6.S966: Exam 1, Spring 2024

Do not tear exam booklet apart!

- This is a closed book exam. One page (8 1/2 in. by 11 in) of notes, front and back, are permitted. Calculators are not permitted.
- The total exam time is 1 hours and 20 minutes.
- The problems are not necessarily in any order of difficulty.
- Record all your answers in the places provided. If you run out of room for an answer, continue on a blank page and mark it clearly.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.
- If you absolutely *have* to ask a question, come to the front.
- **Write your name on every piece of paper.**

Name: ______________________ MIT Email: ______________________

Question	Points	Score
1	15	
2	28	
3	10	
4	27	
5	20	
Total:	100	

Name: ______________________________

Isomorphisms of Multiplication Tables

1. (15 points) Below are three group multiplication tables. Here, we are using various symbols instead of numbers, so it is clear in your answers, which table you are giving answers to.

	♋	♌	♊	♒
♋	♌	♋	♒	♊
♌	♋	♌	♊	♒
♊	♒	♊	♌	♋
♒	♊	♒	♋	♌

Table 1

	♣	♢	♠	♡
♣	♡	♠	♢	♣
♢	♠	♡	♣	♢
♠	♢	♣	♡	♠
♡	♣	♢	♠	♡

Table 2

	●	▲	■	★
●	★	■	●	▲
▲	■	★	▲	●
■	●	▲	■	★
★	▲	●	★	■

Table 3

(a) Give the symbol that corresponds to the identity element for each table

(b) For each symbol, give it's inverse.

♋$^{-1} \rightarrow$	♣$^{-1} \rightarrow$	●$^{-1} \rightarrow$
♌$^{-1} \rightarrow$	♢$^{-1} \rightarrow$	▲$^{-1} \rightarrow$
♊$^{-1} \rightarrow$	♠$^{-1} \rightarrow$	■$^{-1} \rightarrow$
♒$^{-1} \rightarrow$	♡$^{-1} \rightarrow$	★$^{-1} \rightarrow$

(c) Two of the three tables represent groups that are isomorphic. Which table represents a group that is not isomorphic to the other two? Explain your reasoning.

Name: ______________________

Parsing Proofs

2. (28 points) In this problem, we will present a single step of some of the proofs shown in class, exercises, or notes and ask what properties of matrices or groups or lemmas or theorem, allows us to take this step.

   (a) In our proof of Schur's Lemma Part 2, we are trying to prove the properties of matrix M in the equation

   $$MD^{(1)}(R) = D^{(2)}(R)M, \tag{1}$$

   for two unitary irreducible representations (irreps) of a group G, $D^{(1)}$ and $D^{(2)}$ and for all $R \in G$. We start by taking the conjugate transpose of this equations.

   $$[MD^{(1)}(R)]^\dagger = [D^{(2)}(R)M]^\dagger \tag{2}$$
   $$= [D^{(1)}(R)]^\dagger M^\dagger = M^\dagger [D^{(2)}(R)]^\dagger \tag{3}$$
   $$= D^{(1)}(R^{-1})M^\dagger = M^\dagger D^{(2)}(R^{-1}), \tag{4}$$

   i. What property of unitary matrices did we use between line (3) and (4)?

   ii. We multiply line (4) on the left by M

   $$MD^{(1)}(R^{-1})M^\dagger = MM^\dagger D^{(2)}(R^{-1}). \tag{5}$$

   Which line can we use above in lines (1 - 4) to arrive at

   $$D^{(2)}(R^{-1})MM^\dagger = MM^\dagger D^{(2)}(R^{-1})? \tag{6}$$

   Explain your reasoning.

iii. Using the lemmas and theorems that you've learned in this class what can you say about the matrix $MM^\dagger$ in line (6)? Explain your reasoning.

