way, which is $t \mapsto 0 \in k \mapsto 0 \in R \mapsto 0 \in R^{\text{red}}$, so $r \in \ker(R \twoheadrightarrow R^{\text{red}})$, i.e. r is nilpotent.

It follows that the pullback X is just the functor sending an affine scheme $\operatorname{Spec} R$ to its set (groupoid) of nilpotents. This is precisely the ind-scheme $\widehat{\mathbf{0}}$ described in Example 3.3.5. So what we see is that in the projection $\mathbb{A}^1 \to \mathbb{A}^1_{dR}$, around every single point, we're essentially "pinching down" the infinitesimal fuzz around the point (even though $\mathbb{A}^1$ is already reduced!), doing the equivalent of the map $\widehat{\mathbf{0}} \to *$ for every point.

3.5 Monadic pairs

Monadic pairs form an extremely important type of adjunction. To motivate what's going on, let's start with the ultimate prototype.

Example 3.5.1. Let $B \to A$ be a map of commutative rings. Then we have the adjunction

$$\begin{array}{c} A\text{−}\mathsf{mod} \\ \mathbf{Ind} \uparrow \dashv \downarrow \mathbf{Res} \\ B\text{−}\mathsf{mod} \end{array}$$

Here, the restriction map simply realizes an A-module M as a B-module via the map $B \to A$, and the induction map sends a B-module N to the A-module $A \otimes_B N$; this is just pushforward and pullback of quasicoherent sheaves on $\operatorname{Spec} B$ and $\operatorname{Spec} A$.

Definition 3.5.2. Let $(\mathcal{C}, \otimes)$ be a monoidal category acting on a category $\mathcal{M}$. The **algebra objects** are objects $X \in \mathcal{C}$ equipped with maps $\mu_X : X \otimes X \to X$ and $\varepsilon_X : \mathbb{1} \to X$ satisfying certain (infinity-categorical) properties. They form a category called $\mathsf{Alg}(\mathcal{C})$.

Let $A \in \mathsf{Alg}(\mathcal{C})$. We can form the **category of A-modules**, denoted by $A\text{−}\mathsf{mod}(\mathcal{M})$, consisting of $M \in \mathcal{M}$ equipped with maps $\mu_M : A \otimes M \to M$ satisfying the usual properties and compatibilities with μ_X and ε_X.

Let's abstract Example 3.5.1.

Definition 3.5.3. Suppose we have an adjunct pair

$$\begin{array}{c} \mathcal{C} \\ F \uparrow \dashv \downarrow G \\ \mathcal{D} \end{array}$$

Denote

$$T := G \circ F \in \mathsf{End}(\mathcal{D}).$$

Then we have adjunctions

$$\varepsilon : \mathrm{id}_{\mathcal{D}} \to GF,$$
$$\eta : FG \to \mathrm{id}_{\mathcal{C}}.$$

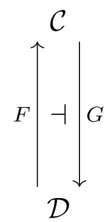
This gives us a map

$$\mu : T \circ T = G(FG)G \xrightarrow{\eta} FG = T.$$

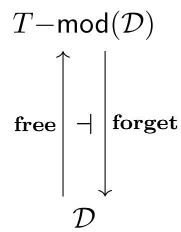
These two maps, $\varepsilon : \mathrm{id}_{\mathcal{D}} \to T$ and $\mu : T \circ T \to T$, turn T into an algebra object in $\mathsf{End}(\mathcal{D})$. When we have an equivalence $\mathcal{C} \simeq T{-}\mathsf{mod}(\mathcal{D})$, then we say that $(\mathcal{C}, \mathcal{D}, F, G)$ form a **monadic adjunction**, and that $\mathcal{C}$ **is monadic over** $\mathcal{D}$.

In Example 3.5.1, ε was the map $B \to A$, and μ was the map $A \otimes A \to A$. On the other hand, T was the map $M \mapsto A \otimes_B M$.

However, given the data of an adjoint pair (F, G) between $\mathcal{C}, \mathcal{D}$, it's not guaranteed that these will actually form a monadic adjunction. Given a pair of adjoint functors



we can form $T \coloneqq G \circ F \in \mathsf{End}(\mathcal{D})$. Then we consider the adjunction

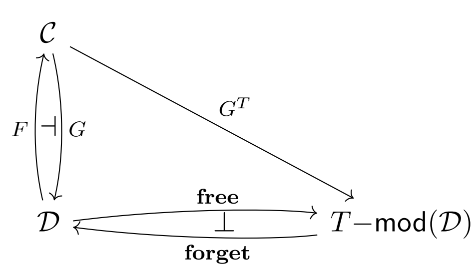


Recall that $T{-}\mathsf{mod}(\mathcal{D})$ has objects (d, α_d) where $d \in \mathcal{D}$ and $\alpha_d : T(d) \to d$ compatible with the relations on T. In Example 3.5.1, this is equivalent to the data of $A \otimes M \to M$ for every B-module M. (The morphisms are just commutative squares indicating the compatibility of the α_d with T.) Then **forget** is the forgetful functor sending $(d, \alpha_d) \mapsto d$. On the other hand, **free** is the induction funfctor $\mathcal{D} \to T{-}\mathsf{mod}(\mathcal{D})$ sending $d \mapsto (T(d), \mu_d)$ where $\mu_d : T \circ T(d) \to T(d)$ is given by the natural transformation $\mu : T \circ T \to T$ encoded in the data of T being an algebra object.

We also have a natural map $G^T : \mathcal{C} \to T{-}\mathsf{mod}(\mathcal{D})$, as follows. Given $d \in \mathcal{D}$, the object $G(d)$ has a natural T-module structure:

$$T(G(d)) = G \circ F \circ G(d) \xrightarrow{G \circ \eta} G(d),$$

so we set $G^T : d \mapsto (G(d), G \circ \eta)$. It turns out that $\textbf{free} = G^T \circ F$:



If G^T induces an equivalence $\mathcal{C} \simeq T\text{−}\mathsf{mod}(\mathcal{D})$, then F, G indeed form a monadic adjunction. In particular, we're looking for an equivalence induced by the original adjoint functors F, G via the functor G^T described above.

Remark 3.5.4. $T\text{−}\mathsf{mod}(\mathcal{D})$ is a terminal object in the category of adjunctions. As such, we're looking for an adjoint pair which is "terminal" in some sense.

