

BASIC MODULI THEORY

YURI J. F. SULYMA

Slogan 0.1.

Groupoids + Sites = Stacks

1. GROUPOIDS

Definition 1.1. Let G be a discrete group acting on a set X . Let $X//G$ be the category with

- **objects** the elements of X ;
- **arrows** $x \rightarrow y$ the elements of g such that $g \cdot x = y$.

$X//G$ is a groupoid, i.e. all arrows in the category $X//G$ are isomorphisms.

We have that

$$\begin{aligned}\pi_0(X//G) &= X/G \\ \pi_1(X//G, x) &= \text{stabilizer of } x\end{aligned}$$

(If B is a groupoid, $\pi_0(B)$ means the path components/isomorphism classes of B , while $\pi_1(B, b)$ just means the automorphism group of $b \in B$.)

In particular,

$$\begin{aligned}X \text{ is nonempty and transitive} &\implies \pi_0(X//G) = 0 \\ X \text{ is free} &\implies \pi_1(X//G, x) = 0 \text{ for all } x\end{aligned}$$

Thus, $X//G$ looks like the orbit set X/G “at the level of homotopy”, but still remembers information about automorphisms. In fact we can recover X (with its G -action) from $X//G$:

$$X = \lim_{\substack{\longrightarrow \\ X//G}} (\text{regular representation of } G)^\dagger$$

so really $X//G$ doesn't forget anything at all.

Slogan 1.2. Stacky quotients don't forget information, they just reorganize it in a useful way.

Notice that the groupoid takes the form

$$\begin{array}{ccc} \text{arrows} & & G \times X \\ & & \downarrow \downarrow \\ \text{objects} & & X \end{array}$$

i.e. $\{\text{arrows}\} = \{\text{objects}\} \times \{\text{something}\}$.

Definition 1.3. Ravenel calls a groupoid *split* when this happens (although I've never heard anyone else have a name for it). Anyway, it basically means that the symmetries out of any object look the same, or is equivalent to a groupoid coming from the construction above.

All the examples we've seen so far have had this form. At the Hopf algebroid level, this means that our Hopf algebroids (A, Γ) have all had the form $\Gamma = A \otimes_{\mathbb{Z}} (\text{something})$. Now when we have

[†]This is a categorical triviality known as the *co-Yoneda lemma*, which is just an abstract-nonsensical version of saying that every module is a quotient of a free module.

a group G acting on an object X in some category (e.g. this could be a topological group, a group scheme, ...), we'll define $X//G$ to be the groupoid object

$$\left. \begin{array}{c} G \times X \\ \downarrow \downarrow \\ X \end{array} \right\} X//G$$

Warning 1.4. Once we define stacks, we'll have a machine where you feed in a groupoid object and it spits out a stack. By $X//G$ I really mean the stack you get out of this machine when you feed in the above groupoid object.

Example 1.5. On Tuesday, Carolyn showed us how to construct the “moduli space of elliptic curves (over \mathbb{C})” as $\mathbb{H}/SL(2, \mathbb{Z})$. If we instead take the stacky quotient,

$$\left. \begin{array}{c} SL(2, \mathbb{Z}) \times \mathbb{H} \\ \downarrow \downarrow \\ \mathbb{H} \end{array} \right\} \mathbb{H}//SL(2, \mathbb{Z})$$

we get the moduli *stack* of elliptic curves over \mathbb{C} . So once we define the moduli stack of elliptic curves \mathcal{M}_{ell} , as well as the pullback of stacks, we'll have

$$\mathbb{H}//SL(2, \mathbb{Z}) = \mathcal{M}_{\text{ell}} \times \text{Spec } \mathbb{C}$$

(This statement is derived from intuition; I've been informed it's true, but it takes a lot of work to make sense of.)