(b) In our proof of the Wonderful Orthogonality Theorem we started with the following

$$M = \sum_R D^{(\Gamma_{j'})}(R) X D^{(\Gamma_j)}(R^{-1}) \tag{7}$$

where $D^{(\Gamma_{j'})}$ and $D^{(\Gamma_j)}$ are representations of group G and X is an arbitrary matrix of size $l_{j'} \times l_j$, where $l_{j'}$ and l_j dimensions of the representations, respectively. We then multiplied by $D^{(\Gamma_{j'})}(S)$ on both sides

$$D^{(\Gamma_{j'})}(S)M = \sum_R D^{(\Gamma_{j'})}(S) D^{(\Gamma_{j'})}(R) X D^{(\Gamma_j)}(R^{-1}) \tag{8}$$

$$= \sum_R D^{(\Gamma_{j'})}(SR) X D^{(\Gamma_j)}(R^{-1}S^{-1}) D^{(\Gamma_j)}(S) \tag{9}$$

$$= M D^{(\Gamma_j)}(S) \tag{10}$$

i. What property of groups did we use to go from line (8) to line (9)? Explain your reasoning.

ii. What theorem did we use to go from line (9) to line (10)? Explain your reasoning.

(c) Recall that we defined group correlation as:

$$[f \star \psi](x,k) = \sum_{y \in \mathbb{Z}^2} \sum_{i \in G} f(y,i)\psi(k^{-1}(y-x), k^{-1}i) \tag{11}$$

We proved that this definition (11) satisfies equivalence as follows:

$$[L_g(f) \star \psi](x,k) = \sum_{y \in \mathbb{Z}^2} \sum_{i \in G} f(g^{-1}y, g^{-1}i)\psi(k^{-1}(y-x), k^{-1}i) \tag{12}$$

let $y' = g^{-1}y$ and $i' = g^{-1}i$

$$[L_g(f) \star \psi](x,k) = \sum_{y' \in \mathbb{Z}^2} \sum_{i' \in G} f(y', i')\psi(k^{-1}(gy' - x), k^{-1}gi') \tag{13}$$

$$= \sum_{y' \in \mathbb{Z}^2} \sum_{i' \in G} f(y', i')\psi((g^{-1}k)^{-1}(y' - g^{-1}x), (g^{-1}k)^{-1}i') = L_g([f \star \psi])(x,k) \tag{14}$$

i. What properties have we relied on when when replacing $\sum_i$ with $\sum_{i'}$ in going between (12) and (13) ?

ii. What properties / assumptions have we made when replacing $\sum_y$ with $\sum_{y'}$ in going between (12) and (13)? If there's any assumption made here, can you try to defend why it's a reasonable assumption?

Name: ______________________________

Sudoku for Group Theorists

3. (10 points) Complete the character tables below using First and Second Wonderful Orthogonal Theorems for Character given below, respectively.

$$\sum_R \chi^{(\Gamma_j)}(R)\chi^{(\Gamma_{j'})}(R^{-1}) = \sum_k N_k \chi^{(\Gamma_j)}(C_k)[\chi^{(\Gamma_{j'})}(C_k)]^* = h\delta_{\Gamma_j \Gamma_{j'}} \tag{15}$$

$$\sum_{\Gamma_i} N_k \chi^{(\Gamma_i)}(C_k)[\chi^{(\Gamma_i)}(C_{k'})]^* = h\delta_{kk'}. \tag{16}$$

Γ are representations of the group, R is an element of the group, h is the size (order) of the group, C_k is a conjugacy class, N_k is the size of a conjugacy class.

(a) Complete character table below where A_1 represents the trivial irrep, and A_2, B_1, and B_2 are 1D irreps.

	e	C_2 (z)	$\sigma_v(xz)$	$\sigma_v(yz)$
A_1	____	____	____	____
A_2	____	____	-1	-1
B_1	____	____	1	-1
B_2	____	-1	-1	1

(b) Cyclic groups are only irreducible over complex numbers (rather than real numbers), but have all 1D irreps. Complete the character table below for the cyclic group C_5 where A_1 is the trivial irrep and A_2 and A_3 are (complex) 1D representations. $\omega = e^{i2\pi/5}$, $\omega^2 = e^{i4\pi/5}$, etc. Hint: $1 + \omega + \omega^2 + \omega^3 + \omega^4 = 0$

