EXPLANATION OF CODE

ALEXANDER HAZELTINE, BAIYING LIU, AND CHI-HENG LO

Contents

1. Introduction

Local Arthur packets were introduced by Arthur in order to classify the local components of automorphic representations occurring in the discrete spectrum of square integrable automorphic forms ([Art13]). These packets can be computed using Mœglin's parameterization of local Arthur packets ([Mœ06a, Mœ06b, Moe09a, Mœ10, Mœ11a]) and Atobe's refinement ([Ato22a]). Unlike local L-packets, local Arthur packets often have nontrivial intersections with each other. This complicates their study and introduces a desire to understand these intersections. For symplectic and split odd special orthogonal groups over a p-adic field, these intersections are understood by the recent works [Ato22b] and [HLL23a] independently using different methods.

The authors developed Sage code in conjunction with [HLL23a] to compute intersections of local Arthur packets. In this document, we explain the commands of the code and give examples. A large portion of the code is built upon

Date: November 12, 2023.

2000 *Mathematics Subject Classification.* Primary 11F70, 22E50; Secondary 11F85.

Atobe's code which is available at https://github.com/atobe31/Local-A-packets with detailed explanation. To load the code, use the following command.

```
sage: load("packet_HLL.sage")
```

Below is the list of the commands which we explain. The commands in blue are defined by Atobe. We omit the explanation of the commands from Atobe (e.g. those for derivatives) that are not necessary for computing the intersection of local Arthur packets. The explanation of those commands can be found in Atobe's code.

- `Lpacket(phi)`, see §2.2;
- `Lpar_rep(pi)` see §2.2;
- `psi_diag(psi)`, see §2.3;
- `psi_hat(psi)`, see §2.3;
- `phi_psi(psi)`, see §2.3;
- `Symbol(E)`, see §3.1;
- `psi_E(E)`, see §3.1;
- `nonzero(E)`, see §3.1;
- `rep(E)`, see §3.1;
- `Packet(psi,+1)`, see §3.1;
- `change(E,i)`, see §3.2;
- `rearrange(E)`, see §3.2;
- `admissible_order(E)`, see §3.2;
- `dual(E)`, see §3.3;
- `ui(E,i)`, see §3.4;
- `ui_ij(E,i,j)`, see §3.4;
- `UI_inverse_set(E)`, see §3.4;
- `partialdual(E,k)`, see §3.5;
- `E_can(E)`, see §3.6;
- `Packets_E(E)`, see §3.6;
- `Atype_rep(pi)`, see §3.7;
- `Psi_pi_1(pi)`, see §3.7;
- `Psi_pi_2(pi)`, see §3.7;
- `in_Lpacket(E)`, see §3.8;
- `E_max(E)`, see §3.9;
- `E_min(E)`, see §3.9;

2. Representations, L-parameters, and Local Arthur parameters of classical groups

Let F be a p-adic field. In this section, we explain the commands related to local Arthur parameters and L-parameters of the split classical groups $G_n = \mathrm{Sp}_{2n}(F)$ or $\mathrm{SO}_{2n+1}(F)$.

2.1. L-parameters of G_n.

An L-parameter ϕ of G_n is a homomorphism

$$\phi : W_F \times SL_2(\mathbb{C}) \to \widehat{G}_n(\mathbb{C}).$$

Composing with with the embedding $\mathrm{Sp}_{2n}(\mathbb{C}) \hookrightarrow \mathrm{GL}_{2n}(\mathbb{C})$ or $\mathrm{SO}_{2n+1}(\mathbb{C}) \hookrightarrow \mathrm{GL}_{2n+1}(\mathbb{C})$, the L-parameter ϕ decomposes as

$$\phi = \bigoplus_{i \in I} \rho_i |\cdot|^{b_i} \otimes S_{a_i},$$

where ρ_i corresponds to a unitary supercuspidal representations of general linear groups, $b_i \in \mathbb{R}$, and S_{a_i} is the unique a_i-dimensional irreducible representation of $\mathrm{SL}_2(\mathbb{C})$.