Example 1.6. When G acts trivially on $*$, the quotient $*//G$ is called BG . When G acts on itself via the regular representation, the quotient $G//G$ is called EG . Note that EG is *contractible* in the sense that it is equivalent to the trivial groupoid $*$ (because its π_0 and π_1 vanish). Once we define the pullback of stacks we'll be able to say that we have a “fiber sequence”

$$G \rightarrow EG \rightarrow BG$$

in the sense that there is a pullback diagram

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

BG is very useful because it represents principal G -bundles, in the sense that maps

$$X \xrightarrow{\xi} BG$$

correspond to principal G -bundles on X , and the trivialization of such a bundle ξ correspond to lifts

$$\begin{array}{ccc} & & EG \\ & \nearrow & \downarrow \\ X & \xrightarrow{\xi} & BG \end{array}$$

When $G = GL(n)$ (or $U(n)$, whatever), this is equivalent to classifying n -plane bundles. **In particular, $B\mathbb{G}_m$ classifies line bundles. We'll use this.**

Definition 1.7. We may as well define the pullback of categories here.

$$\begin{array}{ccc} \mathcal{P} = \mathcal{X} \times_{\mathcal{Z}} \mathcal{Y} & \longrightarrow & \mathcal{Y} \\ \downarrow & \lrcorner & \downarrow g \\ \mathcal{X} & \xrightarrow{f} & \mathcal{Z} \end{array}$$

Normally the pullback would be defined by

$$\mathcal{P} = \{x \in \mathcal{X}, y \in \mathcal{X} \mid fx = gy \text{ in } \mathcal{Z}\},$$

but now these things are living in categories, not sets. What does it mean for two things in a category to be equal? If I tell you that two things in a category are equal, like $\pi_1 S^1 = \mathbb{Z}$, what that means is that I have an isomorphism between them. So the categorical pullback (or 2-pullback) is

$$\begin{aligned} \mathcal{P} &= \{x \in \mathcal{X}, y \in \mathcal{Y} \mid \cancel{fx = gy} \text{ in } \mathcal{Z}\} \\ &= \{x \in \mathcal{X}, y \in \mathcal{Y}, \phi: fx \cong gy \text{ in } \mathcal{Z}\} \end{aligned}$$

(note that we have to *specify* the isomorphism ϕ , not just assert that it exists).

Example 1.8. The following diagram is a pullback

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

where G is a discrete monoidal category and BG has a single object. We earlier got the same pullback when one of the $*$ s was replaced by EG ; the diagrams are equivalent because $EG \simeq *$ (we often write $EG = *$). More generally, when c and d are objects in a category \mathcal{C} , we have pullback diagrams

$$\begin{array}{ccc} \{\text{isos } c \cong d\} & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow d \\ * & \xrightarrow{c} & \mathcal{C}. \end{array} \qquad \begin{array}{ccc} \text{Aut}_e(c) & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow c \\ * & \xrightarrow{c} & \mathcal{C}. \end{array}$$

2. SITES

Many of the constructions we use have “extra functoriality”, in that they can be assembled from compatible local data in much the same way we can assemble a global section of a sheaf from compatible local sections over an open cover. We want to pin this behavior down and study it, so we’re going to want to define sheaves (and eventually stacks) on categories, instead of just on spaces.

Fix a category \mathcal{C} . It’s easy to generalize the idea of presheaf: a presheaf on X is just a functor $\mathcal{O}(X)^{\text{op}} \rightarrow \mathbf{Set}$, so we define

Definition 2.1. A *presheaf* on \mathcal{C} is a functor $P: \mathcal{C}^{\text{op}} \rightarrow \mathbf{Set}$.

A presheaf P on a space X is a *sheaf* when it is determined by local data, in the following sense: if $\{U_i\}$ is a cover of a space X , then a section of P over X is equivalent to the data of compatible sections of P over each U_i . I can represent the open sets $U_i \subset X$ as a family of maps $\{U_i \hookrightarrow X\}$. In order to generalize the notion of sheaf to categories, I want a notion of when a family of maps $\{U_i \rightarrow C\}$ is a “cover” of C .