	E	C_5	$(C_5)^2$	$(C_5)^3$	$(C_5)^4$
A_1	____	____	____	____	____
A_2	____	ω	ω^2	ω^3	ω^4
A_3	____	ω^4	____	____	____
A_4	____	ω^2	____	ω	ω^3
A_5	____	ω^3	____	ω^4	ω^2

Interpreting Outputs

4. (27 points) In the following questions, we will present you code snippets using the functions you have coded in the exercises and ask you to interpret what the outputs means. You may assume the following has been imported. The Docstrings for functions used in this problem are available at near the end of your exam booklet, before the Work space pages.

```
import numpy as np
from symm4ml import groups, linalg, rep, vis
import torch
```

(a) In this problem, we will chain together several code snippets to determine properties of two representations of the group D_4, the rotation and mirror symmetries of a square.

i. Consider the following code snippet.

```
1  subset_D4 = np.array([
2      # mirror across y axis
3      [[-1., 0.],
4       [0., 1]],
5      # Four-fold (90 degree) rotation counterclockwise
6      [[np.cos(np.pi / 2), -np.sin(np.pi / 2)],
7       [np.sin(np.pi / 2), np.cos(np.pi / 2)]],
8  ])
9  D4 = groups.generate_group(subset_D4)
10 D4_table = groups.make_multiplication_table(D4)
11 D4_reg_rep = rep.regular_representation(D4_table)
```

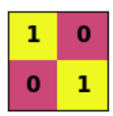
Describe what is happening in lines 1-9.

ii. Describe how the (left) regular representation `D4_reg_rep` is constructed from `D4_table` in line (10-11).

iii. What are the dimensions (shape) of the representation of `D4` and `D4_reg_reg`? Explain your reasoning.

iv. Now, we continue with the following code snippet.

```
12 M = linalg.infer_change_of_basis(D4, D4)
13 norm_M = M / np.max(np.abs(M))
14 norm_M[np.isclose(norm_M, -0.0)] = +0.0
15
16 vis.plot_image_values(
17     norm_M, norm_M, vmax=1, vmin=-1, colormap='plasma',
18     decimal_places=0, figsize=(1, 1), size=12,
19     fontcolor=['white', 'black'], fontcolor_cutoff=-0.1
20 );
21 >>
```

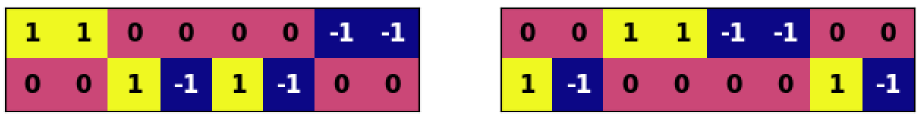


In line 13-14, we are simply normalizing the matrix by its absolute maximum value and replacing `-0`s with `+0`s, so it's nicer to visualize using the code in lines 16-20. You can assume there are no rounding errors.

Describe what is happening in line 12 and explain the significance of `linalg.infer_change_of_basis` returning a single matrix equal to a constant matrix (identity multiplied by a constant). What does this tell us about the representation of D_4 as 2D rotations and mirrors? Explain your reasoning.

v. Finally, we finish with the following code snippet.

```
22 cob = linalg.infer_change_of_basis(D4, D4_reg_rep)
23 norm_cob = cob/np.max(np.abs(cob))
24 norm_cob[np.isclose(norm_cob, -0.0)] = +0.0
25
26 vis.plot_image_values(
27     norm_cob, norm_cob, vmax=1, vmin=-1, colormap='plasma',
28     decimal_places=0, figsize=(8, 2), size=12,
29     fontcolor=['white', 'black'], fontcolor_cutoff=-0.1
30 );
31 >>
```



Again line 23-24, we are simply normalizing the matrix by its absolute maximum value and replacing `-0`s with `+0`s, so it's easier to visualize using the code in lines 26-30. You can assume there are no rounding errors.

Describe what is happening in line 22. Why does `linalg.infer_change_of_basis(D4, D4_reg_rep)` return two matrices? What does this tell us about `D4_reg_rep` given what we know about `D4` from part (iv)? How does this align with what we know about the properties of the regular representation from lecture?

(b) Consider the following code snippet and output.

```
linalg.nullspace(np.array([[1, 0, 0], [0, 0, 0], [0, 0, 0]]))
>> (array([[0., 1., 0.],
        [0., 0., 1.]]),
 array([[0., 0., 0.],
        [0., 1., 0.],
        [0., 0., 1.]]))
```

i. What does `linalg.nullspace` do and how are the first output `array([[0., 1., 0.], [0., 0., 1.]])` and the second output `array([[0., 0., 0.], [0., 1., 0.], [0., 0., 1.]])` related?

ii. Is the first output unique (e.g. are there any other possible outputs that would also be correct)? Explain your reasoning.

iii. Is the second output unique? Explain your reasoning.

Name: ____________________

This Side Up

5. (20 points) After graduation, you join a startup building next generation photo digitizers. Your first task is to implement a feature that automatically rotates square photos so that they are in the right direction regardless of how users input the photo. You wonder if this task is well-suited for group convolution.

   (a) Assume the scanner scans a single side of square photos. What group should you use to build your Group Convolutional Neural Network? You can use its shorthand name or describe what elements are in the group. Feel free to draw a diagram. Explain your reasoning.

   (b) Recall that group convolution was defined to be equivariant under under the action $L_g(f) = f \circ g^{-1}$, i.e., $L_g(f) \star \psi = L_g(f \star \psi)$. If f was a function over $\mathbb{Z}^2$ such that $f(x)$ is the value of the pixel at x, then $[L_g(f)](x) = f(D_{2d}(g^{-1})x)$, where D_{2d} is a 2-dimensional representation of group $\mathcal{G}$. If $D_{2d}(g)$ was a matrix that rotated by 90° clockwise, how does the image of the function represented by $L_g(f)$ relate to that represented by f?.

   Hint: If you do an example, define f and apply $l_g(f)$, before drawing images.

   ◯ 90° rotation counter-clockwise

   ◯ 90° rotation clockwise

   Explain your answer with a simple example (like a single dot on a corner) or a proof:

   Relating this to lecture: We defined L_g when f has an additional regular representation dimension. That is f has arguments $(x, i) \in \mathbb{Z} \times \mathcal{G}$. $L_g(f) = f(D_{2d}(g^{-1})x, g^{-1}i)$. This has the effect of permuting the elements in the group dimension on top of the effect of the action on the spacial dimension (this question is only asking you to identify the effect on the spacial dimension), since we repeat the given image across the regular representation dimension in the input before passing it to the first layer. The action you identified is the way to act on the input of the network (as permuting a repeated value has no effect).

(c) You build a model by purely stacking group convolutions, point-wise non-linearities and pooling operations, as discussed in class. The second convolution layer has 4 input channels and 4 output channels with kernel height 3 and kernel width 3.

How many parameters does the second layer have?

Hint: You might want to look at the doc string for the following, as it takes the filter parameters as input: `group_conv.image2D_group_convolution_filter_bank`.

(d) In your model from part (c), the last layer outputs a single channel with a regular representation dimension equal to the size of the group you chose in Part (a), and spatial dimensions $H = W = 1$, i.e. the output has shape `[1, |G|, 1, 1]`. Finally, we apply a soft-max function ($\frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$) to this output to convert the output values to probabilities over the group elements (the vectorspace of the regular representation).

In training, you use cross entropy loss between the output of the model, and a one hot encoding of a group element (assume the content of the images is not symmetric for this task to be meaningful). For a properly oriented images you use a one hot encoding of the identity e as a label.

What element should you use as a label for an image that resulted from applying L_g to a properly oriented image, where both images are defined to be functions of pixel coordinates? Look at part (b) to remember what this means.

Which of the following should you use a one hot encoding of in the loss, and why?

- ○ g
- ○ g^{-1}
- ○ Either works.
- ○ Something else.

Explain your answer:

Hint: If we predict either, we can recover the transformation we need to apply to the image. So the questions is concerning what can group convolution learn as a target, not what group element directly represents the transformation we want to apply to our image.

(e) You are sharing your finding with your team, and of them says that you should augment the data by rotating the images and updating the labels (the one hot encodings). Do you think that would help improve performance? Why?

(f) What do you think the model would predict when given an blank photo?

Name: ____________________

symm4ml Docstring listing

Modules listed in order: `groups`, `group_conv`, `linalg`, `rep`

```
groups
----------------------
groups.generate_group:
Generate new group elements from matrices (group representations)
    Input:
        matrices: np.array of shape [n, d, d] of known elements
        decimals: int number of decimals to round to when comparing matrices
    Output:
        group: np.array of shape [m, d, d], where m is the size of the resultant
            group


groups.make_multiplication_table:
Makes multiplication table for group.
    Input:
        matrices: np.array of shape [n, d, d], n matrices of dimension d that form
             a group under matrix multiplication.
        tol: float numberical tolerance
    Output:
        Group multiplication table.
        np.array of shape [n, n] where entries correspond to indices of first dim
            of matrices.



group_conv
----------------------
group_conv.image2D_group_convolution:
Performs group convolution of inputs and filters over the regular representation
    Input:
        rep_2D: torch.Tensor of shape [|G|, 2, 2] of the group representation as 2
            D rotations and mirrors
        rep_reg: torch.Tensor of shape [|G|, |G|, |G|] of the left regular
            representation
        inverse: torch.LongTensor of shape [|G|] with the indices of the inverses
            of the group elements
        input: torch.Tensor of shape [batch, channel_in, rep_reg_in, height, width
            ]
        filter: torch.Tensor of shape [channel_out, channel_in, rep_reg_filter,
            kernel_height, kernel_width]
    Output:
        output: torch.Tensor of shape [batch, channel_out, rep_reg_out]
```

```
group_conv.image2D_group_convolution_filter_bank:
Creates rotated filter bank for 2D image convolution
    Input:
        rep_2D: torch.Tensor of shape [|G|, 2, 2] of the group representation as 2
            D rotations and mirrors
        filter: torch.Tensor of shape [channel_out, channel_in, rep_reg_filter,
            kernel_height, kernel_width]
    Output:
        filter_bank: torch.Tensor of shape [rep_reg_out, channel_out, channel_in,
            rep_reg_filter, kernel_height, kernel_width]


linalg
----------------------
linalg.infer_change_of_basis:
Compute the change of basis matrix from X1 to X2.
    tip: Use the function nullspace
    Input:
        X1: an (n, d1, d1) array of n (d1, d1) matrices
        X2: an (n, d2, d2) array of n (d2, d2) matrices
    Output:
        Sols: An (m, d1, d2) array of m solutions.
        Each solution is a (d1, d2) matrix that satisfies X1 @ S = S @ X2.

linalg.nullspace:
Return the nullspace of the matrix.
    Input:
        matrix: an (n, d) matrix of n complex vectors of dimension d
        tol: a tolerance for the zero eigenvalue
    Output:
        Q: an (m, d) matrix containing orthogonal vectors spanning the nullspace (
            obtained by Gram-Schmidt)
        P: a (d, d) projector onto the span of the nullspace


rep
----------------------
rep.regular_representation:
Returns regular representation for group represented by a multiplication table.
    Input:
        table: np.array [n, n] where table[i, j] = k means i * j = k.
    Output:
        Regular representation. array [n, n, n] where reg_rep[i, :, :] = D(i) and
            D(i)e_j = e_{ij}.
                             Equivalently, D(g) |h> = |gh>
```

Work space

Work space