In the code, we only deal with the case that ϕ is **unipotent** and of **good parity**. To be explicit, we assume ϕ is trivial on the inertial subgroup of W_F and write the decomposition as

$$\phi = \bigoplus_{i=0}^{t-1} |\cdot|^{b_i} \otimes S_{a_i}.$$

We require that

- If $G_n = \mathrm{Sp}_{2n}(F)$, then each $2b_i + a_i$ is an even integer.
- If $G_n = \mathrm{SO}_{2n+1}(F)$, then each $2b_i + a_i$ is an odd integer.

We record this L-parameter by the tuple

$$((a_0, b_0), \ldots, (a_{t-1}, b_{t-1})).$$

For example, the following L-parameter of $\mathrm{SO}_5(F)$

$$\phi = |\cdot|^{-1/2} \otimes S_1 + |\cdot|^0 \otimes S_2 + |\cdot|^{1/2} \otimes S_1$$

is recorded as follows in the code.

```
sage: phi=((1,-1/2),(2,0),(1,1/2))
sage: phi
((1, -1/2), (2, 0), (1, 1/2))
```

2.2. **Enhanced L-parameters and representations of G_n.** In this subsection, the following commands are explained:

- `Lpacket(phi)`: the L-packet of $\phi \in \Phi(G_n)$;
- `Lpar_rep(pi)`: the L-parameter of $\pi \in \Pi(G_n)$.

Let $\Phi(G_n)$ denote the equivalence class of L-parameters of G_n, and let $\Phi^{\mathfrak{e}}(G_n)$ denote the collection of enhanced L-parameters, which are pairs (ϕ, ε) where $\phi \in \Phi(G_n)$ and $\varepsilon \in \widehat{\mathcal{S}}_\phi$, and

$$\mathcal{S}_\phi = \pi_0(\mathrm{Cent}(\mathrm{im}(\phi), \widehat{G}_n(\mathbb{C})/Z(\widehat{G}_n(\mathbb{C}))).$$

Also let $\Phi_{temp}(G_n)$ denote the equivalence class of tempered L-parameters of G_n.

Let $\Pi(G_n)$ denote the isomorphism class of G_n and let $\Pi_{temp}(G_n)$ denote its subset of tempered representations. The Local Langlands Correspondence states that there is a one-to-one correspondence

$$\begin{aligned} \{(\phi, \varepsilon) \in \Phi^{\mathfrak{e}}(G_n) \mid \phi \in \Phi_{temp}(G_n)\} &\to \Pi_{temp}(G_n), \\ (\phi, \varepsilon) &\to \pi(\phi, \varepsilon). \end{aligned}$$

In the following, we explain how we record a unipotent tempered enhanced L-parameters of G_n, which parametrize all unipotent tempered representations of G_n.

Let ϕ be a unipotent tempered L-parameter of G_n of good parity and write

$$\phi = \bigoplus_{i=0}^{t-1} \mathbf{1} \otimes S_{a_i}.$$

For G_n, the component group $\mathcal{S}_\phi$ is always an abelian 2-group, and each character of the component group $\mathcal{S}_\phi$ corresponds to a tuple of signs $(\varepsilon_0, \ldots, \varepsilon_{t-1})$ satisfying the following conditions.

- If $a_i = a_j$, then $\varepsilon_i = \varepsilon_j$.
- $\prod_{i \in I_2} \varepsilon_i = 1$.

Thus, in the code, we record the enhanced L-parameter $(\phi, (\varepsilon_0, \ldots, \varepsilon_{t-1}))$ by the tuple

$$([x_0, \varepsilon_0], \ldots, [x_{t-1}, \varepsilon_{t-1}]),$$

where $x_i = \frac{a_i - 1}{2}$, and we use the same tuple to represent the corresponding unipotent tempered representation. For example:

```
sage: T=([0,-1],[1,1],[2,-1])
sage: T
([0, -1], [1, 1], [2, -1])
```

This records the enhanced L-parameter $(\mathbf{1} \otimes S_1 + \mathbf{1} \otimes S_3 + \mathbf{1} \otimes S_5, (-1, +1, -1))$, which corresponds to the unique supercuspidal representation in this tempered L-packet.