Definition 2.2. Let \mathcal{C} be a category. Assume that \mathcal{C} has pullbacks (this assumption isn’t necessary, but some definitions get messier without it). A *Grothendieck topology* is a function J which to each $c \in \mathcal{C}$ assigns a collection $J(c)$ of families of maps into c which we call “covers” (or J -covers).

idea	for spaces	for sites
presheaf	$P: \mathcal{O}(X)^{\text{op}} \rightarrow \mathbf{Set}$	$P: \mathcal{C}^{\text{op}} \rightarrow \mathbf{Set}$
everything covers itself	$\{X\}$ covers X	if $f: D \cong C$, then $\{f\}$ covers C
cover the whole thing \implies cover all smaller things	$\{U_i\}$ covers $X \implies \{U_i \cap V\}$ covers V for any $V \subset X$	if $\{U_i \rightarrow C\}$ covers C , then $\{U_i \times_C D \rightarrow D\}$ covers D for any $D \xrightarrow{f} C$
cover all the pieces \implies cover the whole thing	if $\{U_i\}_i$ covers U , and each U_i is covered by $\{V_{ij}\}_j$, then $\{V_{ij}\}_{i,j}$ covers U	if $\{U_i \rightarrow C\}$ covers C , and $\{V_{ij} \rightarrow U_i\}$ covers each U_i , then $\{V_{ij} \rightarrow C\}$ covers C

A *site* is a pair (\mathcal{C}, J) consist of a category \mathcal{C} and a Grothendieck topology J on it.

Remark 2.3. If I have a cover $\{U_i \rightarrow C\}$, I can throw in extra junk $\{V_j \rightarrow C\}$ and the union $\{U_i \rightarrow C\} \cup \{V_j \rightarrow C\}$ will still be a cover. So when I tell you that the covers in a site are such-and-such families of maps, I mean that the covers are families of maps which contain such-and-such families of maps.

Example 2.4. Here are some examples of sites.

- $\mathcal{C} = \mathcal{O}(X)$; a collection $\{U_i \rightarrow U\}$ is a cover if $\bigcup U_i = U$.
- $\mathcal{C} = \{\text{topological spaces}\}$; a cover of X is a family $\{U_i \hookrightarrow X\}$ of inclusions of open subspaces, which form a cover in the ordinary sense. This is a “global” version of the preceding example.
- $\mathcal{C} = \{\text{finitely presented } \mathbb{Z}\text{-algebras}\}^{\text{op}}$; a cover is a family of maps $\{\text{Spec}(R[f_i^{-1}]) \rightarrow \text{Spec}(R)\}$ such that the f_i generate the unit ideal of R . (In other words, a cover by distinguished open affines.) This is called the *big Zariski site*, and is the algebro-geometric analogue of the preceding example.
- $\mathcal{C} = \{\text{topological spaces}\}$; a cover of X is a family $\{U_i \rightarrow X\}$ whose *image* is an open cover of X , and for which each U_i is a *covering space* of its image. In this case the “open subsets” have automorphisms! This analogous to the *étale site* in algebraic geometry (which Artur will tell you about)
- Let $(X, \mathcal{O}_X) \xrightarrow{f} (Y, \mathcal{O}_Y)$ be a map between locally ringed spaces. We say f is *flat* if each local ring $\mathcal{O}_{X,x}$ is flat over $\mathcal{O}_{Y,f(x)}$. It is *faithfully flat* if it is flat and surjective. We declare a map to be a cover if it is faithfully flat (+ some finiteness conditions that I won’t get into).

When we specialize to schemes, we get the *flat topology*. Actually there are two different versions, the *fqc* topology and the *fppf* topology. The qc and pf are finiteness conditions (quasicompact/finite presentation) which are automatic in the affine case, which is all we really care about, so we’ll just talk about the flat topology.

The flat and étale topologies are usually what we’ll be interested in here.

Here is a nice classical example of faithfully flat covers for general locally ringed spaces (not schemes).