However there's actually a very easy way to construct/check for monadic adjunction.

Theorem 3.5.5 (Barr-Beck)**.** *Suppose we have a functor $G : \mathcal{C} \to \mathcal{D}$. Suppose that*

1) G is continuous (preserves colimits).

2) G is conservative (if $G(f)$ is an isomorphism, then f is an isomorphism).

3) G admits a left adjoint F.

Then $\mathcal{C}$ is monadic over $\mathcal{D}$ via the functor G, and (F, G) are a monadic adjunction.

As an immediate corollary, we can deduce the following about IndCoh functors.

Proposition 3.5.6. *In the setup of Proposition 3.2.9, then f_*^{IndCoh} and $f^!_{\mathsf{IndCoh}}$ form a monadic pair.*

4 D-modules via crystals

Following [GR17, §4], we define the category of D-modules (on a prestack) to be the category of crystals. The reason for working with crystals is that it's easy to set up the necessary functorialities. We'll see that when our prestack is a smooth scheme, this construction agrees with the classical definition and notion of D-modules. As a bonus, we'll compute the category of D-modules on BG.

4.1 Crystals

Definition 4.1.1. Let $\mathfrak{X}$ be a prestack. We define the **category of crystals** on $\mathfrak{X}$ to be

$$\mathsf{Crys}(\mathfrak{X}) \coloneqq \mathsf{IndCoh}(\mathfrak{X}_{dR}).$$

This assignment is functorial; we have the contravariant functor

$$\mathsf{Crys}^! \coloneqq \mathsf{IndCoh}^! \circ \mathsf{dR}.$$

This will be our definition for D-modules on a prestack: **the category of D-modules on a prestack $\mathfrak{X}$ is defined to be $\mathsf{Crys}(\mathfrak{X}) = \mathsf{IndCoh}(\mathfrak{X}_{dR})$.**

Since this assignment is functorial, for a map of prestacks $f : \mathfrak{X} \to \mathfrak{Z}$, we get a resulting map $\mathsf{Crys}(\mathfrak{Z}) \to \mathsf{Crys}(\mathfrak{X})$, which we will denote by $f^!_{dR}$. When f is a reasonably nice map (for technical conditions, it should be ind-nil-proper), this functor has a left adjoint denoted by f_*^{dR}, which agrees with the composition $(f_{dR})_*^{\mathsf{IndCoh}}$, i.e. the resulting pushforward map on IndCoh after applying the deRham functor dR.

We also have a more concrete description of $\mathsf{Crys}(\mathfrak{X})$.

Theorem 4.1.2. *Let $\mathfrak{X}$ be a prestack. Then*

$$\mathsf{Crys}(\mathfrak{X}) \simeq \varprojlim_{f\colon \operatorname{Spec} A \to \mathfrak{X}} \mathsf{Crys}(\operatorname{Spec} A).$$

4.2 oblv and ind

Let $\mathfrak{X}$ be a prestack. Then we have a natural projection map

$$p_{\mathfrak{X}} : \mathfrak{X} \to \mathfrak{X}_{dR}.$$

This induces a natural map

$$p^!_{\mathsf{IndCoh}} : \mathsf{Crys}(\mathfrak{X}) \to \mathsf{IndCoh}(\mathfrak{X}).$$

We call this map $\mathbf{oblv}_{\mathfrak{X}}$, or simply **oblv**, and it plays the role of the forgetful functor from the "D-module" to the "underlying sheaf." Under certain technical conditions (i.e., admits deformation theory), **oblv** admits a left adjoint p_*^{IndCoh}, which we'll denote by $\mathbf{ind}_{\mathfrak{X}}$, or simply **ind**, for induction.

We can also describe the functoriality of **oblv** and **ind** more concretely. Let $f : \mathfrak{X} \to \mathfrak{Z}$ be a (ind-inf-schematic) morphism of prestacks (which admit deformation theory, laft?). Then we have the following commutative diagrams of functors. We have the interaction between **oblv** and $f^!$:

$$\begin{array}{ccc} \mathsf{IndCoh}(\mathfrak{X}) & \xleftarrow{\mathbf{oblv}} & \mathsf{Crys}(\mathfrak{X}) \\ {\scriptstyle f^!_{\mathsf{IndCoh}}}\uparrow & & \uparrow{\scriptstyle f^!_{dR}} \\ \mathsf{IndCoh}(\mathfrak{Z}) & \xleftarrow{\mathbf{oblv}} & \mathsf{Crys}(\mathfrak{Z}) \end{array}$$

We also have the interaction between **ind** and f_*.

$$\begin{array}{ccc} \mathsf{IndCoh}(\mathfrak{X}) & \xrightarrow{\mathbf{ind}} & \mathsf{Crys}(\mathfrak{X}) \\ {\scriptstyle f_*^{\mathsf{IndCoh}}}\downarrow & & \downarrow{\scriptstyle f_*^{dR}} \\ \mathsf{IndCoh}(\mathfrak{Z}) & \xrightarrow{\mathbf{ind}} & \mathsf{Crys}(\mathfrak{Z}) \end{array}$$

Remark 4.2.1. I was told that the people who named **oblv** did so after "obliviate," the spell in Harry Potter which is effectively the forgetful functor. I guess they didn't extend the same courtesy to the induction functor **ind**.

So for a prestack $\mathfrak{X}$, we have the adjoint pair

$$\begin{array}{c} \mathsf{Crys}(\mathfrak{X}) \\ p_*^{\mathsf{IndCoh}}=\mathbf{ind} \uparrow \ \dashv \ \downarrow \mathbf{oblv}=p^!_{\mathsf{IndCoh}} \\ \mathsf{IndCoh}(\mathfrak{X}) \end{array} \tag{1}$$

Proposition 4.2.2. *The pair (1) is monadic.*

Essentially, we just need to check that **oblv** is conservative, which follows from Proposition 3.2.9.

Definition 4.2.3. To a prestack $\mathfrak{X}$ we define

$$\mathrm{Diff}_{\mathfrak{X}} := p^!_{\mathsf{IndCoh}} \circ p_*^{\mathsf{IndCoh}} = \mathbf{oblv} \circ \mathbf{ind}$$

for the projection map $p_{\mathfrak{X}} : \mathfrak{X} \to \mathfrak{X}_{dR}$, as in (1). This is an algebra object (hence endomorphism) in $\mathsf{IndCoh}(\mathfrak{X})$.

Due to the fact that **oblv** and **ind** (for the map $p_{\mathfrak{X}} : \mathfrak{X} \to \mathfrak{X}_{dR}$) form a monadic adjunction, we have the equivalence

$$\mathsf{Crys}(\mathfrak{X}) \simeq \mathrm{Diff}_{\mathfrak{X}}-\mathsf{mod}(\mathsf{IndCoh}(\mathfrak{X})).$$

Thus, to identify $\mathsf{Crys}(\mathfrak{X})$ with D-modules on $\mathfrak{X}$, **$\mathbf{Diff}_{\mathfrak{X}}$ plays the role of the sheaf of differential operators, and IndCoh plays the role of quasicoherent sheaves**. This part is *literally* true in the case of quasicoherent sheaves: see §4.5.

4.3 t-structure on Crys

The category $\mathsf{Crys}(\mathfrak{X})$ carries a canonical t-structure, characterized by

$$M \in \mathsf{Crys}(\mathfrak{X})^{\geq 0} \iff \mathbf{oblv}(M) \in \mathsf{IndCoh}(\mathfrak{X})^{\geq 0}.$$

4.4 Base change

Theorem 4.4.1 ([GR17, §4.2.1.3])**.** *Suppose we have a Cartesian diagram of prestacks*

$$\begin{array}{ccc} \widehat{\mathfrak{X}} & \xrightarrow{\widehat{g}} & \mathfrak{X} \\ {\scriptstyle \widehat{f}}\downarrow & \lrcorner & \downarrow{\scriptstyle f} \\ \widehat{\mathfrak{Z}} & \xrightarrow[g]{} & \mathfrak{Z} \end{array}$$

with some technical conditions on f (i.e., f is ind-nil-schematic). Then we have a canonical isomorphism

$$\widehat{f}^{dR}_* \circ \widehat{g}^!_{dR} \xrightarrow{\sim} g^!_{dR} \circ f^{dR}_*.$$

4.5 D-modules on (smooth) schemes

Let's see why this indeed recovers the classical notion of D-modules on smooth schemes.

First let's consider $X = \mathbb{A}^1$. We know that $X_{dR} = \mathbb{A}^1_{dR}$, as a functor sends $\operatorname{Spec} R \mapsto R^{\text{red}}$. This means that we can represent this deRham prestack as $X_{dR} = \mathbb{A}^1_{dR} \cong \mathbb{A}^1/\widehat{\mathbf{0}}$, where $\widehat{\mathbf{0}}$ is the functor sending $\operatorname{Spec} R$ to the nilradical of R, as explained in Example 3.3.5.

But then we have a quotient stack, so

$$\mathsf{QCoh}(\mathbb{A}^1_{dR}) = \mathsf{QCoh}(\mathbb{A}^1)^{\widehat{\mathbf{0}}-\text{equivariant}}.$$

But this is just (the derived category of) $k[t]$-modules M which are also $\mathcal{O}(\widehat{\mathbf{0}})$-comodules, or equivalently $k[t]$-modules M which also carry an $\mathcal{O}(\widehat{\mathbf{0}})^\vee$-action. Since $\widehat{\mathbf{0}} = \varinjlim \operatorname{Spec} k[t]/t^n$, we have $\mathcal{O}(\widehat{\mathbf{0}}) = \mathcal{O}(\varinjlim \operatorname{Spec} k[t]/t^n) = \varprojlim k[t]/t^n = k[[t]]$, whose dual vector space is $k\left[\frac{\partial_x^n}{n!} \mid n \in \mathbb{Z}_{>0}\right]$. So M should be a module for both $k[t] \otimes k\left[\frac{\partial_x^n}{n!}\right]$, which just the Weyl algebra $W = k\langle x, \frac{\partial_x^n}{n!}\rangle$. Therefore for $\mathbb{A}^1$, we see that

$$\mathsf{Crys}(\mathbb{A}^1) := \mathsf{IndCoh}(\mathbb{A}^1_{dR}) = \mathsf{QCoh}(\mathbb{A}^1_{dR}) \simeq W{-}\mathsf{mod},$$

which indeed recovers the classical notion of a D-module on $\mathbb{A}^1$.

Now let's see what the functors oblv and ind do. We have the natural projection

$$p_{\mathbb{A}^1} : \mathbb{A}^1 \to \mathbb{A}^1_{dR}.$$

This induces the monadic adjunction

$$\begin{array}{c} \mathsf{IndCoh}(\mathbb{A}^1_{dR}) \\ p_*^{\mathsf{IndCoh}} = \mathbf{ind} \uparrow \dashv \downarrow \mathbf{oblv} = p^!_{\mathsf{IndCoh}} \\ \mathsf{IndCoh}(\mathbb{A}^1) \end{array}$$

Since $\mathsf{IndCoh}(\mathbb{A}^1) = k[t]{-}\mathsf{mod}$ and $\mathsf{IndCoh}(\mathbb{A}^1_{dR}) = W{-}\mathsf{mod}$, we have that $\mathsf{oblv} = p^!_{\mathsf{IndCoh}}$ is just the standard pullback map from $W - \mathsf{mod}$ to $k[t]{-}\mathsf{mod}$, i.e. the forgetful functor from D-modules on $\mathbb{A}^1$ to the underlying quasicoherent sheaves on $\mathbb{A}^1$. On the other hand, ind is just the left adjoint to oblv, which in this case sends a module M to the W-module $W \otimes_{k[t]} M$. So oblv and ind are really the standard pair of induction-restriction functors. To be even more explicit, we have the identification

$$\mathsf{Crys}(\mathbb{A}^1) \simeq \mathrm{Diff}_{\mathbb{A}^1}{-}\mathsf{mod}(\mathsf{IndCoh}(\mathbb{A}^1)).$$

Here, $\mathsf{IndCoh}(\mathbb{A}^1) = \mathsf{QCoh}(\mathbb{A}^1) = k[t]{-}\mathsf{mod}$. We can compute $\mathrm{Diff}_{\mathbb{A}^1}$ explicitly on a $k[t]$-module M: $\mathrm{Diff}_{\mathbb{A}^1}(M)$ consists of the data of a map $W \otimes_{k[t]} M \to M$ compatible with the $k[t]$-module structure on M. That's precisely the data of a $D(\mathbb{A}^1)$-module structure on M! So the statement $\mathsf{Crys}(\mathbb{A}^1) \simeq \mathrm{Diff}_{\mathbb{A}^1}{-}\mathsf{mod}(\mathsf{IndCoh}(\mathbb{A}^1))$ recovers the classical definition: a $D(\mathbb{A}^1)$-module is just quasicoherent sheaf on $\mathbb{A}^1$

equipped with an action of the sheaf of differential operators $D(\mathbb{A}^1)$ (which in this case is just a compatible action of the Weyl algebra).

