REPRESENTATION THEORY (WITH LEAN!)

ATTICUS KUHN

Date: March 19, 2023.

Contents

1. Introduction

In this paper, I will be giving an introduction to representation theory. The basic idea of representation theory is to use knowledge about linear algebra to "provide structure" to groups so we can learn more about groups. I will assume the reader knows some basic group theory and linear algebra.

Additionally, I will be using the computer theorem verifier **Lean** from Microsoft Research to formally verify what we are doing. I will intersperse pieces of Lean code in between mathematical notation. The benefits of Lean are that it forces clarity into proofs and definitions because the computer will only accept a very rigorours argument.

2. Basics of Representations

Definition 1. *Given a vector space V and a group G, we say that $\rho : G \to V$ is a* ***representation*** *of G if ρ is a group homomorphism from G to $GL(V)$. In other words, this means that*

$$\rho : G \to Gl(V)$$
$$\forall g_1, g_2 \in G \qquad \rho(g_1 g_2) = \rho(g_1)\rho(g_2)$$

```
variables (k G V : Type*) [comm_semiring k] [monoid G]
  [add_comm_monoid V] [module k V]

abbreviation representation := G →* (V →ₗ[k] V)
variables (ρ : representation k G V)
```

2.1. Examples of Representations.

In order to give some intuition about how representations typically look, we will give 5 different examples of representations.

The first example of a representation is the simplest: it is called the trivial representation.

2.1.1. *Trivial Representation.* The trivial representation of G is given by

$$\forall g \in G \qquad \rho(g) = Id$$

```
def trivial : representation k G k := 1
```

It turns out that some operations on vector spaces (such as tensor product, direct sum, and dual) also can work on representations, and they also preseve the homomorphism property.

2.1.2. *Tensor Product.* Given two representations $\rho_V : G \to Gl(V)$ and $\rho_W G \to Gl(W)$, we define the **tensor product** $\rho_V \otimes \rho_W : G \to GL(V \otimes W)$ as

$$\rho_V \otimes \rho_W(g) := \rho_V \otimes \rho_W$$

```
def tprod : representation k G (V ⊗[k] W) :=
{ to_fun := λ g, tensor_product.map (ρV g) (ρW g),
  map_one' := by simp only [map_one, tensor_product.map_one],
  map_mul' := λ g h, by simp only [map_mul,
    tensor_product.map_mul] }
local notation ρV ‘ ⊗ ‘ ρW := tprod ρV ρW
```

The direct sum is the dual of the tensor product, and its definition is similar.

2.1.3. *Direct Sum.* Given some indexed representations $\forall i \in I, \rho_i : G \to GL(V_i)$, we can construct a new representation from the direct sum, which is defined as

$$\bigoplus_{i \in I} \rho_i : G \to GL\left(\bigoplus_{i \in I} V_i\right)$$

$$\left(\bigoplus_{i \in I} \rho_i\right)(g) := \bigoplus_{i \in I} \rho_i(g)$$

```
def direct_sum_of_representations (i :Type) (M : i → Type)
  [decidable_eq i]
  [Π j, add_comm_monoid (M j)] [Π j, module k (M j)]
  (r : Π j, representation k G (M j)) :
  representation k G (direct_sum i (λ (j : i), M j)) :=
dfinsupp.map_range.module_End.comp   (pi.monoid_hom r)
```

2.1.4. *Dual Representation.* Just like how we can take the dual of a vector space, we can take the dual of the result of a represnetation to produce a new representation. Given a representation ρ, we can construct another representation ρ^* called the **dual representation** defined by

$$\forall g \in G \qquad \rho^*(g) := \rho(g^{-1})^T$$

```
def dual : representation k G (module.dual k V) :=
{ to_fun := λ g,
  { to_fun := λ f, f ∘ₗ (ρV g⁻¹),
    map_add' := λ f₁ f₂, by simp only [add_comp],
    map_smul' := λ r f,
      by {ext, simp only [coe_comp, function.comp_app,
   smul_apply, ring_hom.id_apply]} },
  map_one' :=
    by {ext, simp only [coe_comp, function.comp_app, map_one,
   inv_one, coe_mk, one_apply]},
  map_mul' := λ g h,
    by {ext, simp only [coe_comp, function.comp_app,
   mul_inv_rev, map_mul, coe_mk, mul_apply]}}
```

Tensor product, direct sum, and dual are 3 examples of vector space operations that can be lifted to operations over representations.

The last example given is for Linear Homomorphism representation. It is a fact from linear algebra that the set of linear maps between V and W forms a vector space. This example is a bit more complicated than the previous examples because it doesn't just work point-wise over the output of the representation.