Example 2.5 (GAGA, c.f. [3]). Let X be a complex algebraic variety (I think this is something both groups of people are relatively comfortable with). We can view X as a locally ringed space in two ways

- $(X^{\text{an}}, \mathcal{H})$ in the analytic topology with the sheaf of holomorphic functions
- $(X^{\text{Zar}}, \mathcal{O})$ in the Zariski topology with the sheaf of regular functions (here I mean the naïve Zariski topology, with no schemey generic points)

There is a map of locally ringed spaces

$$(X^{\text{an}}, \mathcal{H}) \longrightarrow (X^{\text{Zar}}, \mathcal{O})$$

which forgets the analytic structure of X and includes polynomials into analytic functions. *This is a flat map of locally ringed spaces*[†]. It is trivially surjective, so it is actually faithfully flat. The topological sense in which this map is a “cover” is trivial; but it’s also telling us, loosely speaking, that “our knowledge of analytic functions covers (includes) our knowledge of polynomials”.

There is a functor

$$\mathrm{Coh}(\mathcal{O}) \longrightarrow \mathrm{Coh}(\mathcal{H})$$

between categories of sheaves of coherent modules. *When X is projective, this is an equivalence.* We can interpret this as a descent result: given a coherent sheaf on the cover $(X^{\mathrm{an}}, \mathcal{H})$, we can *descend* it to a coherent sheaf on $(X^{\mathrm{Zar}}, \mathcal{O})$ (or we might say that the functor Coh *satisfies descent* for the cover $(X^{\mathrm{an}}, \mathcal{H}) \rightarrow (X^{\mathrm{Zar}}, \mathcal{O})$). This is false when X is not projective (it is not even true for affine varieties), demonstrating that descent questions can be very non-trivial and interesting.

Definition 2.6. Let (\mathcal{C}, J) be a site, and let $F: \mathcal{C}^{\mathrm{op}} \rightarrow \mathbf{Set}$ be a presheaf. We say that F is a *sheaf* with respect to the topology J if

$$F(X) \longrightarrow \prod_i F(U_i) \rightrightarrows \prod_{i,j} F(U_{ij})$$

where $U_{ij} = U_i \times_X U_j$

is an equalizer diagram for every J -covering family $\{U_i \rightarrow X\}$ of X . We write $\mathrm{Sh}(\mathcal{C}, J)$ for the full subcategory of $\mathbf{Set}^{\mathcal{C}^{\mathrm{op}}}$ consisting of sheaves.

Just like ordinary sheaves, we can sheafify a presheaf, which is a good source of sheaves. The construction is a bit involved, but the upshot is that we get an adjunction

$$\mathbf{Set}^{\mathcal{C}^{\mathrm{op}}} \begin{array}{c} \xrightarrow{\text{sheafify}} \\ \perp \\ \xleftarrow{\text{inclusion}} \end{array} \mathrm{Sh}(\mathcal{C}, J)$$

Definition 2.7. A *Grothendieck topos* is a category equivalent to $\mathrm{Sh}(\mathcal{C}, J)$ for some site (\mathcal{C}, J) . Note that different sites can give rise to the same topos.

3. STACKS

Definition 3.1. Let (\mathcal{C}, J) be a site. We’ll call a functor $\mathcal{M}: \mathcal{C}^{\mathrm{op}} \rightarrow \mathfrak{Gpd}$ from $\mathcal{C}^{\mathrm{op}}$ to the 2-category of groupoids a *prestack*. We say that \mathcal{M} is a *stack* with respect to the topology J if

$$\mathcal{M}(X) \longrightarrow \prod_i \mathcal{M}(U_i) \rightrightarrows \prod_{i,j} \mathcal{M}(U_{ij}) \rightrightarrows \prod_{i,j,k} \mathcal{M}(U_{ijk})$$

is a (2-)limit diagram for every J -covering family $\{U_i \rightarrow X\}$, for every object $X \in \mathcal{C}$.

Remark 3.2. The diagram above is a truncated bar construction. For sheaves, we were able to chop it off at double intersections, while for stacks we need to consider triple intersections. As one might expect, if we want to talk about ∞ -stacks (for example, to discuss sheaves of E_∞ ring spectra), one needs to consider the entire simplicial object.