We call $\pi \in \Pi(G_n)$ unipotent of good parity if its standard module is of the form

$$\Delta[x_1, y_1] \times \cdots \times \Delta[x_f, y_f] \rtimes \pi(\phi, \varepsilon),$$

where

- $x_1 + y_1 \leq \cdots \leq x_f + y_f < 0$,
- $x_i \in \mathbb{Z}$ if $G_n = \mathrm{Sp}_{2n}(F)$ and $x_i \in \frac{1}{2} + \mathbb{Z}$ if $G_n = \mathrm{SO}_{2n+1}(F)$,
- ϕ is tempered, unipotent and of good parity.

In the code, we record the above irreducible unipotent representation of good parity by the pair of tuples

$$(([x_1, y_1], \ldots, [x_f, y_f]), ([x_{f+1}, \varepsilon_{f+1}], \ldots, [x_t, \varepsilon_t])),$$

where the tuple $([x_{f+1}, \varepsilon_{f+1}], \ldots, [x_t, \varepsilon_t]))$ corresponds to the unipotent tempered representation $\pi(\phi, \varepsilon)$. For example:

```
sage: pi=(([0, -2],), ([0, -1], [0, -1], [1, 1])) # A
    representation of $\Sp_{10}(F)$.
sage: Lpar_rep(pi)
((1, 0), (1, 0), (3, -1), (3, 0), (3, 1)) # The $L$-
    parameter of the representation $\pi$.
```

If $\pi \in \Pi(G_n)$ is tempered, then we leave the first tuple empty. For example:

```
sage: pi=((), ([1/2, -1], [3/2, 1], [5/2, -1])) # A
    tempered supercuspidal representation of $\SO_{13}(F)$.
sage: Lpar_rep(pi)
((2, 0), (4, 0), (6, 0)) # The $L$-parameter of the
    representation $\pi$.
```

For each unipotent L-parameter ϕ of G_n of good parity, we can list the representations in the L-packet Π_ϕ. For example:

```
sage: phi=((1, -1/2), (2, 0), (2, 0), (1, 1/2)) # An $L$-parameter of $\SO_{7}(F)$.
sage: Lpacket(phi) # The $L$-packet of $\phi$.
[(([-1/2, -1/2],), ([1/2, 1], [1/2, 1])),
(([-1/2, -1/2],), ([1/2, -1], [1/2, -1]))]
```

This lists the two representations in the L-packet of

$$\phi = |\cdot|^{-1/2} \otimes S_1 + \mathbf{1} \otimes S_2 + \mathbf{1} \otimes S_2 + |\cdot|^{1/2} \otimes S_1.$$

2.3. **Local Arthur parameter of G_n.** In this subsection, the following commands are explained:

- `psi_diag(psi)`: the diagonal restriction of ψ;
- `psi_hat(psi)`: the dual of ψ;
- `phi_psi(psi)`: the L-parameter associated to ψ.

A local Arthur parameter of G_n is a homomorphism

$$\psi : W_F \times \mathrm{SL}_2(\mathbb{C}) \times \mathrm{SL}_2(\mathbb{C}) \to \widehat{G_n}(\mathbb{C}).$$

Composing with the embedding $\mathrm{Sp}_{2n}(\mathbb{C}) \hookrightarrow \mathrm{GL}_{2n}(\mathbb{C})$ or $\mathrm{SO}_{2n+1}(\mathbb{C}) \hookrightarrow \mathrm{GL}_{2n+1}(\mathbb{C})$, the local Arthur parameter ψ decomposes as

$$\psi = \bigoplus_{\rho} \bigoplus_{i \in I_\rho} \rho \otimes S_{a_i} \otimes S_{b_i},$$

and ψ is uniquely determined by this decomposition.

In this code, we only deal with local Arthur parameters of G_n that are **unipotent** and of good parity. That is we assume $\psi|_{W_F}$ is trivial. Write

$$\psi = \bigoplus_{i=1}^{t} \mathbf{1} \otimes S_{a_i} \otimes S_{b_i},$$

we require that a_i+b_i is even if $G_n = \mathrm{Sp}_{2n}(F)$ and a_i+b_i is odd if $G_n = \mathrm{SO}_{2n+1}(F)$. In the code, we record this parameter by the tuple of pairs

$$((a_1, b_1), \ldots, (a_t, b_t)).$$

From ψ, we define ψ^Δ, the diagonal restriction of ψ, and $\widehat{\psi}$, the dual of ψ by

$$\psi^\Delta(w, x, y) := \psi(w, x, x),$$
$$\widehat{\psi}(w, x, y) := \psi(w, y, x).$$

We also define ϕ_ψ, the L-parameter ϕ_ψ by

$$\phi_\psi(w, x) := \psi\left(w, x, \begin{pmatrix} |w|^{1/2} & \\ & |w|^{-1/2} \end{pmatrix}\right).$$

For example:

```
sage: psi=((1,6),(2,1),(4,1)) # A local Arthur parameter of $\SO_{13}(F)$.
sage: psi_diag(psi) # The diagonal restriction of $\psi$.
((2, 1), (4, 1), (6, 1))
```

```
sage: psi_hat(psi) # The dual of $\psi$.
((1, 2), (1, 4), (6, 1))
sage: phi_psi(psi) # The $L$-parameter associated to $\psi$
((1, -5/2), (1, -3/2), (1, -1/2), (1, 1/2), (1, 3/2), (1, 5/2), (2, 0), (4, 0))
```

3. Extended multi-segments

In this section, we explain the commands related to extended multi-segments.

3.1. Basic definition. In this subsection, the following commands are explained:

- `Symbol(E)`: the pictograph associated to an extended multi-segment $\mathcal{E}$;
- `psi_E(E)`: the local Arthur parameter associated to an extended multi-segment $\mathcal{E}$;
- `nonzero(E)`: the non-vanishing criterion of $\pi(\mathcal{E})$;
- `rep(E)`: the representation associated to $\mathcal{E}$;
- `Packet(psi,+1)`: the local Arthur packet associated to ψ.

The commands `nonzero(E)`, `rep(E)` and `Packet(psi,+1)` are defined by Atobe.

To compute local Arthur packets, Atobe introduced extended multi-segments.

Definition 3.1 ([Ato20a, Definition 3.1])**.** *(Extended multi-segments)*

(1) An **extended segment** *is a triple* $([A,B]_\rho, l, \eta)$*, where*
- $[A,B]_\rho = \{\rho|\cdot|^A, \rho|\cdot|^{A-1}, \ldots, \rho|\cdot|^B\}$ *is a segment for an irreducible unitary supercuspidal representation* ρ *of some* $\mathrm{GL}_d(F)$*;*
- $l \in \mathbb{Z}$ *with* $0 \le l \le \frac{b}{2}$*, where* $b = \#[A,B]_\rho = A - B + 1$*;*
- $\eta \in \{\pm 1\}$.

(2) An **extended multi-segment** *for* G_n *is an equivalence class (via the equivalence defined below) of multi-sets of extended segments*

$$\mathcal{E} = \cup_\rho \{([A_i,B_i]_\rho, l_i, \eta_i)\}_{i\in(I_\rho,>)}$$

such that
- I_ρ *is a totally ordered finite set with a fixed admissible total order* $>$*, that is,* $>$ *satisfies the following condition.*

  For $i,j \in I_\rho$*, if* $A_i > A_j$ *and* $B_i > B_j$*, then* $i > j$*;*
- $A_i + B_i \ge 0$ *for all* ρ *and* $i \in I_\rho$*;*
- *as a representation of* $W_F \times \mathrm{SL}_2(\mathbb{C}) \times \mathrm{SL}_2(\mathbb{C})$*,*

$$\psi_{\mathcal{E}} = \bigoplus_\rho \bigoplus_{i\in I_\rho} \rho \otimes S_{a_i} \otimes S_{b_i}$$

  where $(a_i,b_i) = (A_i+B_i+1, A_i-B_i+1)$*, is a local Arthur parameter for* G_n *of good parity. We shall denote* $\psi_{\mathcal{E}}$ *the local Arthur parameter associated with* $\mathcal{E}$*.*
- *The sign condition*

$$\prod_\rho \prod_{i\in I_\rho} (-1)^{[\frac{b_i}{2}]+l_i} \eta_i^{b_i} = 1 \tag{3.1}$$

  holds.

(3) Two extended segments $([A,B]_\rho, l, \eta)$ and $([A',B']_{\rho'}, l', \eta')$ are **weakly equivalent** *if*
- $[A,B]_\rho = [A',B']_{\rho'}$;
- $l = l'$; *and*
- $\eta = \eta'$ *whenever* $l = l' < \frac{b}{2}$.

Two extended multi-segments $\mathcal{E} = \cup_\rho \{([A_i,B_i]_\rho, l_i, \eta_i)\}_{i \in (I_\rho, >)}$ *and* $\mathcal{E}' = \cup_\rho \{([A'_i,B'_i]_\rho, l'_i, \eta'_i)\}_{i \in (I_\rho, >)}$ *are* **weakly equivalent** *if for any* ρ *and* $i \in I_\rho$*, the extended segments* $([A_i,B_i]_\rho, l_i, \eta_i)$ *and* $([A'_i,B'_i]_\rho, l'_i, \eta'_i)$ *are weakly equivalent.*

In the code, we assume the associated local Arthur parameter $\psi_{\mathcal{E}}$ is unipotent. Thus $\mathcal{E}$ is of the form

$$\mathcal{E} = \{[A_i,B_i]_\rho, l_i, \eta_i\}_{i \in (I_\rho, >)}.$$

We identify $(I_\rho, >)$ as $\{0, 1, \dots, |I_\rho| - 1\}$, where

$$0 < 1 < \cdots < |I_\rho| - 1.$$

For each extended multi-segment, we attach a pictograph to each extended multi-segment as in [Ato22a, Section 3]. We give an example to explain this.

Example 3.2. *Let* ρ *be the trivial representation. The pictograph*

$$\mathcal{E} = \begin{array}{c} \begin{array}{cccccc} \frac{-5}{2} & \frac{-3}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{3}{2} & \frac{5}{2} \end{array} \\ \left(\begin{array}{cccccc} \lhd & \lhd & \oplus & \ominus & \rhd & \rhd \\ & & & \oplus & & \\ & & & & \ominus & \end{array} \right)_\rho \end{array}$$

corresponds to the extended multi-segment $\mathcal{E} = \{([A_i,B_i]_\rho, l_i, \eta_i)\}_{i \in (0<1<2)}$ *of* $\mathrm{SO}_{13}(F)$ *where*
- $([A_0,B_0]_\rho, [A_1,B_1]_\rho, [A_2,B_2]_\rho) = ([\frac{-5}{2}, \frac{5}{2}]_\rho, [\frac{1}{2}, \frac{1}{2}]_\rho, [\frac{3}{2}, \frac{3}{2}]_\rho)$ *specifies the "support" of each row.*
- $(l_0, l_1, l_2) = (2, 0, 0)$ *counts the number of pairs of triangles in each row.*
- $(\eta_0, \eta_1, \eta_2) = (1, 1, -1)$ *records the sign of the first circle in each row.*

The associated local Arthur parameter is

$$\psi_{\mathcal{E}} = \rho \otimes S_1 \otimes S_6 + \rho \otimes S_2 \otimes S_1 + \rho \otimes S_4 \otimes S_1.$$

For each extended multi-segment $\mathcal{E}$ of G_n, Atobe associated a representation $\pi(\mathcal{E})$ of G_n ([Ato22a, §3.2]), which is either irreducible or zero. Atobe also gave a non-vanishing criterion in [Ato22a, §4]. We demonstrate these commands for the extended multi-segment in Example 3.2 below. We remark that our command `symbol(E)` is slightly different from the one in Atobe's code.

```
sage: E = (([5/2, -5/2], 2, 1), ([1/2, 1/2], 0, 1), ([3/2,
    3/2], 0, -1)) #An extended multi-segment.
sage: symbol(E) # The pictograph associated to $E$.
+------+------+------+-----+-----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+------+------+------+-----+-----+-----+
|  <   |  <   |  +   |  -  |  >  |  >  |
+------+------+------+-----+-----+-----+
|      |      |      |  +  |     |     |
```

```
+------+------+------+-----+-----+-----+
|      |      |      |     |  -  |     |
+------+------+------+-----+-----+-----+
sage: psi_E(E) # The local Arthur parameter associated to
   $E$.
((1, 6), (2, 1), (4, 1))
sage: nonzero(E,1) # Is $\pi(E)\neq 0?$
True
sage: rep(E) # The representation associated to $E$,
((), ([1/2, -1], [3/2, 1], [5/2, -1]))
```

The output of the command `rep(E)` shows that the representation $\pi(\mathcal{E})$ associated to the extended multi-segment in Example 3.2 is tempered.

Remark 3.3. *The command* `rep(E)` *computes the correct representation $\pi(\mathcal{E})$ if the admissible order of $\mathcal{E}$ on I_ρ satisfies the following condition*

$$\text{(P')} \qquad : \textit{For } i, j \in I_\rho, \textit{ if } B_i > B_j, \textit{ then } i > j.$$

The reason is that if above property fails, the definition of $\pi(\mathcal{E})$ may involve non-highest derivatives, which is not included in the code. However, we may always find an admissible order $\gg$ on I_ρ satisfying the property (P') (see §3.2). Then $\pi(\mathcal{E}) = \pi(\mathcal{E}_\gg)$, and the latter is computable by code.

Atobe proved that ([Ato22a, Theorem 3.4]) the local Arthur packet Π_ψ of a good parity local Arthur parameter ψ can be constructed by the multi-segments as follows:

$$\Pi_\psi = \{\pi(\mathcal{E}) \mid \psi_\mathcal{E} = \psi\} \setminus \{0\}.$$

We give an example of the construction in code.

```
sage: psi=((1, 6), (2, 1), (4, 1)) # A local Arthur
   parameter of $\SO_{13}(F)$.
sage: P=Packet(psi,1) # The local Arthur packet associated
    to $\psi$.
sage: len(P) # The cardinality of the local Arthur packet
   $\Pi_{\psi}$.
4
sage: for E in P:
....:     symbol(E)
....:     rep(E)
....:
+------+------+------+-----+-----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+------+------+------+-----+-----+-----+
|  <   |  <   |  <   |  >  |  >  |  >  |
+------+------+------+-----+-----+-----+
|      |      |      |  +  |     |     |
+------+------+------+-----+-----+-----+
|      |      |      |     |  +  |     |
+------+------+------+-----+-----+-----+
```

```
(([-5/2, -5/2], [-3/2, -3/2], [-1/2, -1/2]), ([1/2, 1],
    [3/2, 1]))
+-----+-----+-----+----+----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+-----+-----+-----+----+----+-----+
|  <   |  <   |  +   |  -  |  >  |  >  |
+-----+-----+-----+----+----+-----+
|      |      |      |  -  |     |     |
+-----+-----+-----+----+----+-----+
|      |      |      |     |  +  |     |
+-----+-----+-----+----+----+-----+
(([-5/2, -5/2], [-3/2, -3/2]), ([1/2, -1], [1/2, -1],
    [3/2, 1]))
+-----+-----+-----+----+----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+-----+-----+-----+----+----+-----+
|  <   |  <   |  +   |  -  |  >  |  >  |
+-----+-----+-----+----+----+-----+
|      |      |      |  +  |     |     |
+-----+-----+-----+----+----+-----+
|      |      |      |     |  -  |     |
+-----+-----+-----+----+----+-----+
((), ([1/2, -1], [3/2, 1], [5/2, -1]))
+-----+-----+-----+----+----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+-----+-----+-----+----+----+-----+
|  <   |  <   |  <   |  >  |  >  |  >  |
+-----+-----+-----+----+----+-----+
|      |      |      |  -  |     |     |
+-----+-----+-----+----+----+-----+
|      |      |      |     |  -  |     |
+-----+-----+-----+----+----+-----+
(([-5/2, -5/2], [-3/2, -3/2], [-1/2, -1/2]), ([1/2, -1],
    [3/2, -1]))
```

3.2. **Row exchanges.** In this subsection, the following commands are explained:

- `change(E,i)`: swapping indices i and $i+1$ in $\mathcal{E}$;
- `rearrange(E)`: change to an admissible order satisfying (P'), see Remark 3.3.
- `admissible_order(E)`: The set of all extended multi-segments obtained from $\mathcal{E}$ by changing admissible orders.

The command `change(E,i)` is defined by Atobe.

Let $\mathcal{E} = \{[A_i, B_i], l_i, \eta_i\}_{i\in(I_\rho,>)}$ be an extended multi-segment. Suppose $\gg$ is another admissible order on I_ρ (see Definition 3.1(2)). Then there exists a unique extended multi-segment

$$\mathcal{E}_{\gg} = \{[A_i', B_i'], l_i', \eta_i'\}_{i\in(I_\rho,\gg)}$$

such that $\pi(\mathcal{E}_{\gg}) = \pi(\mathcal{E})$. In [Xu21b], Xu gave an algorithm to describe the change of Mœglin's parametrization of local Arthur packets due to the change of admissible order. Then the algorithm is rephrased in terms of extended multi-segment in [Ato22a, §4.2].

Suppose $[A_{i+1}, B_{i+1}] \supseteq [A_i, B_i]$ or $[A_{i+1}, B_{i+1}] \subseteq [A_i, B_i]$, then the total order $>_i$ on $I_\rho = \{0 < \cdots < n-1\}$ given by

$$0 <_i 1 <_i \cdots <_i i-1 <_i i+1 <_i i <_i i+2 <_i \cdots <_i n-1$$

is also admissible. We define

$$R_i(\mathcal{E}) = \mathcal{E}_{>_i},$$

and then again identify $(I_\rho, >_i) = \{0 < \cdots < n\}$. Here is an example.

```
sage: E=(([2, 0], 1, -1), ([1, 1], 0, 1), ([3, 3], 0, -1))
+---+---+---+---+
| 0 | 1 | 2 | 3 |
+---+---+---+---+
| < | - | > |   |
+---+---+---+---+
|   | + |   |   |
+---+---+---+---+
|   |   |   | - |
+---+---+---+---+
sage: change(E,0) # Swap indices 0 and 1.
(([1, 1], 0, 1), ([2, 0], 0, 1), ([3, 3], 0, -1))
sage: symbol(change(E,0)) # The pictograph associated to $R_0(E)$.
+---+---+---+---+
| 0 | 1 | 2 | 3 |
+---+---+---+---+
|   | + |   |   |
+---+---+---+---+
| + | - | + |   |
+---+---+---+---+
|   |   |   | - |
+---+---+---+---+
sage: nonzero(change(E,0),1) # Is the representation associated to $R_0(E)$ nonzero?
True
sage: rep(change(E,0)) # The representation associated to $R_0(E)$.
((), ())
sage: rep(rearrange(change(E,0))) # The representation associated to $R_0(E)_{>}$ for some admissible order $>$ satisfying (P').
(([0, -1],), ([1, -1], [2, 1], [3, -1]))
sage: rearrange(change(E,0))==E # Is $rearrange_prime(E)=E$?
True
```

See Remark 3.3 for the explanation why the code outputs $\pi(R_0(\mathcal{E}))=0$.