2.1.5. *Linear Homomorphism Representation.* Given 2 representations $\rho_v : G \to Gl(V)$ and $\rho_w : G \to GL(W)$, we can create a new representation $\rho_{VW} : G \to GL(V \to W)$ by

$$\forall g \in G \qquad \rho_{vw}(g, f) = \rho_v(g) \circ f \circ \rho_w(g^{-1})$$

.

```
def lin_hom : representation k G (V →ₗ[k] W) :=
{ to_fun := λ g,
  { to_fun := λ f, (ρW g) ∘ₗ f ∘ₗ (ρV g⁻¹),
    map_add' := λ f₁ f₂, by simp_rw [add_comp, comp_add],
    map_smul' := λ r f, by simp_rw [ring_hom.id_apply,
   smul_comp, comp_smul]},
  map_one' := linear_map.ext $ λ x,
```

```
    by simp_rw [coe_mk, inv_one, map_one, one_apply,
   one_eq_id, comp_id, id_comp],
  map_mul' := λ g h,  linear_map.ext $ λ x,
    by simp_rw [linear_map.coe_mul, coe_mk,
   function.comp_apply, mul_inv_rev, map_mul, mul_eq_comp,
                  comp_assoc ]}
```

With these examples of how representations work, we can move onto an interesting theorem about representations.

3. Mashcke's Theorem

Just like we can factor a natural number into primes, we can break down representations into sub-representations.

Definition 2. ***Sub-representation***: *Given a representation* $\rho : G \to GL(V)$*, we say that* $\pi : G \to GL(W)$ *is a sub-representation of* ρ *if* W *is a vector subspace of* V *and* W *is g-invariant, which means that*

$$\forall g \in G, \forall w \in W \qquad \pi(g)(w) \in W$$

We will now cover a famous theorem in representation theory: Mashcke's Theorem.

Theorem 3. ***Maschkes Theorem***: *Given a group* G *and a representation* V *of* G *over a finite field* $\mathbb{F}$ *such that char*$\mathbb{F} = 0$*, if* U *is a sub-representation of* V*, then there exists a sub-representation* W *such that*

$$V = W \oplus U$$

```
lemma exists_is_compl
  (p : submodule (monoid_algebra k G) V) :
  ∃ q : submodule (monoid_algebra k G) V, is_compl p q :=
let ⟨f, hf⟩ := monoid_algebra.exists_left_inverse_of_injective
   p.subtype p.ker_subtype in
⟨f.ker, linear_map.is_compl_of_proj  linear_map.ext_iff.1 hf⟩
```

Proof: The idea behind Maschke's theorem is to find a homomorphism $\pi : V \to U$ such that

$$\forall u \in U \qquad \pi(u) = u$$

Imagine we had such a homomorphism π. Then, define W as

$$W = \ker \pi$$

We claim that $V = U \oplus W$. To see why, first we will show that $W \cap U = 0$. If $v \in U \cap W$, then $\pi(v) = 0$ because $v \in \ker \pi$, also $\pi(v) = v$ because $v \in U$, so $v = 0$. Now I will show that $v \in V$ is an arbitrary element of V, we can write v as a combination of U and W. It is

$$v = \pi(v) + (v - \pi(v))$$

We know that $\pi(v) \in V$, and we can calculate that

$$\pi(v - \pi(v)) = \pi(v) - \pi(\pi(v)) = \pi(v) - \pi(v) = 0 \implies v - \pi(v) \in W$$

This gives us that $V = U \oplus W$

Now the only step left is to find the homomorphism π. Let $\pi_0; V \to U$ be defined to be the projection of V onto U, i.e.

$$\pi_0(v) := u, \text{ where } v = w + u, v \in V, w \in W, u \in U$$

```
def conjugate (g : G) : W →ₗ[k] V :=
((group_smul.linear_map k V g⁻¹).comp π).comp
   (group_smul.linear_map k W g)
```

Now the idea is to define π as the "average" of π_0 across G, i.e.

$$\pi : V \to U$$

$$\pi(v) := \frac{1}{|G|} \sum_{g \in G} \rho(g)\pi_0\left(\rho(g^{-1})(v)\right)$$

```
def sum_of_conjugates : W →ₗ[k] V :=
Σ g : G, π.conjugate g
def equivariant_projection : W →ₗ[monoid_algebra k G] V :=
1/(fintype.card G : k) · (π.sum_of_conjugates_equivariant G)
```

Now since π is a scalar multiple of a sum of linear transformations, then π is a linear transformation.