Naturally, this construction generalizes immediately to $\mathbb{A}^n$. It then follows from étale descent and using the fact that any smooth scheme has an étale cover by affine spaces that

$$\mathsf{Crys}(X) \simeq D\text{−}\mathsf{mod}(X).$$

Namely, for any smooth scheme X, **crystals on X agree with D-modules on X**, so indeed we can afford to call them D-modules.

Remark 4.5.1. For a more direct and concrete computation that crystals on smooth k-schemes are just D-modules, see [Lur09, Theorem 0.4].

4.6 Example: D-modules on BG

Let's describe an explicit example: the D-modules on BG, where G is reductive and finite type (in characteristic 0, hence also smooth and affine) group scheme. We start with the covering map

$$\sigma : * \to */G = BG.$$

This induces the monadic adjunction

$$\begin{array}{c} \mathsf{Crys}(BG) \\ \sigma_!^{dR} \Big\uparrow \dashv \Big\downarrow \sigma^!_{dR} \\ \mathsf{Crys}(*) \end{array}$$

Let $\mathcal{C} = \mathsf{Crys}(BG)$ and $\mathcal{D} = \mathsf{Crys}(*)$, as in the setup of Definition 3.5.3. Note that

$$\mathcal{D} = \mathsf{IndCoh}(*_{dR}) = \mathsf{IndCoh}(*) = \mathsf{QCoh}(*) = \mathsf{Vec}_k,$$

the category of vector spaces over k. On the other hand, writing

$$T := \sigma^!_{dR} \circ \sigma_!^{dR} \in \mathsf{Alg}(\mathsf{Vec}_k),$$

we have that

$$\mathsf{Crys}(BG) = \mathcal{C} = T\text{−}\mathsf{mod}(\mathcal{D}) = T\text{−}\mathsf{mod}(\mathsf{Vec}_k).$$

So it suffices to determine how T acts on vector spaces. But in fact the monoidal category of endomorphisms of Vec_k (together with composition of functors) is just Vec_k itself. By definition the category of endomorphisms consists of exact continuous functors, hence commutes with colimits - but any vector space is just a colimit of the generator k, hence any such functor is uniquely determined by where it sends the generator k. So under this interpretation, T is itself identified with a k-algebra (loosely... more correctly, an algebra object in Vec_k, which should be something like a differential graded algebra), i.e. a k-vector space with multiplication. In order to determine what algebra this is, we just need to check where T sends the generator k.

Now we turn to the Cartesian square

$$\begin{array}{ccc} G & \xrightarrow{\pi} & * \\ {\scriptstyle\pi}\downarrow & \lrcorner & \downarrow{\scriptstyle\sigma} \\ * & \xrightarrow{\sigma} & BG \end{array}$$

which we computed in Example 2.7.4. Using the base change theorem 4.4.1, we find that

$$\sigma^!_{dR} \circ \sigma^{dR}_!(k) = \pi^{dR}_! \circ \pi^!_{dR}(k).$$

But now we apply the Verdier duality functors. Noting that $\mathbb{D}(k) = k$, we find that

$$\begin{aligned} \sigma^!_{dR} \circ \sigma^{dR}_!(k) &= \pi^{dR}_! \circ \pi^!_{dR}(k), \\ &= \mathbb{D}\pi_*\mathbb{D}\mathbb{D}\pi^*\mathbb{D}(k), \\ &= \mathbb{D}\pi_*\pi^*(k), \\ &= (\pi_*\pi^*(k))^\vee, \\ &= H^\bullet(G,k)^\vee, \\ &= H_\bullet(G,k), \end{aligned}$$

where $H_\bullet(G,k)$ is the homology ring of G with coefficients in k (realized as the dual of $H^\bullet(G,k)$, the cohomology ring). Therefore,

$$\mathsf{Crys}(BG) = H_\bullet(G,k)-\mathsf{mod}.$$

5 D-modules via sheaves

5.1 Motivation from classical case

There's another way to define D-modules, which is as a quasicoherent sheaf on $X \times X$. Suppose X is a smooth variety. Then we can define the sheaf of differential operators D_X on X as the sheaf of Grothendieck differential operators, and this has a natural algebra structure on it. This naturally has the structure of an $\mathcal{O}_X$-module, since it's defined on X. However, $\mathcal{O}_X$ acts on D_X on both the left and the right; since $\mathcal{O}_X$ is *not* in the center of D_X, this gives us two different actions. So actually we can define D_X on $X \times X$, corresponding to the left and right multiplication by $\mathcal{O}_X$:

$$\mathcal{O}_X \curvearrowright D_X \curvearrowleft \mathcal{O}_X, \qquad D_X \in \mathsf{QCoh}(X \times X).$$

The natural question is: what is the ring structure? For this, we need to define a map $D_X \otimes_{\mathcal{O}_X} D_X \to D_X$. It will turn out that the multiplication map is just $f \otimes g \mapsto fg$, where f, g are some product of differential operators and functions, but we need to abstract this so that we have a construction which works on a general prestack. Let's see how this works.

5.2 Constructing $D_{\mathfrak{X}} \in \mathsf{IndCoh}(\mathfrak{X} \times \mathfrak{X})$

Let $\mathfrak{X}$ be a prestack. We will construct the object $D_{\mathfrak{X}} \in \mathsf{IndCoh}(\mathfrak{X} \times \mathfrak{X})$.

We have a canonical projection map

$$p_{\mathfrak{X}} : \mathfrak{X} \to \mathfrak{X}_{dR}.$$

Then we have the Cartesian diagram

$$\begin{CD} \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} @>>> \mathfrak{X} \\ @VVV @VV{p_{\mathfrak{X}}}V \\ \mathfrak{X} @>>{p_{\mathfrak{X}}}> \mathfrak{X}_{dR} \end{CD}$$

Let us denote

$$\mathfrak{X}^2_{\widehat{\Delta}} := \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X}$$

as in the Cartesian square above; then we have a natural map

$$\widehat{\Delta} : \mathfrak{X}^2_{\widehat{\Delta}} \to \mathfrak{X}^2, \quad \mathfrak{X}^2 := \mathfrak{X} \times \mathfrak{X}.$$

To each prestack $\mathfrak{Z}$ we also have a natural projection map to a point

$$a_{\mathfrak{Z}} : \mathfrak{Z} \to *.$$

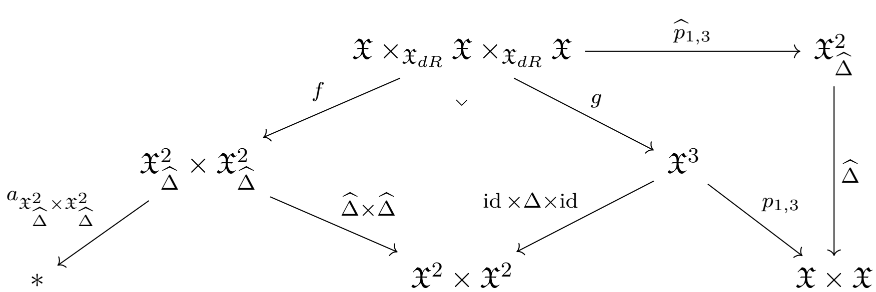
Definition 5.2.1. We define the **dualizing complex** on $\mathfrak{Z}$ to be the ind-coherent sheaf $\omega_{\mathfrak{Z}} := a^!_{\mathfrak{Z}}\, \underline{k}$, where $\underline{k}$ is the constant sheaf (on $* = \operatorname{Spec} k$).

Definition 5.2.2. We define $D_{\mathfrak{X}}$ to be the ind-coherent sheaf on $\mathfrak{X} \times \mathfrak{X}$

$$D_{\mathfrak{X}} := \widehat{\Delta}_* \omega_{\mathfrak{X}^2_{\widehat{\Delta}}} = \widehat{\Delta}_* a^!_{\mathfrak{X}^2_{\widehat{\Delta}}}\, \underline{k}.$$

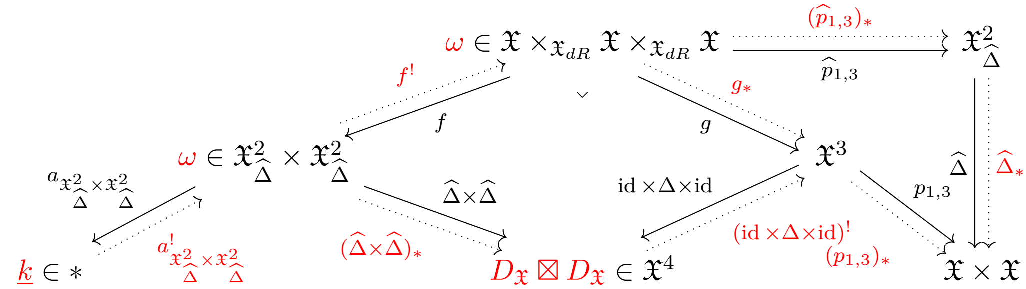
5.3 Multiplication on $D_{\mathfrak{X}}$

In this subsection, we'll describe the "multiplication" map on $D_{\mathfrak{X}}$. It will arise out of the following diagram.



The maps $p_{1,3} : \mathfrak{X}^3 \to \mathfrak{X}^2$ are $\mathrm{id} \times a_{\mathfrak{X}} \times \mathrm{id}$, sending $(x, y, z) \mapsto (x, z)$, and similarly for $\widehat{p}_{1,3}$. The maps f and g are the induced maps from the pullback.

Let's now construct a map $D_{\mathfrak{X}} \otimes_{\mathcal{O}_{\mathfrak{X}}} \to D_{\mathfrak{X}}$. For a map $f : X \to Y$, we have $f_*^{\mathsf{IndCoh}} : \mathsf{IndCoh}(X) \to \mathsf{IndCoh}(Y)$ and $f^!_{\mathsf{IndCoh}} : \mathsf{IndCoh}(Y) \to \mathsf{IndCoh}(X)$. For the sake of brevity, the IndCoh notation is omitted, and just written as f_* and $f^!$.



First we need to justify that the red is indeed correct.

Lemma 5.3.1. *The (ind-coherent) sheaves in red are the results of the $*$-pushforward and $!$-pullback functors, also written in red.*

Proof. First we start with $\underline{k}$ on $*$. Next, by definition, $a^!_{\mathfrak{X}^2_{\widehat{\Delta}} \times \mathfrak{X}^2_{\widehat{\Delta}}} \underline{k} = \omega_{\mathfrak{X}^2_{\widehat{\Delta}} \times \mathfrak{X}^2_{\widehat{\Delta}}} = \omega_{\mathfrak{X}^2_{\widehat{\Delta}}} \boxtimes \omega_{\mathfrak{X}^2_{\widehat{\Delta}}}$. Next, it's clear that the map $(\widehat{\Delta} \times \widehat{\Delta})_*$ sends it to $\widehat{\Delta}_* \omega_{\mathfrak{X}^2_{\widehat{\Delta}}} \boxtimes \widehat{\Delta}_* \omega_{\mathfrak{X}^2_{\widehat{\Delta}}} = D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}}$. Taking $f^!$ instead yields

$$f^! \omega_{\mathfrak{X}^2_{\widehat{\Delta}} \times \mathfrak{X}^2_{\widehat{\Delta}}} = f^! a^!_{\mathfrak{X}^2_{\widehat{\Delta}} \times \mathfrak{X}^2_{\widehat{\Delta}}} \underline{k} = a^!_{\mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X}} \underline{k} = \omega_{\mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X}}.$$