Let’s unwrap this a bit.

Definition 3.3. Let $\{U_i \rightarrow X\}$ be a J -cover of X . The *category of descent data* $\mathfrak{Desc}(U_i, \mathcal{M})$ has

- **objects** collections of objects $E_i \in \mathcal{M}(U_i)$, together with isomorphisms $\phi_{ij}: E_i|_{U_{ij}} \xrightarrow{\sim} E_j|_{U_{ij}}$ satisfying the cocycle condition $\phi_{jk}\phi_{ij} = \phi_{ik}$

[†]Proof: the question is local, so it suffices to treat $X = \mathbb{C}^n$. The local ring of \mathcal{O} is then $\mathbb{C}[z_1, \dots, z_n]$, while the local ring of \mathcal{H} is the ring of germs of holomorphic functions. These have the same completion $\mathbb{C}[[z_1, \dots, z_n]]$, which turns out to imply that the original map between local rings is faithfully flat.

- **arrows** $(E_i, \phi_{ij}) \rightarrow (F_i, \psi_{ij})$ collections of maps $E_i \xrightarrow{f_i} F_i$ in $\mathcal{M}(U_i)$, giving commutative diagrams

$$\begin{array}{ccc} E_i|_{ij} & \xrightarrow{f_i} & F_i|_{ij} \\ \phi_{ij} \downarrow & & \downarrow \psi_{ij} \\ E_j|_{ij} & \xrightarrow{f_j} & F_j|_{ij} \end{array}$$

In other words, $\mathfrak{Desc}(U_i, \mathcal{M})$ consists of “the things I’d want to be able to glue together into a global section $E \in \mathcal{M}(X)$ ”. By definition it is the limit of the diagram

$$\mathfrak{Desc}(U_i, \mathcal{M}) \longrightarrow \prod_i \mathcal{M}(U_i) \rightrightarrows \prod_{i,j} \mathcal{M}(U_{ij}) \rightrightarrows \prod_{i,j,k} \mathcal{M}(U_{ijk})$$

and the stack condition is equivalent to requiring that $\mathcal{M}(X) \rightarrow \mathfrak{Desc}(U_i, \mathcal{M})$ be an equivalence of groupoids for every cover $\{U_i \rightarrow X\}$ of every object $X \in \mathcal{C}$.

Warning 3.4. Lots of sources talk about stacks in terms of “categories (co)fibered in groupoids” instead of functors into \mathfrak{Spd} . Ask me about that afterwards if you want to know more.

Example 3.5. Consider the category of topological spaces \mathbf{Top} with the usual Grothendieck topology. Then the functors

$$\begin{aligned} \mathit{Sh}: \mathbf{Top}^{\text{op}} &\rightarrow \mathfrak{Spd}, & X &\mapsto \{\text{groupoid of sheaves on } X\} \\ \mathit{Vect}: \mathbf{Top}^{\text{op}} &\rightarrow \mathfrak{Spd}, & X &\mapsto \{\text{groupoid of vector bundles on } X\} \end{aligned}$$

are stacks. Explicitly, this means that given

- a topological space X ;
- an open cover $\{U_i \subset X\}$ of U ;
- sheaves (or vector bundles) F_i on U_i ;
- isomorphisms $\phi_{ij}: F_i|_{U_{ij}} \xrightarrow{\sim} F_j|_{U_{ij}}$;
- satisfying the *cocycle condition* $\phi_{jk}\phi_{ij} = \phi_{ik}$ on U_{ijk} ,

there exists a sheaf (or vector bundle) F on X

- such that $F_i \cong F|_{U_i}$;
- which is unique to unique isomorphism;
- and probably satisfies some other pleasant compatibility conditions.

Just as with sheaves, we frequently obtain stacks by producing a prestack and then stackifying this.