For any element $u \in U$, π will just be the $\frac{1}{|G|}$ times $|G|$ sums of the identity, which shows that $\pi(u) = u$.

```
lemma equivariant_projection_condition (v : V) :
   (π.equivariant_projection G) (i v) = v :=
begin
  rw [equivariant_projection, smul_apply,
   sum_of_conjugates_equivariant,
    equivariant_of_linear_of_comm_apply, sum_of_conjugates],
  rw [linear_map.sum_apply],
  simp only [conjugate_i π i h],
  rw [finset.sum_const, finset.card_univ, nsmul_eq_smul_cast k,
    ←mul_smul, invertible.inv_of_mul_self, one_smul],
end
```

This means that π satisfies our requirements, so we are done with the proof. □

An interesting corollary of Maschke's theorem is that representations form a complemented lattice.

4. Gems

Up until now, we have seen the basics of Representation Theory, but now we will see a "gem" of Representation Theory to show why it is so interesting. Specifically, the character of a representation, which is a numerical invariant of a representation.

Definition 4. *The **character** of a representation is a function which associates an element of the group G with the the trace of the corresponding matrix.*

$$\chi_\rho : G \to F$$
$$\chi_\rho(g) = Tr(\rho(g))$$

```
def character (V : fdRep k G) (g : G) := linear_map.trace k V
    (V.ρ g)
```

The usefullness of the character comes from how it is a numerical invariant which encapsulates a representation. This is seen in the following theorem:

Theorem 5. *Two representations are equivalent if and only if they have the same character*

```
lemma char_iso  {V W : fdRep k G} (i : V ≅ W) : V.character =
    W.character :=
by { ext g, simp only [character, fdRep.iso.conj_ρ i], exact
    (trace_conj' (V.ρ g) _).symm }
```

Proof: The trace of a matrix is invariant under change-of-basis, which implies the desired result. □

Now the character has several interesting algebraic properties.

Theorem 6. *The character distributes over tensor product.*

$$\chi_{\rho\otimes\sigma} = \chi_\rho \cdot \chi_\sigma$$

```
@[simp] lemma char_tensor (V W : fdRep k G) : (V ⊗
    W).character = V.character * W.character :=
by { ext g, convert trace_tensor_product' (V.ρ g) (W.ρ g) }
```

Theorem 7. *The character distributes over dual.*

$$\chi_{\rho^*} = \overline{\chi_\rho}$$

```
@[simp] lemma char_dual (V : fdRep k G) (g : G) : (of (dual
    V.ρ)).character g = V.character g⁻¹ :=
  trace_transpose' (V.ρ g⁻¹)
```

Here is one more interesting property of character functions, which will be useful for proving their orthogonality.

Theorem 8. *A character is a* ***class function****, or in other words*

$$\forall g, x \in G \qquad \chi(gxg^{-1}) = \chi(x)$$

Proof:

```
/-- The character of a representation is constant on conjugacy
    classes. -/
@[simp] lemma char_conj (V : fdRep k G) (g : G) (h : G) :
  V.character (h * g * h⁻¹) = V.character g :=
by rw [char_mul_comm, inv_mul_cancel_left]
```

This theorem may be proved by applying properties of the trace, namely that $Tr(AB) = Tr(BA)$.

$$\begin{aligned} \phi(gxg^{-1}) &= \phi(g)\phi(x)\phi(g^{-1}) \\ &\implies \chi_\phi(gxg^{-1}) = Tr(\phi(g)\phi(x)\phi(g^{-1})) \\ &= Tr(\phi(x)\phi(g)\phi(g^{-1})) \\ &= \chi_\phi(x) \end{aligned}$$

□

Now where characters get interesting is with orthogonality. We can define an inner product on characters.

Definition 9. *Given two characters, χ_ρ and χ_σ, we can define their* ***inner product*** *to be*

$$\langle \chi_\rho, \chi_\sigma \rangle := \frac{1}{|G|} \sum_{g \in G} \chi_\rho(g) \overline{\chi_\sigma(g)}$$

Using this definition, we can prove (but I will not) that the characters form a basis of the class functions

Theorem 10. *The irreducible characters are a basis for the space of all complex class functions.*

From this, we get that all characters are orthogonal, or more precisely

Theorem 11. *All distinct characters are orthogonal.*

$$\langle \chi_i, \chi_j \rangle = \delta_{ij} = \begin{cases} 0 & \textit{if } i \neq j, \\ 1 & \textit{if } i = j. \end{cases}$$

(δ_{ij} denotes the Kroenecker delta function)

Proof: We have just established that the irreducible characters form an orthonormal basis for the space of class functions. If θ is any class function, write $\theta = \sum_{1=1}^{r} a_1 \chi_i$ for some $a_i \in \mathbb{C}$. It follows from linearity of the Hermitian product that $a_i = \langle \theta, \chi_i \rangle$. Specifcally, if we take $\theta = \chi_i$, then we get the desired result.

```
lemma char_orthonormal (V W : fdRep k G) [simple V] [simple W]
   :
  1/(fintype.card G : k) · Σ g : G, V.character g *
   W.character g⁻¹ =
  if nonempty (V ≅ W) then ↑1 else ↑0 :=
begin
  -- First, we can rewrite the summand ‘V.character g *
   W.character g⁻¹‘ as the character
  -- of the representation ‘V ⊗ W* ≅ Hom(W, V)‘ applied to
   ‘g‘.
  conv in (V.character _ * W.character _)
  { rw [mul_comm, ←char_dual, ←pi.mul_apply, ←char_tensor],
    rw [char_iso (fdRep.dual_tensor_iso_lin_hom W.ρ V)], } ,
  -- The average over the group of the character of a
   representation equals the dimension of the
  -- space of invariants.
  rw average_char_eq_finrank_invariants,
  rw [show (of (lin_hom W.ρ V.ρ)).ρ = lin_hom W.ρ V.ρ, from
   fdRep.of_ρ (lin_hom W.ρ V.ρ)],
  -- The space of invariants of ‘Hom(W, V)‘ is the subspace of
   ‘G‘-equivariant linear maps,
  -- ‘Hom_G(W, V)‘.
  rw (lin_hom.invariants_equiv_fdRep_hom W V).finrank_eq,
  -- By Schur’s Lemma, the dimension of ‘Hom_G(W, V)‘ is ‘1‘
   is ‘V ≅ W‘ and ‘0‘ otherwise.
  rw_mod_cast [finrank_hom_simple_simple W V,
   iso.nonempty_iso_symm],
end
```

□

4.1. **Character Tables.** Using the character, we can represent a group more compactly. One way to represent a group is with its multiplication table, but this grows quickly. For example, the monster group has order 10^{53}, but its character table is "only" a 194×194 array.

For this section, there will not be as much Lean code because it is harder to specify character tables in Lean.

Definition 12. *Given a group G, the* ***character table*** *of that group is a table of characters formatted as follows: list the representatives from each conjugacy class across the top and list the irreducible characters down the left side. The entry in row χ_i and column g_j is equal to $\chi_i(g_j)$.*

let's look at a simple example:

Example 13. *Let G be the group $G = (\{1, -1\}, \times, 1)$. Then, the character table of G is*

conjugacy classes	1	-1
sizes	1	1
χ_1	1	1
χ_2	1	-1

Perhaps one notices that the rows of the character table are orthogonal. This is related to the character orthogonality theorem from earlier. More precisely,

Theorem 14. *the rows of a character table are othogonal when multiplying by the size of the conjugacy classes.*

Let's look at another (slightly larger) example: the symmetric group S_3

Example 15. *Let $G = S_3$. The conjugacy classes (written in cycle notation) are $(), (12), (123)$. It's character table is*

conjugacy classes	$()$	(12)	(123)
sizes	1	3	2
χ_1	1	1	1
χ_2	1	-1	1
χ_3	2	0	1

One should notice that the first row of the character table will always be all 1 because ρ_1 is the trivial representation, and so $\chi_1 = 1$.

This example prompts another property of character tables: they are always square

Theorem 16. *All character tables are square, or put another way, there are as many irreducible characters as there are conjugacy classes.*

The way to see this is that because the characters are orthogonal, the matrix of the character table will be full-rank, so it can have at most n rows (if there are n columns/conjugacy classes).

Now we will give a more substantial example of a character table.

Example 17. *Let G be the group $G = D_8 = \{r^i s^j | i, j \in \mathbb{N}, r^4 = s^2 = 1\}$*

conjugacy classes	*1*	r^2	s	r	sr
sizes	1	1	*2*	*2*	*2*
χ_1	1	1	*1*	*1*	*1*
χ_2	1	1	*-1*	*1*	*-1*
χ_3	1	1	*1*	*-1*	*-1*
χ_4	1	1	*-1*	*-1*	*1*
χ_5	2	2	*0*	*0*	*0*

This example is more complicated, yet it still demonstrates the orthogonality relation on character tables.

The field of character theory is much deeper, but we do not have time to cover all of it.

5. Conclusion

I hope in this paper, you have gained an appreciation for Representation Theory, and how it can give a new perspective on problems in Group Theory by Viewing them through Linear Algebra. I also hope that the use of the Lean Theorem Prover has make the definitions and theorems more clear and apparant.