MODULI PROBLEMS AND GEOMETRIC INVARIANT THEORY (GIT)

GRK Workshop WS 2022/23

Karen Martinez and David Sembowski

A key question in algebra is how to classify objects up to some notion of equivalence. Moduli problems are a special case of such problems where one seeks to find a scheme/variety such that each (k-)point represents an equivalence class. These schemes are then referred to as the moduli space of the corresponding moduli problem.

The construction of moduli spaces is, in general, a very difficult problem and sometimes not even possible. One way to address these problems is through the machinery of geometric invariant theory (GIT). GIT is a method one can use to build good quotients of algebraic group actions, where a geometrically reductive group acts on a quasi-projective (k-)scheme.

This workshop aims to present this machinery, to explain the connection between good quotients and moduli problems and, finally, to use GIT to construct a coarse moduli space for the moduli problem of vector bundles on a smooth projective curve.

Talks

Talk 1: Moduli problems and algebraic groups

This talk intends to give a description of a moduli problem and to recall briefly some notions of moduli spaces and algebraic groups.

Suggested plan

- Moduli problems: [§ 2. [1]] Recall the definition of a functor of points of a scheme. Introduce the functor $h : \mathcal{C} \to \mathrm{Psh}(\mathcal{C})$ and define a representable presheaf $F \in \mathrm{Psh}(\mathcal{C})$. Comment in the following question: Is every presheaf in Psh(Sch) representable by a scheme?.[1]

  Explain the main ingredients of a moduli problem and, in particular, present Definition 2.8, (two) Examples 2.9, Definition 2.10 and Lemma 2.11[2].

  Give the definitions of fine moduli spaces, universal families and coarse moduli spaces and, discuss why we need to introduce this last notion (for this, have a look at section 2.4 [1]).

- Algebraic groups: [§ 3. [1]] Define (affine) algebraic groups, linear representations of an affine linear group and (algebraic) actions of an affine algebraic group on a scheme. Moreover, introduce the notion of character groups, fix notation for orbits and stabilizers and state Proposition 3.20.

[1] Focus in the interest of exploring it for moduli functors.

[2] Ignore the rest of [Section 2.]

Talk 2: Notions of quotients and the affine GIT one

The aim of this talk is to present different notions of quotients and, in particular, to construct the GIT quotient making use of Nagata's theorem.

Suggested plan

- Notions of quotients: [§ 3.4 and 3.5 [1]] Give the definition of a categorical quotient and comment on why we are interested in this notion, in particular, introduce Proposition 3.35. Moreover, define good quotients, present Proposition 3.30[3] and remark that any good quotient is a categorical quotient.

- Affine GIT quotients: [§ 4. [1]] Let G be an affine algebraic group. Introduce in details the ring of G-invariant regular functions and Hilbert's 14th problem. Give an example of a finitely generated invariant ring and state that there are examples in which it is non-finitely generated (see [3], by Nagata). Define geometrically reductive groups and comment that there are other notions of reductivity which are more often used, namely reductive groups and linearly reductive groups. Moreover, discuss about the relationship between these different notions, like in Theorem 4.16. Present an example of a geometrically reductive group (see [2]) and also Remark 4.17. State Nagata's solution for Hilbert's problem, namely Theorem 4.20. Define affine GIT quotients and, state Theorem 4.30 and Corollary 4.31.

Talk 3: (Quasi-)projective GIT quotients

In this talk we will learn the notions and techniques that are necessary to construct a quasi-projective GIT quotient and, in particular, we will apply these techniques presenting (in a detailed example) a projective GIT quotient. Reference [§ 5. [1]].

Suggested plan

State that the (affine) GIT quotient cannot be glued together and therefore we need to introduce a notion of linearization for making it possible. Shortly, present the Definition 5.1 and give a description on how a rational morphism of projective schemes is defined.

Define semistable and stable sets[4] and, state Theorem 5.3 and Theorem 5.6 as one fact. Moreover, define what a linearization is and summarize the main points of section 5.5. GIT for general varieties with linearisations[5]. Finally, explain in details Example 5.8 (the projective case).

---

[3]Give an idea of the proof if time allows.

[4]Forget the set of unstable points.

[5]Leave out the proof of Theorem 5.31.

Talk 4: (Coarse) moduli space of vector bundles over a smooth projective curve

The goal of this talk is to present an example of a moduli problem and an approach to it using (quasi-)projective GIT quotients. The main reference for this talk is [Chapter 5. [4]], by Newstead.

Suggested plan

Present the moduli problem of classifying vector bundles over an algebraic curves with a focus on the case of bundles over a non-singular curve. Use the Riemann-Roch formula to define the degree of a sheaf and, in particular, deduce the formula of the degree of a line bundle.

Give the definition of (semi-)stable vector bundles, present Lemma 5.1[6], Lemma 5.2, introduce the "set" R (see pag. 110) as an open set and state Theorem 5.3.

Present the crucial arguments on the construction of the quotient [Chapter 5. §4 [4]]. Note that the aim is to reach to Theorem 5.8, if necessary, you can state Theorem 5.6 without proving it. Moreover, if time permits, comment that under certain conditions the obtained coarse moduli space is fine.

References

[1] V. Hoskins. *Moduli Problems and Geometric Invariant Theory*. 2015.

[2] S. Mukai. *An introduction to invariants and moduli, Cambridge studies in advanced mathematics*. Cambridge University Press, 2003.

[3] M. Nagata. *On the 14th problem of Hilbert*. Amer. J. Math. 81 (1959), 766-772.

[4] P.E. Newstead. *Introduction to Moduli Problems and Orbit Spaces*. Tata Inst. Lectures, Springer 1978.

[6] If time permits, sketch the proof.