Definition 3.6. Let \mathcal{M} be a prestack. A *stackification* of $\tilde{\mathcal{M}}$ such that any map from \mathcal{M} to a stack \mathcal{N} factors uniquely through $\tilde{\mathcal{M}}$:

$$\begin{array}{ccc} & \tilde{\mathcal{M}} \text{ stack} & \\ & \nearrow & \downarrow \\ \text{prestack } \mathcal{M} & & \downarrow \exists! \\ & \searrow & \downarrow \\ & \mathcal{N} \text{ stack} & \end{array}$$

This gives us nice formal properties, but we’re going to need to know how to actually calculate stackification. We can define it by

$$\tilde{\mathcal{M}}(X) = \varinjlim \mathfrak{Desc}(U_i, \mathcal{M})$$

where the colimit is taken over all coverings $\{U_i \rightarrow X\}$ of X .

Warning 3.7. For this to work, X needs to have a small cofinal collection of coverings. This causes problems with the fpqc site; see [4, Tag 03NV] for discussion.

Warning 3.8. There’s some confusion in the literature as to whether this actually works, or if we need to use “hypercovers”; see [2, §6.5.3] for more discussion. It probably doesn’t make any difference to us.

Earlier, we advertised a machine for turning groupoids into stacks. It is time to deliver the product.

Definition 3.9. Let $B = (B_0, B_1)$ be a groupoid in \mathcal{C} . (For us, \mathcal{C} will be the category of schemes, and $(B_0, B_1) = (\text{Spec } A, \text{Spec } \Gamma)$ will usually come from a Hopf algebroid.) We obtain a stack \mathcal{B} as follows. B determines a prestack $\mathcal{C}^{\text{op}} \rightarrow \mathfrak{Gpd}$ by

$$C \in \mathcal{C} \mapsto \text{Hom}(C, B) = \begin{cases} \text{Hom}(C, B_1) \\ \downarrow \downarrow \\ \text{Hom}(C, B_0) \end{cases}$$

and we define \mathcal{B} to be the stackification of this prestack.

For example, we showed earlier how an action of a group object G on an object X gave rise to a groupoid, which we provisionally called “ $X//G$ ” with a caveat. We now define $X//G$ to be the stack obtained by feeding said groupoid into the machine just introduced.

Definition 3.10. Recall that the affine scheme $\text{Spec } L$ classifies formal group laws, and is acted on by the group of *isomorphisms* $G^+ = \text{Spec } \mathbb{Z}[b_0^{\pm 1}, b_1, \dots]$ as well as the restricted group of *strict isomorphisms* $G = \text{Spec } \mathbb{Z}[b_1, b_2, \dots]$.

The *moduli stack of formal group laws and strict isomorphisms* is

$$\mathcal{M}_{\text{fg}}^{\text{s}} := \text{Spec}(L)//G,$$

while the *moduli stack of formal groups and isomorphisms* is

$$\mathcal{M}_{\text{fg}} := \text{Spec}(L)//G^+.$$

When we build $\mathcal{M}_{\text{fg}}^{\text{s}}$, we do not need to stackify: that is, formal group *laws* can be glued together along *strict* isomorphisms. On the other hand, we cannot in general patch together formal group *laws* along *non-strict* isomorphisms: we obtain a formal group instead. Thus we do need to stackify when forming \mathcal{M}_{fg} .

Warning 3.11. Jacob Lurie has no use for non-stacky quotients, and in his chromatic homotopy theory notes [1], writes $\text{Spec}(L)/G$ (resp. $\text{Spec}(L)/G^+$) for what we have called $\text{Spec}(L)//G$ (resp. $\text{Spec}(L)//G^+$).