The fact that $g_* \omega = (\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}}$ comes from base change; the fact that the last two arrows also compose to $\widehat{\Delta}_* (\widehat{p}_{1,3})_* \omega$ comes from the fact that $\widehat{\Delta}_* (\widehat{p}_{1,3})_* = (\widehat{\Delta} \circ \widehat{p}_{1,3})_* = (p_{1,3} \circ g)_* = (p_{1,3})_* \circ g_*$. □

As a result:

Corollary 5.3.1.1. *We have*

$$(p_{1,3})_* (\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}} = \widehat{\Delta}_* (\widehat{p}_{1,3})_* (\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}}.$$

Proof. By the lemma (and diagram) above, we have

$$\begin{aligned}
\widehat{\Delta}_* (\widehat{p}_{1,3})_* (\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}} &= (\widehat{\Delta} \circ \widehat{p}_{1,3})_* \left((\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}} \right), \\
&= (p_{1,3} \circ g)_* \omega_{\mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X}}, \\
&= (p_{1,3})_* g_* f^! \omega_{\mathfrak{X}^2_{\widehat{\Delta}} \times \mathfrak{X}^2_{\widehat{\Delta}}}, \\
&= (p_{1,3})_* (\mathrm{id} \times \Delta \times \mathrm{id})^! (\widehat{\Delta} \times \widehat{\Delta})_* \omega_{\mathfrak{X}^2_{\widehat{\Delta}} \times \mathfrak{X}^2_{\widehat{\Delta}}}, \\
&= (p_{1,3})_* (\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}}.
\end{aligned}$$

□

Lemma 5.3.2. *We have an adjunction* $(\widehat{p}_{1,3})_* \dashv (\widehat{p}_{1,3})^!$.

Proof. The map $\widehat{p}_{1.3}$ arises out of the Cartesian square

$$\begin{array}{ccc} \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} & \xrightarrow{\widehat{p}_{1,3}} & \mathfrak{X} \times_{\mathfrak{X}_{dR}} \mathfrak{X} \\ \downarrow & \lrcorner & \downarrow \\ \mathfrak{X} & \xrightarrow[p_{\mathfrak{X}}]{} & \mathfrak{X}_{dR} \end{array}$$

Since $(p_{\mathfrak{X}})_* \dashv (p_{\mathfrak{X}})^!$, by base change, the same is true for $\widehat{p}_{1,3}$. ☐

Definition 5.3.3. We define the "multiplication" map $\mu_{D_{\mathfrak{X}}} : (p_{1,3})_*(\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}} \to D_{\mathfrak{X}}$ as follows. Since $(\widehat{p}_{1,3})_* \dashv (\widehat{p}_{1,3})^!$, the identity map

$$\mathrm{id} : (\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}} \to (\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}}$$

gives rise to a map

$$\widetilde{\mu} : (\widehat{p}_{1,3})_* (\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}} \to \omega_{X^2_{\widehat{\Delta}}}.$$

Then we define

$$\mu_{D_{\mathfrak{X}}} = \widehat{\Delta}(\widetilde{\mu}) : \quad \underbrace{\widetilde{\Delta}_* (\widehat{p}_{1,3})_* (\widehat{p}_{1,3})^! \omega_{X^2_{\widehat{\Delta}}}}_{(p_{1,3})_*(\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}}} \to \underbrace{\widehat{\Delta}_* \omega_{X^2_{\widehat{\Delta}}}}_{D_{\mathfrak{X}}}.$$

This gives us a map $(p_{1,3})_*(\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}} \to D_{\mathfrak{X}}$, as a map of ind-coherent sheaves on $\mathfrak{X} \times \mathfrak{X}$. The reason for the name "multiplication" will become apparent soon. However, for now, note that for X a smooth scheme, we actually have

$$(p_{1,3})_*(\mathrm{id} \times \Delta \times \mathrm{id})^! D_X \boxtimes D_X = D_X \otimes_{\mathcal{O}_X} D_X,$$

hence the name is justified (at least in this case).

5.4 D-modules via $D_{\mathfrak{X}}$

In order to describe the category of D-modules, we'll need several facts. First, $\mathsf{IndCoh}(\mathfrak{X} \times \mathfrak{X})$ is monoidal, but we have two monoidal structures. The first is $- \otimes^! -$; this holds for $\mathsf{IndCoh}(\mathfrak{Z})$ for any prestack $\mathfrak{Z}$, not just those which are of the form $\mathfrak{X} \times \mathfrak{X}$, and it comes from the map

$$\Delta : \mathfrak{Z} \to \mathfrak{Z}^2.$$

The monoidal structure here is defined by

$$\mathcal{F} \otimes^! \mathcal{G} := \Delta^! (\mathcal{F} \boxtimes \mathcal{G}).$$

The second monoidal structure, special to $\mathfrak{X} \times \mathfrak{X}$, comes from the maps

$$\begin{array}{ccc} & \mathfrak{X} \times \mathfrak{X} \times \mathfrak{X} & \\ {}^{\mathrm{id} \times \Delta \times \mathrm{id}} \swarrow & & \searrow {}^{\mathrm{id} \times a_{\mathfrak{X}} \times \mathrm{id}} \\ \mathfrak{X}^2 \times \mathfrak{X}^2 & & \mathfrak{X}^2 \end{array}$$

Then to $\mathcal{F}, \mathcal{G} \in \mathsf{IndCoh}(\mathfrak{X}^2)$ we define

$$\mathcal{F} \star \mathcal{G} := (p_{1,3})_*(\mathrm{id} \times \Delta \times \mathrm{id})^! \mathcal{F} \boxtimes \mathcal{G}$$

on $\mathfrak{X}^2$. We call this monoidal operation **convolution** and will denote it by $(- \star -)$. In particular, note that

$$(p_{1,3})_*(\mathrm{id} \times \Delta \times \mathrm{id})^! D_{\mathfrak{X}} \boxtimes D_{\mathfrak{X}} = D_{\mathfrak{X}} \star D_{\mathfrak{X}},$$

so that the map $\mu_{D_{\mathfrak{X}}}$ becomes a map

$$\mu_{D_{\mathfrak{X}}} : D_{\mathfrak{X}} \star D_{\mathfrak{X}} \to D_{\mathfrak{X}},$$

thus justifying the name "multiplication." This turns $D_{\mathfrak{X}}$ into an algebra object:

$$D_{\mathfrak{X}} \in \mathsf{Alg}((\mathsf{IndCoh}(\mathfrak{X} \times \mathfrak{X}), \star)).$$

