
Emory Math 788, Stacks
TuTh 11:30 - 12:45

28 Lectures about Stacks
Last updated: September 28, 2022

Contents

Lecture 1: introduction	2
Lecture 2: Categories	3
Lecture 3: Equivalences of categories and (co)limits	4
Lecture 4: Computing with (co)limits	5
Lecture 5: More (co)limits	6
Lecture 6: Yoneda's lemma	7
Lecture 7: More Yoneda; Sheaves and Grothendieck Topologies	8
Lecture 8:	9

Lecture 1: introduction

One thing to focus on: understand the functors attached to an affine variety (the functor of solutions, the functor of points, and why they agree).

1. A stack is a (quick definition). An algebraic stack is a (quick definition). End of course jk. A fibered category is basically a (pseudo) functor.
2. Why rings? Let k be a field and let A be finitely generated k algebra. Then maps from A to a k -algebra R are naturally in bijection with solutions to the defining equations of A . If R is a field then \ker is a max idea, if $k = \bar{k}$ then \ker is $(x_1 - a_1 \dots)$. So maximal ideals correspond to solutions to the defining equations. Better to just work with the ring up to isomorphism, rather than keeping track of the equations. Lets you generalize and apply geometric ideas elsewhere (e.g. $\text{Spec } \mathbb{Z}$ is a curve, in that it is one dimensional).
3. Why Schemes and sheaves? Glueing, limits, transferring info (complex numbers vs characteristic p), cohomology; need sheaves to really talk about functions.
4. What and why of functors. X affine variety. Solve equations. Triangle. Define S_X , the “functor of solutions”; S_X has an “injectivity” property. (Note it doesn’t have the “surjectivity” property; example: $xy = p$).
5. Functor of points h_X , define, explain. Note that even for a crazy ring (like $\widehat{\mathbb{Z}}$, or a perfectoid ring) you still have some injectivity properties. (Give a short proof.)
6. Best thing you can do in mathematics: the Missy Elliot approach (flip it and reverse it). I.e., start with the functor, i.e., the collection of sets and maps between sets, and ask if these are the solutions to some equations. This is a very useful perspective, but not obviously so.
7. Example: lines in \mathbb{P}^2 vs \mathbb{P}^2 . To make this easy, stick to fields. There is one issue: not literally the same. Just “equivalent” in some sense. This will come up a lot. [Put a picture of a soapbox here.] “The Same” is a concept with a lot of content. [Grad school: when I started, the older students where always up in arms about “equal vs isomorphic”; once I was an older student, I was too].
8. Define representable.
9. OK, so I haven’t mentioned stack in awhile. Here is a functor: R maps to curves over R . Not injective! So do we give up and go home???
10. Ideas. Replace everything with categories
11. Note: automorphisms are the problem. Twists, and $H^1(\text{Aut})$.

Lecture 2: Categories

One thing to focus on: equivalence of categories. Make sure you understand the definition, some examples, and how to verify correctness of the examples.

Terse list of what I covered: Category definition, examples (top, sch, rings, k -alg, $vect_k$; finite examples, opposite category); initial and final objects; functors, examples; group as a category, representatons as functors to vector spaces; natural transformations, examples; equivalence of categories, examples: 1 dot vs 2 dots, rings op vs affine schemes; fully faithful and essentially surjective. Enriched categories (over ab, over top, over cat).

Lecture 3: Equivalences of categories and (co)limits

One thing to focus on: verify that the description of limits and colimits of sets is correct, and verify some of the examples.

Recall equivalence of categories. Two dots example in detail. Theorem: f is a bijection if and only if it is invertible. (Note: not true for topological spaces.) Define full, faithful, essentially surjective. Theorem: F is an equivalence of categories if and only if it is fully faithful and essentially surjective.

Define Comma (or slice) categories; category of morphisms; categories with extra data. Diagrams.

Limits and colimits. Define (co)limits as (final) initial objects. Examples. Explicit construction for sets. Examples: Fiber products (for topological spaces, curve mapping into the base of a bundle); glueing (example of a glueing two discs along boundary circle); \mathbb{Z}_p and power series; algebra (kernels, cokernels, equalizers). Quotients. Limits tend to stay the same when you enlarge a category, colimits tend to change.

Lecture 4: Computing with (co)limits

One thing to focus on: make sure you understand why you can compute (co)limits just with (hom) sets, and do a few such computations (like the magic square computation).

Recall definition of (co)limit. Recall construction for sets. Recall kernel example, and do cokernel example. Quotient by a group. Intersections. Scheme structures on fibers ($\mathbb{A}^2 \rightarrow \mathbb{A}^1, (x, y) \mapsto x^2y^3$).

Limits and colimits via Hom sets. This part is very important for the generalization to stacks. Limits in $\text{Fun}(C, \text{Sets})$.

Magic squares. See examples from Ravi's books. This is a good example of the type of calculation you really want to be able to do. Example of a useful computation: the graph of a separated morphism is closed.

Lecture 5: More (co)limits

One thing to focus on: study the example where colimits change.

A few more magic square examples (equalizers, sections of morphisms).

Limits and colimits in the category of affine schemes, via rings; examples: fiber product of schemes; quotients by finite groups.

Limits tend to stay the same when you enlarge a category (but this needs proof in each case), colimits tend to change. Example of glueing affine schemes in Aff vs Sch . Quotients by infinite groups like \mathbb{G}_m , not always correct “on points”. Taking quotients gets worse and worse the more you do it, which is part of the point of stacks. How to take quotients by finite groups: cover by affines and see what goes wrong.

Lecture 6: Yoneda's lemma

One thing to focus on: prove the “generalized Yoneda's lemma” (for maps from representable functors to arbitrary functors) from the homework.

Functor of points, representable functors. Motivation: point mapping into a topological space; residue field of a point of a scheme. Yoneda's lemma (i.e., the “fundamental theorem of category theory”). Example of the fully faithful part (\mathbb{A}^1). Functor of points of m . Limits via functor of points. Part of the point of the functor of points is that you can solve problems by “starting with the answer” (i.e., write down the functor that you want, then ask if it is representable). Examples: well, any limit. Residue fields

Lecture 7: More Yoneda; Sheaves and Grothendieck Topologies

One thing to focus on: come to terms with writing the sheaf axiom as a limit. Think about the details of adding disjoint unions to Top^{op} , and how that doesn't change the sheaf condition.

Recall Yoneda; another example or two of functor of points. There is a co-Yoneda lemma. Not as useful; time flows forward and sets are left handed. Colimits? There is a co-Yoneda lemma. Not as useful; time flows forward and sets are left handed. For colimits, actually, also true for functors. Problem: usually won't be a sheaf, and often isn't the correct limit. (Revisit after discussing sheaves.)

Recall definition of a sheaf from Hartshorne. Reformulate sheaf axiom as a limit. Talk about disjoint unions. Reformulate via $X' \rightarrow X$. Define Grothendieck topology and site. Examples: big Zariski site; $P - Q$ sites. Useful generalities (like using Yoneda to reformulate the sheaf axiom as a *colimit*).

Adjoints and sheafification.

Lecture 8:

One thing to focus on: