

# SL2Reps

**Constructs representations of  $SL_2(\mathbb{Z})$ .**

0.1

24 September 2021

**Siu-Hung Ng**

**Yilong Wang**

**Samuel Wilson**

**Siu-Hung Ng**

Email: [rng@math.lsu.edu](mailto:rng@math.lsu.edu)

**Yilong Wang**

Email: [wyl@bimsa.cn](mailto:wyl@bimsa.cn)

**Samuel Wilson**

Email: [swil311@lsu.edu](mailto:swil311@lsu.edu)

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Usage . . . . .	3
<b>2</b>	<b>Description</b>	<b>4</b>
2.1	Construction . . . . .	4
2.2	Weil representation types . . . . .	5
<b>3</b>	<b>Lists of representations</b>	<b>7</b>
3.1	Lists by degree . . . . .	7
3.2	Lists by level . . . . .	7
3.3	Lists of exceptional representations . . . . .	8
<b>4</b>	<b>Methods for testing</b>	<b>9</b>
4.1	Testing . . . . .	9
<b>5</b>	<b>Irreducible representations of prime-power level</b>	<b>10</b>
5.1	Representations of type D . . . . .	10
5.2	Representations of type N . . . . .	11
5.3	Representations of type R . . . . .	11
	<b>References</b>	<b>14</b>
	<b>Index</b>	<b>15</b>

# Chapter 1

## Introduction

This package, `SL2Reps`, provides methods for constructing and testing matrix presentations of the representations of  $SL_2(\mathbb{Z})$  whose kernels are congruence subgroups of  $SL_2(\mathbb{Z})$ .

Irreducible representations of prime-power level are constructed individually by using the Weil representations of quadratic modules, and from these a list of all representations of a given degree or level can be produced. The format is designed for the study of modular tensor categories in particular.

### 1.1 Installation

To install `SL2Reps`, first download it from `TODD`, then place it in the `pkg` subdirectory of your GAP installation (or in the `pkg` subdirectory of any other GAP root directory, for example one added with the `-l` argument).

`SL2Reps` is then loaded with the GAP command

```
gap> LoadPackage( "SL2Reps" );
```

### 1.2 Usage

Specific irreducible representations may be constructed via the methods in Chapter 5, while lists of irreducible representations with a given degree or level may be constructed with those in Chapter 3.

This package uses an `InfoClass`, `InfoSL2Reps`. It may be set to 0 (silent), 1 (info), or 2 (verbose). To change it, use

```
gap> SetInfoLevel(InfoSL2Reps, k);
```

## Chapter 2

# Description

The group  $\mathrm{SL}_2(\mathbb{Z})$  is generated by  $\mathfrak{s} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$  and  $\mathfrak{t} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$  (which satisfy the relations  $\mathfrak{s}^4 = (\mathfrak{st})^3 = \mathrm{id}$ ). Thus, any complex representation  $\rho$  of  $\mathrm{SL}_2(\mathbb{Z})$  on  $\mathbb{C}^n$  (where  $n \in \mathbb{Z}^+$  is called the *degree* of  $\rho$ ) is defined by the  $n \times n$  matrices  $S = \rho(\mathfrak{s})$  and  $T = \rho(\mathfrak{t})$ .

This package constructs representations which factor through  $\mathrm{SL}_2(\mathbb{Z}/\ell\mathbb{Z})$  for some  $\ell \in \mathbb{Z}^+$ ; the smallest such  $\ell$  is called the *level* of the representation. It may be shown that any representation arising from a modular tensor category has this property [DLN15]. We therefore present representations in the form of a record `rec(S, T, degree, level, name)` where the name follows the conventions of [NW76].

Note that our definition of  $\mathfrak{s}$  follows that of [Nob76]; other authors prefer the inverse, i.e.  $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ . (which provides a different presentation of  $\mathrm{SL}(2, \mathbb{Z})$ .) When working with that convention, one must invert the  $S$  matrices output by this package.

Throughout, we denote by  $\mathbf{e}$  the map  $k \mapsto e^{2\pi i k}$  (an isomorphism from  $\mathbb{Q}/\mathbb{Z}$  to the group of finite roots of unity in  $\mathbb{C}$ ).

### 2.1 Construction

Any representation  $\rho$  of  $\mathrm{SL}_2(\mathbb{Z})$  can be decomposed into a direct sum of irreducible representations (irreps). Further, each irrep can be factorized into a tensor product of irreps whose levels are powers of distinct primes (using the Chinese remainder theorem). Therefore, to characterize all representations of  $\mathrm{SL}_2(\mathbb{Z})$ , it suffices to consider irreps of  $\mathrm{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$  for primes  $p$  and positive integers  $\lambda$ .

Such representations may be constructed using Weil representations as described in [Nob76, Section 1]. We give a brief summary of the process here. First, if  $M$  is any additive abelian group, a *quadratic form* on  $M$  is a map  $Q : M \rightarrow \mathbb{Q}/\mathbb{Z}$  such that  $Q(-x) = Q(x)$  for all  $x \in M$  and  $B(x, y) = Q(x + y) - Q(x) - Q(y)$  defines a  $\mathbb{Z}$ -bilinear map  $M \times M \rightarrow \mathbb{Q}/\mathbb{Z}$ .

Now let  $p$  be a prime number and  $\lambda \in \mathbb{Z}^+$ . Choose a  $\mathbb{Z}/p^\lambda\mathbb{Z}$ -module  $M$  of rank 1 or 2 and a quadratic form  $Q$  on  $M$  such that the pair  $(M, Q)$  is of one of the three types described in Section 2.2. Then the *quadratic module*  $(M, Q)$  gives rise to a representation of  $\mathrm{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$  on the vector space  $V = \mathbb{C}^M$  of complex-valued functions on  $M$ . This representation is denoted  $W(M, Q)$ .

With a finite number of exceptions, every representation of  $\mathrm{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$  may be found as a subrepresentation of  $W(M, Q)$  for an appropriate choice of  $(M, Q)$  [NW76, Hauptsatz 2] (the 18 exceptions can be constructed as the tensor product of two such subrepresentations; these can be generated with `SL2ExceptionalIrreps` (3.3.1)).

The subrepresentations in question are often of the form  $W(M, Q, \chi)$ , defined as follows. Let  $\mathfrak{A}$  be an abelian subgroup of

$$\text{Aut}(M, Q) = \{\varepsilon \in \text{Aut}(M) \mid Q(\varepsilon x) = Q(x) \text{ for all } x \in M\}$$

and let  $\chi \in \widehat{\mathfrak{A}}$  be a 1-dimensional character of  $\mathfrak{A}$ . Define

$$V_\chi = \{f \in V \mid f(\varepsilon x) = \chi(\varepsilon)f(x) \text{ for all } x \in M \text{ and } \varepsilon \in \mathfrak{A}\},$$

which is a  $SL_2(\mathbb{Z}/p^\lambda\mathbb{Z})$ -invariant subspace of  $V$ . We then denote by  $W(M, Q, \chi)$  the subrepresentation of  $W(M, Q)$  on  $V_\chi$ .

In this context, we will frequently refer to a character  $\chi$  as being *primitive*, which means that there exists some  $\varepsilon \in \mathfrak{A}$  such that  $\chi(\varepsilon) \neq 1$  and  $\varepsilon$  fixes the submodule  $pM$  pointwise (except for one special case of type  $R$ ; for this, see Section 2.2.3). This amounts to saying that  $\chi$  does not factor through a quadratic module of lower level.

All the irreducible representations of  $SL_2(\mathbb{Z})$  of finite level can now be constructed by taking tensor products of these prime-power irreps. Note that, if two representations are defined by pairs  $[S1, T1]$  and  $[S2, T2]$ , then their tensor product may be calculated using the GAP command `KroneckerProduct`, namely as `[KroneckerProduct(S1,S2), KroneckerProduct(T1,T2)]`.

## 2.2 Weil representation types

### 2.2.1 Type D

Let  $p$  be prime and  $\lambda \geq 1$ . Then the Weil representation arising from the quadratic module with  $M = \mathbb{Z}/p^\lambda\mathbb{Z} \oplus \mathbb{Z}/p^\lambda\mathbb{Z}$  and  $Q(x, y) = \frac{xy}{p^\lambda}$  is said to be of type  $D$  and denoted  $D(p, \lambda)$ . Information on  $(M, Q)$  may be obtained via `SL2ModuleD` (5.1.1), and subrepresentations of  $D(p, \lambda)$  with level  $p^\lambda$  may be constructed via `SL2IrrepD` (5.1.2).

Here we define

$$\mathfrak{A} = (\mathbb{Z}/p^\lambda\mathbb{Z})^\times$$

acting on  $M$  by multiplication; see [NW76, Section 2.1]. This group has the following structure. When  $p = 2$  and  $\lambda \geq 3$ , it is generated by  $\alpha = -1$  and  $\zeta = 5$ . Otherwise, it is cyclic; in this case, we choose a generator  $\alpha$  and (for simplicity) say  $\zeta = 1$ .

A character of  $\mathfrak{A}$  is primitive if and only if it is injective on the subgroup  $\langle o \rangle \leq \mathfrak{A}$ , where  $o$  is chosen as follows. If  $p = 2$  and  $\lambda \geq 3$ ,  $o = 5$ . If  $\lambda = 1$ ,  $o = \alpha$ . In all other cases,  $o = 1 + p$ .

### 2.2.2 Type N

Let  $p$  be prime and  $\lambda \geq 1$ . Then the Weil representation arising from the quadratic module with  $M = \mathbb{Z}/p^\lambda\mathbb{Z} \oplus \mathbb{Z}/p^\lambda\mathbb{Z}$  and  $Q(x, y) = \frac{x^2 + xy + \frac{1+u}{4}y^2}{p^\lambda}$  (where, for  $p \neq 2$ ,  $u$  is chosen so that  $u \equiv 3 \pmod{4}$  with  $\left(\frac{-u}{p}\right) = -1$ , and for  $p = 2$ ,  $u = 3$ ) is said to be of type  $N$  and denoted  $N(p, \lambda)$ . Information on  $(M, Q)$  may be obtained via `SL2ModuleN` (5.2.1), and subrepresentations of  $D(p, \lambda)$  with level  $p^\lambda$  may be constructed via `SL2IrrepN` (5.2.2).

Here we define

$$\mathfrak{A} = \{\varepsilon \in M^\times \mid \text{Nm}(\varepsilon) = 1\}$$

acting on  $M$  by multiplication; see [NW76, Section 2.2]. This group has the following structure. When  $\lambda \geq 2$ , it is generated by  $\alpha$  and  $\zeta$ : for  $p = 2$ ,  $|\alpha| = 2^{\lambda-2}$  and  $|\zeta| = 6$ , while for  $p \neq 2$ ,  $|\alpha| = p^{\lambda-1}$  and  $|\zeta| = p + 1$ . On the other hand, when  $\lambda = 1$ ,  $\mathfrak{A}$  is cyclic; in this case, we choose a generator  $\zeta$  with  $|\zeta| = p + 1$  and (for simplicity) say  $\alpha = 1$ .

As a special case, for  $p = 2$ ,  $\lambda = 2$ , a character of  $\mathfrak{A}$  is primitive if and only if  $\chi(-1) = -1$ . For all other cases, a character of  $\mathfrak{A}$  is primitive if and only if it is injective on the subgroup  $\langle o \rangle \leq \mathfrak{A}$ , where  $o$  is chosen as follows. If  $\lambda = 1$ ,  $o = \zeta$ ; otherwise  $o = \alpha$ .

### 2.2.3 Type R

The structure of  $(M, Q)$  of type  $R$  depends upon three additional parameters:  $\sigma$ ,  $r$ , and  $t$ . The relevant values thereof depend on whether  $p = 2$ , as follows.

First, if  $p$  is an odd prime, let  $\lambda \geq 2$ ,  $\sigma \in \{1, \dots, \lambda\}$ , and  $r, t \in \{1, u\}$  with  $u$  a quadratic non-residue mod  $p$ . Then define  $M = \mathbb{Z}/p^\lambda \mathbb{Z} \oplus \mathbb{Z}/p^{\lambda-\sigma} \mathbb{Z}$  and  $Q(x, y) = \frac{r(x^2 + p^\sigma t y^2)}{p^\lambda}$ .

On the other hand, if  $p = 2$ , let  $\lambda \geq 2$ ,  $\sigma \in \{0, \dots, \lambda - 2\}$  and  $r, t \in \{1, 3, 5, 7\}$ . Then define  $M = \mathbb{Z}/2^{\lambda-1} \mathbb{Z} \oplus \mathbb{Z}/2^{\lambda-\sigma-1} \mathbb{Z}$  and  $Q(x, y) = \frac{r(x^2 + 2^\sigma t y^2)}{2^\lambda}$ .

In either case, the resulting representation is said to be of type  $R$  and denoted  $R(p, \lambda, \sigma, r, t)$ . Information on  $(M, Q)$  may be obtained via `SL2ModuleR` (5.3.1), and subrepresentations of  $R(p, \lambda, \sigma, r, t)$  with level  $p^\lambda$  may be constructed via `SL2IrrepR` (5.3.2). Note that if  $\sigma = \lambda$  for  $p \neq 2$ , then the second factor of  $M$  is trivial (and hence  $t$  is irrelevant); this special case is handled by `SL2IrrepRUnary` (5.3.3).

## Chapter 3

# Lists of representations

### 3.1 Lists by degree

#### 3.1.1 SL2PrimePowerIrrepsOfDegree

- ▷ `SL2PrimePowerIrrepsOfDegree(degree)` (function)  
**Returns:** a list of records of the form `rec(S, T, degree, level, name)`  
Constructs a list of all irreps of  $SL_2(\mathbb{Z})$  that are exactly the given degree and have prime power level.

#### 3.1.2 SL2PrimePowerIrrepsOfDegreeAtMost

- ▷ `SL2PrimePowerIrrepsOfDegreeAtMost(max_degree)` (function)  
**Returns:** a list of records of the form `rec(S, T, degree, level, name)`  
Constructs a list of all irreps of  $SL_2(\mathbb{Z})$  that are at most the given degree and have prime power level.

#### 3.1.3 SL2IrrepsOfDegree

- ▷ `SL2IrrepsOfDegree(degree)` (function)  
**Returns:** a list of records of the form `rec(S, T, degree, level, name)`  
Constructs a list of all irreps of  $SL_2(\mathbb{Z})$  that are exactly the given degree.

#### 3.1.4 SL2IrrepsOfDegreeAtMost

- ▷ `SL2IrrepsOfDegreeAtMost(degree)` (function)  
**Returns:** a list of records of the form `rec(S, T, degree, level, name)`  
Constructs a list of all irreps of  $SL_2(\mathbb{Z})$  that are at most the given degree.

### 3.2 Lists by level

#### 3.2.1 SL2PrimePowerIrrepsOfLevel

- ▷ `SL2PrimePowerIrrepsOfLevel(p, lambda)` (function)  
**Returns:** a list of records of the form `rec(S, T, degree, level, name)`  
Constructs a list of all irreps of  $SL_2(\mathbb{Z})$  with level exactly  $p^\lambda$ .

### 3.3 Lists of exceptional representations

#### 3.3.1 SL2ExceptionalIrreps

▷ `SL2ExceptionalIrreps(arg)`

(function)

**Returns:** a list of records of the form `rec(S, T, degree, level, name)`

Constructs a list of the 18 exceptional irreps of  $\mathrm{SL}_2(\mathbb{Z})$ .



## Chapter 4

# Methods for testing

### 4.1 Testing

#### 4.1.1 SL2WithConjClasses

▷ `SL2WithConjClasses(p, ld)` (function)

**Returns:** the group  $\mathrm{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$  with conjugacy classes set to the format we use.

#### 4.1.2 SL2ChiST

▷ `SL2ChiST(S, T, p, ld)` (function)

**Returns:** a list representing a character of  $\mathrm{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$

Converts the modular data  $(S, T)$ , which must have level dividing  $p^\lambda$ , into a character of  $\mathrm{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$ , presented in a form matching the conjugacy classes used in `SL2WithConjClasses`.

#### 4.1.3 SL2IrrepPositionTest

▷ `SL2IrrepPositionTest(p, lambda)` (function)

**Returns:** a boolean

Constructs and tests all irreps of level dividing  $p^\lambda$  by checking their positions in  $\mathrm{Irr}(G)$ .

## Chapter 5

# Irreducible representations of prime-power level

Methods for generating individual irreducible representations of  $SL_2(\mathbb{Z}/p^\lambda\mathbb{Z})$  for a given level  $p^\lambda$ .

In each case (except the unary type  $R$ , for which see `SL2IrrepRUnary` (5.3.3)), the underlying module  $M$  is of rank 2, so its elements have the form  $(x, y)$  and are thus represented by lists  $[x, y]$ .

### 5.1 Representations of type D

See Section 2.2.1.

#### 5.1.1 SL2ModuleD

▷ `SL2ModuleD(p, ld)` (function)

**Returns:** a record `rec(Agrp, Bp, Char, IsPrim)` describing  $(M, Q)$

Constructs information about the underlying quadratic module  $(M, Q)$  of type  $D$ , for  $p$  a prime and  $\lambda \geq 1$ .

`Agrp` is a list describing the elements of  $\mathfrak{A}$ . Each element  $a \in \mathfrak{A}$  is represented in `Agrp` by a list `[v, a, a_inv]`, where `v` is a list defined by  $a = \alpha^{v[1]} \zeta^{v[2]}$ . Note that  $\zeta$  is trivial, and hence `v[2]` is irrelevant, when  $\mathfrak{A}$  is cyclic.

`Bp` is a list of representatives for the  $\mathfrak{A}$ -orbits on  $M^\times$ , which correspond to a basis for the  $SL_2(\mathbb{Z}/p^\lambda\mathbb{Z})$ -invariant subspace associated to any primitive character  $\chi \in \hat{\mathfrak{A}}$  with  $\chi^2 \neq 1$ . For other characters, we must use different bases which are particular to each case.

`Char(i, j)` converts two integers  $i, j$  to a function representing a character of  $\mathfrak{A}$ . Each character in  $\hat{\mathfrak{A}}$  is of the form  $\chi_{i,j}$ , given by  $\chi_{i,j}(\alpha^v \zeta^w) \mapsto \mathbf{e}^{\left(\frac{vi}{\alpha} - \frac{wj}{\zeta}\right)}$ . Note that  $j$  is irrelevant when  $\mathfrak{A}$  is cyclic.

`IsPrim(chi)` tests whether the output of `Char(i, j)` represents a primitive character.

#### 5.1.2 SL2IrrepD

▷ `SL2IrrepD(p, ld, chi_index)` (function)

**Returns:** a list of lists of the form  $[S, T]$

Constructs the modular data for the irreducible representation(s) of type  $D$  with level  $p^\lambda$ , for  $p$  a prime and  $\lambda \geq 1$ , corresponding to the character  $\chi$  indexed by `chi_index = [i,j]` (see the discussion of `Char(i,j)` in `SL2ModuleD` (5.1.1)).

Depending on the parameters,  $W(M, Q)$  will contain either 1 or 2 such irreps.

## 5.2 Representations of type N

See Section 2.2.2.

### 5.2.1 SL2ModuleN

▷ `SL2ModuleN(p, ld)` (function)

**Returns:** a record `rec(Agrp, Bp, Char, IsPrim, Nm, Prod)` describing  $(M, Q)$

Constructs information about the underlying quadratic module  $(M, Q)$  of type  $N$ , for  $p$  a prime and  $\lambda \geq 1$ .

`Agrp` is a list describing the elements of  $\mathfrak{A}$ . Each element  $a \in \mathfrak{A}$  is represented in `Agrp` by a list `[v, a]`, where  $v$  is a list defined by  $a = \alpha^{v[1]} \zeta^{v[2]}$ . Note that  $\alpha$  is trivial, and hence  $v[1]$  is irrelevant, when  $\mathfrak{A}$  is cyclic.

`Bp` is a list of representatives for the  $\mathfrak{A}$ -orbits on  $M^\times$ , which correspond to a basis for the  $SL_2(\mathbb{Z}/p^\lambda\mathbb{Z})$ -invariant subspace associated to any primitive character  $\chi \in \hat{\mathfrak{A}}$  with  $\chi^2 \neq 1$ . For other characters, we must use different bases which are particular to each case.

`Char(i,j)` converts two integers  $i, j$  to a function representing a character of  $\mathfrak{A}$ . Each character in  $\hat{\mathfrak{A}}$  is of the form  $\chi_{i,j}$ , given by  $\chi_{i,j}(\alpha^v \zeta^w) = \mathbf{e}\left(\frac{vi}{\alpha} - \frac{wj}{\zeta}\right)$ . Note that  $i$  is irrelevant in the cases where  $\mathfrak{A}$  is cyclic.

`IsPrim(chi)` tests whether the output of `Char(i,j)` represents a primitive character.

`Nm(a)` and `Prod(a,b)` are the norm and product functions on  $M$ , respectively.

### 5.2.2 SL2IrrepN

▷ `SL2IrrepN(p, ld, chi_index)` (function)

**Returns:** a list of lists of the form `[S, T]`

Constructs the modular data for the irreducible representation(s) of type  $N$  with level  $p^\lambda$ , for  $p$  a prime and  $\lambda \geq 1$ , corresponding to the character  $\chi$  indexed by `chi_index = [i,j]` (see the discussion of `Char(i,j)` in `SL2ModuleN` (5.2.1)).

Depending on the parameters,  $W(M, Q)$  will contain either 1 or 2 such irreps.

## 5.3 Representations of type R

See Section 2.2.3.

### 5.3.1 SL2ModuleR

▷ `SL2ModuleR(p, ld, sigma, r, t)` (function)

**Returns:** a record `rec(Agrp, Bp, Char, IsPrim, Nm, Ord, Prod, c, tM)` describing  $(M, Q)$  a record `rec(Agrp, Char, IsPrim, Nm, Ord, Prod, c, tM)` describing  $(M, Q)$

Constructs information about the underlying quadratic module  $(M, Q)$  of type  $R$ , for  $p$  a prime. The additional parameters  $\lambda$ ,  $\sigma$ ,  $r$ , and  $t$  should be integers chosen as follows.

If  $p$  is an odd prime, let  $\lambda \geq 2$ ,  $\sigma \in \{1, \dots, \lambda - 1\}$ , and  $r, t \in \{1, u\}$  with  $u$  a quadratic non-residue mod  $p$ . Note that  $\sigma = \lambda$  is a valid choice for type  $R$ , however, this gives the unary case, and so is not handled by this function, as it is decomposed in a different way; for this case, use `SL2IrrepRUnary` (5.3.3) instead.

If  $p = 2$ , let  $\lambda \geq 2$ ,  $\sigma \in \{0, \dots, \lambda - 2\}$  and  $r, t \in \{1, 3, 5, 7\}$ .

`Agrp` is a list describing the elements of  $\mathfrak{A} = \{\varepsilon \in M^\times \mid \text{Nm}(\varepsilon) = 1\}$  (see [NW76, Section 2.3 - 2.5]). The group  $\mathfrak{A}$  has the following form.

First consider the special case  $p = 2$  and  $\sigma = \lambda - 2$ . If  $\lambda = 4$ , then  $\mathfrak{A} \cong \mathbb{Z}/2\mathbb{Z}$ , generated by  $\zeta = 7$ ; for simplicity we say  $\alpha = 1$ . Otherwise,  $\mathfrak{A} = \langle \alpha \rangle \times \langle \zeta \rangle$  with  $\zeta = -1$  and  $\alpha \neq \zeta$  of order 2.

Each element  $a$  of  $\mathfrak{A}$  is represented in `Agrp` by a list `[v, a]`, where  $v$  is a list defined by  $a = \alpha^{v[1]} \zeta^{v[2]}$ .

`Bp` is a list of representatives for the  $\mathfrak{A}$ -orbits on  $M^\times$ , which correspond to a basis for the  $\text{SL}_2(\mathbb{Z}/p^\lambda\mathbb{Z})$ -invariant subspace associated to any primitive character  $\chi \in \widehat{\mathfrak{A}}$  with  $\chi^2 \neq 1$ . For other characters, we must use different bases which are particular to each case.

`Char(i, j)` converts two integers  $i, j$  to a function representing a character of  $\mathfrak{A}$ . Each character in  $\widehat{\mathfrak{A}}$  is of the form  $\chi_{i,j}$ , given by  $\chi_{i,j}(\alpha^v \zeta^w) = \mathbf{e}(\frac{vi}{\lambda}) \mathbf{e}(\frac{wj}{\lambda})$ . Note that  $i$  is irrelevant in the cases where  $\mathfrak{A}$  is cyclic.

`IsPrim(chi)` tests whether the output of `Char(i, j)` represents a primitive character. As a special case, for  $p = 2$ ,  $\lambda \geq 5$ ,  $\sigma = \lambda - 2$ , a character is primitive if  $\chi(\alpha) = -1$  (this case differs from the usual definition of primitivity; see [NW76, Section 2.5]). For all other cases, a character is primitive if it is injective on  $\langle o \rangle \leq \mathfrak{A}$ , where  $o$  is defined as follows.

First suppose  $p = 2$ . If  $\lambda = 4$ ,  $\sigma = 2$ , then  $o = \alpha$ .

`Nm(a)`, `Ord(a)`, and `Prod(a, b)` are the norm, order, and product functions on  $M$ , respectively.

$c$  is a scalar used in calculating the  $S$ -matrix; namely  $c = \frac{1}{|M|} \sum_{x \in M} \mathbf{e}(Q(x))$ . Note that this is equal to  $S_Q(-1)/\sqrt{|M|}$ , where  $S_Q(-1) = \frac{1}{\sqrt{|M|}} \sum_{x \in M} \mathbf{e}(Q(x))$  is also known as the central charge.

`tM` is a list describing the elements of the group  $M - pM$ . Constructs information about the underlying quadratic module  $(M, Q)$  of type  $R$ , for  $p$  a prime,  $\lambda \geq 1$ . See

`Agrp` describes the elements of  $\mathfrak{A} = \{\varepsilon \in M^\times \mid \text{Nm}(\varepsilon) = 1\}$  (see [NW76, Section 2.3 - 2.5]).

Representatives for the  $\mathfrak{A}$ -orbits on  $M^\times$  can depend on the choice of character, even for primitive characters  $\chi$  with  $\chi^2 \neq 1$ . Thus, we cannot provide them here, and they are instead calculated by `SL2IrrepR` (5.3.2).

`Char(i, j)` converts the `chi_index` used in `SL2IrrepR` (5.3.2) to a function.

`IsPrim(chi)` tests whether a given character (e.g. from `Char`) is primitive.

`Nm(a)`, `Ord(a)`, and `Prod(a, b)` are the norm, order, and product functions on  $M$ , respectively.

$c$  is a scalar used in calculating the  $S$ -matrix; namely  $c = \frac{1}{|M|} \sum_{x \in M} \mathbf{e}(Q(x))$ .

`tM` is the group  $M - pM$ .

### 5.3.2 SL2IrrepR

▷ `SL2IrrepR(p, ld, sigma, r, t, chi_index)`

(function)

**Returns:** a list of lists of the form  $[S, T]$

Constructs the modular data for the irreducible representation(s) of type  $R$  with parameters  $p, \lambda, \sigma, r, t$  as described in SL2ModuleN (5.2.1), corresponding to the character  $\chi$  indexed by `chi_index = [i, j]` (see the discussions of  $\sigma, r, t$ , and `Char(i, j)` in SL2ModuleN (5.2.1)).

Depending on the parameters,  $W(M, Q)$  will contain either 1 or 2 such irreps.

If  $\sigma = \lambda$  for  $p \neq 2$ , then the second factor of  $M$  is trivial (and hence  $t$  is irrelevant) so this falls through to SL2IrrepUnary (5.3.3).

### 5.3.3 SL2IrrepUnary

▷ SL2IrrepUnary( $p, ld, r$ )

(function)

**Returns:** a list of lists of the form  $[S, T]$

Constructs the modular data for the irreducible representation(s) of unary type  $R$  (that is, the special case where  $\sigma = \lambda$ ) with  $p$  an odd prime,  $\lambda$  a positive integer, and  $r \in \{1, u\}$  with  $u$  a quadratic non-residue mod  $p$ .

In this case,  $W(M, Q)$  always contains exactly 2 such irreps.

# References

- [DLN15] Chongying Dong, Xingjun Lin, and Siu-Hung Ng. Congruence property in conformal field theory. *Algebra Number Theory*, 9(9):2121–2166, 2015. [4](#)
- [Nob76] Alexandre Nobs. Die irreduziblen Darstellungen der Gruppen  $SL_2(\mathbb{Z}_p)$ , insbesondere  $SL_2(\mathbb{Z}_2)$ . I. *Comment. Math. Helv.*, 51(4):465–489, 1976. [4](#)
- [NW76] Alexandre Nobs and Jürgen Wolfart. Die irreduziblen Darstellungen der Gruppen  $SL_2(\mathbb{Z}_p)$ , insbesondere  $SL_2(\mathbb{Z}_p)$ . II. *Comment. Math. Helv.*, 51(4):491–526, 1976. [4](#), [5](#), [6](#), [12](#)

# Index

SL2ChiST, [9](#)  
SL2ExceptionalIrreps, [8](#)  
SL2IrrepD, [10](#)  
SL2IrrepN, [11](#)  
SL2IrrepPositionTest, [9](#)  
SL2IrrepR, [12](#)  
SL2IrrepRUnary, [13](#)  
SL2IrrepsOfDegree, [7](#)  
SL2IrrepsOfDegreeAtMost, [7](#)  
SL2ModuleD, [10](#)  
SL2ModuleN, [11](#)  
SL2ModuleR, [11](#)  
SL2PrimePowerIrrepsOfDegree, [7](#)  
SL2PrimePowerIrrepsOfDegreeAtMost, [7](#)  
SL2PrimePowerIrrepsOfLevel, [7](#)  
SL2WithConjClasses, [9](#)