Definition 3.12. We want to define the pullback of a diagram of stacks.

$$\begin{array}{ccc} \mathcal{P} = \mathcal{X} \times_{\mathcal{Z}} \mathcal{Y} & \longrightarrow & \mathcal{Y} \\ \downarrow & \lrcorner & \downarrow g \\ \mathcal{X} & \xrightarrow{f} & \mathcal{Z} \end{array}$$

This is defined pointwise in terms of the pullback of categories,

$$\mathcal{P}(R) = \mathcal{X}(R) \times_{\mathcal{Z}(R)} \mathcal{Y}(R).$$

Remember that this means

$$\mathcal{P}(R) = \{x \in \mathcal{X}(R), y \in \mathcal{Y}(R), \phi: fx \cong gy \text{ in } \mathcal{Z}(R).\}$$

Example 3.13. Taking a product $\mathcal{M} \times \text{Spec } R = \mathcal{M} \times_{\text{Spec } \mathbb{Z}} \text{Spec } R$ has the effect of restricting to R -algebras. For example, over a \mathbb{Q} -algebra R ,

every formal group law is isomorphic to \mathbb{G}_a	we have seen that	which can be expressed in stacky language as
		$\mathcal{M}_{\text{fg}}^{\text{s}} \times \text{Spec } \mathbb{Q} = \text{Spec } \mathbb{Q}$

the endomorphism ring of \mathbb{G}_a is $R[b_0]$,	$\mathcal{M}_{\text{fg}} \times \text{Spec } \mathbb{Q} = B\mathbb{G}_m \times \text{Spec } \mathbb{Q}$
hence $\text{Aut}(\mathbb{G}_a) = \text{Spec } R[b_0^{\pm 1}] = \mathbb{G}_m$	

Example 3.14. (This is from the problem set. If you wanted to try it yourself, cover your ears and look away until you sense that we've finished discussing it.) We're asked to show that there is a stack map

$$\mathcal{M}_{\mathfrak{g}} \longrightarrow B\mathbb{G}_m$$

and to identify the fiber

$$\begin{array}{ccc} \mathcal{P} & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow \\ \mathcal{M}_{\mathfrak{g}} & \longrightarrow & B\mathbb{G}_m \end{array}$$

Given a formal group $\mathrm{Spec}(R) \xrightarrow{G} \mathcal{M}_{\mathfrak{g}}$, one can define a line bundle (locally free module) \mathfrak{g} over R , called its *Lie algebra* (c.f. [1, Lecture 11]). This defines the map $\mathcal{M}_{\mathfrak{g}} \rightarrow B\mathbb{G}_m$ (one could also send G to \mathfrak{g}^{-1} , it doesn't matter). The Lie algebra \mathfrak{g} has the property that:

- \mathfrak{g} is a trivial bundle (i.e. a free R -module) $\iff G$ is coordinatizable (comes from a formal group law)
- bundle trivializations (choices of module generator) for \mathfrak{g} correspond to choices of formal group law

The map $* \rightarrow B\mathbb{G}_m$ picks out the trivial bundle R . The pullback

$$\mathcal{P}(R) = \{G \text{ formal group over } R, F: \mathfrak{g} \cong R\}$$

consists of formal groups together with chosen trivializations of their Lie algebras. By the above remarks this means a formal group coming from a formal group law, together with a *choice* of formal group law presenting it. This is where the distinction between isomorphisms and *strict* isomorphisms comes from: requiring an isomorphism to commute with the chosen trivializations cuts out a choice of generator \mathfrak{g} , which comes from a choice of unit $b_0 \in \mathbb{G}_m$.

Thus the pullback \mathcal{P} is the moduli stack $\mathcal{M}_{\mathfrak{g}}^s$ discussed above.

$$\begin{array}{ccc} \mathcal{M}_{\mathfrak{g}}^s & \longrightarrow & * \\ \downarrow & \lrcorner & \downarrow \\ \mathcal{M}_{\mathfrak{g}} & \longrightarrow & B\mathbb{G}_m \end{array}$$

REFERENCES

- [1] Jacob Lurie, *Chromatic homotopy theory*, <http://www.math.harvard.edu/~lurie/252x.html>.
- [2] ———, *Higher topos theory*, Annals of Mathematical Studies, vol. 170, Princeton University Press, Princeton, New Jersey, 2009.
- [3] Jean-Pierre Serre, *Géométrie algébrique et géométrie analytique*, Annales de l'institut Fourier **6** (1956), 1–42.
- [4] The Stacks Project Authors, *Stacks project*, <http://stacks.math.columbia.edu>, 2016.