Second, the category $\mathsf{IndCoh}(\mathfrak{X})$ is an $(\mathsf{IndCoh}(\mathfrak{X}^2), \star)$-module (using convolution). The action is given as follows. Let $\mathrm{pr}_1, \mathrm{pr}_2$ be the two projection maps $\mathfrak{X}^2 \to \mathfrak{X}$. Then for $\mathcal{M} \in \mathsf{IndCoh}(\mathfrak{X})$ and $\mathcal{A} \in \mathsf{IndCoh}(\mathfrak{X}^2)$, the action is given by

$$\mathcal{A} \cdot \mathcal{M} := (\mathrm{pr}_2)_* \left(\mathcal{A} \otimes^! \mathrm{pr}_1^! \mathcal{M} \right).$$

Finally, knowing that $D_{\mathfrak{X}}$ is an algebra object in $\mathsf{IndCoh}(\mathfrak{X}^2)$ and $\mathsf{IndCoh}(\mathfrak{X})$ is a module for the category $\mathsf{IndCoh}(\mathfrak{X}^2)$, then we can form the category

$$D_{\mathfrak{X}}\text{−}\mathsf{mod}(\mathsf{IndCoh}(\mathfrak{X})).$$

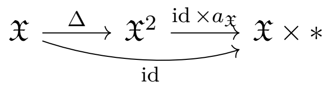
Remark 5.4.1. The monoidal category $\mathsf{IndCoh}(X \times X)$ equipped with convolution can be viewed as a categorification of integral kernels. In the finite-dimensional setting, these are just matrices. The $\otimes^!$ operation on $\mathsf{IndCoh}(X \times X)$ corresponds to pointwise multiplication, which in the finite-dimensional setting corresponds to multiplying matrices entrywise. However, the convolution monoidal structure corresponds to function composition, and in the finite-dimensional setting, corresponds to multiplying matrices the usual way. (Composing integral transforms is even given by convolution of the kernels.) So our "multiplication" map is perhaps better described as "composition" map, in the same way that multiplication on the sheaf D_X on a smooth scheme X corresponds to composition of the endomorphisms of $\mathcal{O}_X$, not pointwise multiplication of the endomorphisms.

5.5 The two constructions agree

First, we need to know the identity under $\otimes^!$.

Lemma 5.5.1. *Let $\mathcal{F} \in \mathsf{IndCoh}(\mathfrak{X})$. Then $\mathcal{F} \otimes^! \omega_{\mathfrak{X}} = \mathcal{F}$.*

Proof. We have the diagram



Note that $\mathrm{id} \times a_{\mathfrak{X}} = \mathrm{pr}_1$. From this, we compute that

$$\begin{aligned}
\mathcal{F} \otimes^! \omega_{\mathfrak{X}} &= \Delta^!(\mathcal{F} \boxtimes \omega_{\mathfrak{X}}),\\
&= \Delta^!(\mathrm{id}^! \mathcal{F} \boxtimes a_{\mathfrak{X}}^! \underline{k}),\\
&= \Delta^! \mathrm{pr}_1^!(\mathcal{F} \boxtimes \underline{k}),\\
&= \mathrm{id}^!(\mathcal{F}),\\
&= \mathcal{F}.
\end{aligned}$$

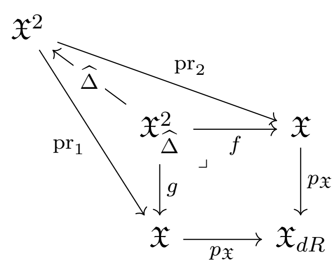
□

This means that ω is the identity element for the monoidal structure $\otimes^!$.

Theorem 5.5.2. *The operation $D_{\mathfrak{X}} \cdot -$ is precisely the endofunctor $\mathit{Diff}_{\mathfrak{X}}$. In particular, we have an equivalence*

$$\mathsf{Crys}(\mathfrak{X}) \simeq D_{\mathfrak{X}}\text{--}\mathsf{mod}(\mathsf{IndCoh}(\mathfrak{X})).$$

Proof. We have the Cartesian square, along with maps to and from $\mathfrak{X}^2$:



Now let $\mathcal{F} \in \mathsf{IndCoh}(\mathfrak{X})$. Then using Lemma 5.5.1 and Theorem 4.4.1 (base change), we can compute

$$\begin{aligned}
D_{\mathfrak{X}} \cdot \mathcal{F} &= (\mathrm{pr}_2)_*(\widehat{\Delta}_* \omega \otimes^! (\mathrm{pr}_1)^! \mathcal{F}),\\
&= (\mathrm{pr}_2)_* \widehat{\Delta}_*(\omega \otimes^! \widehat{\Delta}^! (\mathrm{pr}_1)^! \mathcal{F}),\\
&= (\mathrm{pr}_2 \circ \widehat{\Delta})_* (\mathrm{pr}_1 \circ \widehat{\Delta})^! \mathcal{F},\\
&= f_* g^! \mathcal{F},\\
&= (p_{\mathfrak{X}})^! (p_{\mathfrak{X}})_* \mathcal{F},\\
&= \mathbf{oblv} \circ \mathbf{ind}(\mathcal{F}),\\
&= \mathrm{Diff}_{\mathfrak{X}}(\mathcal{F}).
\end{aligned}$$

□

5.6 D-modules on BG, part 2

Let's compute D_{BG}, and from this deduce the category of D-modules on BG, in a different way than in §4.6. We need some conditions on G again, so assume that G is reductive (hence smooth affine, since we're in characteristic 0) and finite type.

First, we want to compute the pullback

$$\begin{array}{ccc} ? & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow \\ BG^2_{\widehat{\Delta}} & \xrightarrow[\widehat{\Delta}]{} & BG^2 \end{array}$$

where $* \to BG^2$ is the covering map. In other words, we want to compute

$$(BG \times_{BG_{dR}} BG) \times_{BG \times BG} *.$$

We can compute this pullback as the limit of the following diagram:

$$\begin{array}{ccccc} * & \longrightarrow & * & \longleftarrow & * \\ \downarrow & & \downarrow & & \downarrow \\ BG & \longrightarrow & * & \longleftarrow & BG \\ \uparrow & & \uparrow & & \uparrow \\ BG & \longrightarrow & BG_{dR} & \longleftarrow & BG \end{array}$$

The reason for this is that the limits over the rows are given by $*$, $BG \times BG$, and $BG^2_{\widehat{\Delta}}$; it follows that the limit over the entire diagram is given by the pullback via the maps

$$* \to BG \times BG \leftarrow BG^2_{\widehat{\Delta}},$$

which is indeed the pullback we wish to compute. If we instead compute the limit of the columns first, then the limit over the diagrams becomes the limit of the diagram

$$* \to BG_{dR} \leftarrow *,$$

which (after using the fact that $BG_{dR} = B(G_{dR})$) is just the pullback

$$* \times_{BG_{dR}} * = * \times_{B(G_{dR})} * = G_{dR}.$$

It follows that the pullback of the original Cartesian square is G_{dR}, and we find the Cartesian square

$$\begin{array}{ccc} G_{dR} & \xrightarrow{a_{G_{dR}}} & * \\ f\downarrow & \lrcorner & \downarrow q \\ BG^2_{\widehat{\Delta}} & \xrightarrow{\widehat{\Delta}} & BG^2 \end{array}$$

Now consider $\omega = \omega_{BG^2_{\widehat{\Delta}}}$. Then by base change, we find that

$$q^! D_{BG} = q^! \widehat{\Delta}_* \omega = (a_{G_{dR}})_* f^! \omega = (a_{G_{dR}})_* \omega_{G_{dR}} = \Gamma(G_{dR}, \omega_{G_{dR}}) = H_\bullet^{BM}(G),$$

the Borel-Moore homology of G. So we at least find that the underlying vector space of D_{BG} is $H_\bullet^{BM}(G)$. Since we assume that G is connected, then G acts trivially on $H_\bullet^{BM}(G)$.

Since BG^2 is smooth, we have

$$\mathsf{IndCoh}(BG^2) \simeq \mathsf{QCoh}(BG^2) \simeq \mathsf{Rep}(G^2).$$

This means that we need to understand the G^2-action on a $H_\bullet^{BM}(G)$-module in order to understand D_{BG}.

Remark 5.6.1. The equivalences are not completely straightforward; for example, the identification $\mathsf{QCoh}(BG^2) \simeq \mathsf{Rep}(G^2)$ does not commute with pushforwards. This will be an important detail later.

Let's now identify $\mathsf{IndCoh}(BG^2) \simeq \mathsf{Rep}(G^2)$. Note that the pushforward along $BG \xrightarrow{p_{BG}} *$ corresponds to taking G-invariants, and the pullback along $\mathrm{pr}_1 : BG \times BG \to BG$ corresponds to sending $V \mapsto V$, where $G \times G$ acts by factoring through $G \times 1$. Now we have projection maps

$$\begin{array}{ccc} & BG \times BG & \\ {\scriptstyle \mathrm{pr}_1} \swarrow & & \searrow {\scriptstyle \mathrm{pr}_2} \\ BG & & BG \end{array}$$

and a diagonal map

$$\Delta_{BG^2} : BG^2 \to BG^2 \times BG^2.$$

Let $V \in \mathsf{IndCoh}(BG) \simeq \mathsf{Rep}(G)$. Then

$$\begin{aligned} \mathrm{Diff}_{BG}(V) &= (\mathrm{pr}_2)_* \left(D_{BG} \otimes^! (\mathrm{pr}_1)^! V \right), \\ &= (\mathrm{pr}_2)_* (\Delta_{BG^2})^! \left(D_{BG} \boxtimes (\mathrm{pr}_1)^! V \right). \\ &= (\mathrm{pr}_2)_* (D_{BG} \otimes V), \\ &= (\mathrm{pr}_2)_* \left(H_\bullet^{BM}(G) \otimes V \right), \\ &= \mathrm{inv}_{G \times 1} \left(H_\bullet(G) \otimes V \right), \\ &= H_\bullet(G) \otimes \mathrm{inv}_G(V). \end{aligned}$$

Remark 5.6.2. The identification of $(\mathrm{pr}_2)_*$ with taking G-invariants involves identifying $\mathsf{IndCoh}(BG^2)$ with $\mathsf{Rep}(G^2)$, but this identification doesn't preserve pushforwards - there's a discrepancy which involves a shift, which explains the change from $H_\bullet^{BM}(G)$ to $H_\bullet(G)$.

So if we want a module for Diff_{BG}, we need a map $\mu_V : \mathrm{Diff}_{BG}(V) \to V$ which makes the following diagram commute:

$$\begin{array}{ccc} V & \longrightarrow & \mathrm{Diff}_{BG}(V) \\ & \overset{\mathrm{id}}{\searrow} & \downarrow \mu_V \\ & & V \end{array}$$

(The top map comes from adjunction.) But we computed that $\mathrm{Diff}_{BG}(V) = H_\bullet(G) \otimes \mathrm{inv}_G(V)$, hence kills every component with a nontrivial G-action. So actually **a module for $\mathbf{Diff}_{BG}$ is a *trivial* G-representation**. This gives us the result that

$$D_{BG}{-}\mathsf{mod}(G{-}\mathsf{rep}) \simeq \mathrm{Diff}_{BG}{-}\mathsf{mod}(G{-}\mathsf{rep}) \simeq H_\bullet(G){-}\mathsf{mod}.$$

Remark 5.6.3. It may seem strange that a D-module on BG, which should just be a G-representation with an action by D_{BG}, never actually has any G-action. This is essentially due to the interpretation of D-modules as quasicoherent sheaves with flat connection - the flat connection part essentially forces the G-action to be trivial.

References

[DG13] Vladimir Drinfeld and Dennis Gaitsgory. On some finiteness questions for algebraic stacks. *Geometric and Functional Analysis*, 23(1):149–294, 2013.

[GR17] Dennis Gaitsgory and Nick Rozenblyum. A study in derived algebraic geometry: Volume ii: Deformations, lie theory and formal geometry. *Mathematical surveys and monographs*, 221, 2017.

[Lur09] Jacob Lurie. Notes on crystals and algebraic d-modules. *Available on homepage of the author*, 2009.