Note that for an arbitrary admissible order $\gg$, we have

$$\mathcal{E}_{\gg} = R_{i_1} \circ \cdots \circ R_{i_s}(\mathcal{E})$$

for some sequence of indices $(i_1, \ldots, i_s)$. The command `admissible_order(E)` outputs the list of $\mathcal{E}_{\gg}$ where $\gg$ ranges over all admissible orders. Here is an example.

```
sage: E = (([5/2, -5/2], 2, 1), ([1/2, 1/2], 0, 1), ([3/2,
    3/2], 0, -1))
sage: len(admissible_order(E)) # Cardinality of
   admissible_order(E)
3
sage: for E1 in admissible_order(E):
....:     symbol(E1)
....:
+------+------+------+-----+-----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+------+------+------+-----+-----+-----+
|  <   |  <   |  +   |  -  |  >  |  >  |
+------+------+------+-----+-----+-----+
|      |      |      |  +  |     |     |
+------+------+------+-----+-----+-----+
|      |      |      |     |  -  |     |
+------+------+------+-----+-----+-----+
+------+------+------+-----+-----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+------+------+------+-----+-----+-----+
|      |      |      |  -  |     |     |
+------+------+------+-----+-----+-----+
|  <   |  -   |  +   |  -  |  +  |  >  |
+------+------+------+-----+-----+-----+
|      |      |      |     |  -  |     |
+------+------+------+-----+-----+-----+
+------+------+------+-----+-----+-----+
| -5/2 | -3/2 | -1/2 | 1/2 | 3/2 | 5/2 |
+------+------+------+-----+-----+-----+
|      |      |      |  -  |     |     |
+------+------+------+-----+-----+-----+
|      |      |      |     |  +  |     |
+------+------+------+-----+-----+-----+
|  +   |  -   |  +   |  -  |  +  |  -  |
+------+------+------+-----+-----+-----+
```

3.3. **Aubert-Zelevinsky dual.**

In this subsection, the following commands are explained:

- `dual(E)`: The Aubert-Zelevinsky dual of an extended multi-segment $\mathcal{E}$.

This command is defined by Atobe.

Let π be an irreducible representation of G_n. In [Aub95], Aubert showed that there exists $\varepsilon \in \{\pm 1\}$ such that

$$\widehat{\pi} := \varepsilon \sum_{P} (-1)^{\dim(A_P)} [\mathrm{Ind}_P^{G_n}(Jac_P(\pi))]$$

gives an irreducible representation. Here the sum is over all standard parabolic subgroups P of G_n and A_P is the maximal split torus of the center of the Levi subgroup of P. We say $\widehat{\pi}$ is the Aubert-Zelevinsky dual or Aubert-Zelevinsky involution of π.

Given an extended multi-segment $\mathcal{E}$ whose admissible order satisfying (P'), Atobe defined another extended multi-segment $dual(\mathcal{E})$ such that $\pi(dual(\mathcal{E}))$ is the Aubert-Zelevinsky dual of $\pi(\mathcal{E})$ ([Ato22a, Definition 6.1, Theorem 6.2]) and $\psi_{dual(\mathcal{E})} = \widehat{\psi_{\mathcal{E}}}$. The operator $dual(\mathcal{E})$ is implemented in the code as follows. We remark that this command is named `hat(E)` in Atobe's code.

```
sage: E=(([1,-1],1,-1),([0,0],0,1),([2,2],0,-1))
sage: symbol(E)
+----+---+---+---+
| -1 | 0 | 1 | 2 |
+----+---+---+---+
| <  | - | > |   |
+----+---+---+---+
|    | + |   |   |
+----+---+---+---+
|    |   |   | - |
+----+---+---+---+
sage: dual(E) # The Aubert-Zelevinsky dual of $E$.
(([2, -2], 2, -1), ([0, 0], 0, 1), ([1, 1], 0, -1))
sage: symbol(dual(E))
+----+----+---+---+---+
| -2 | -1 | 0 | 1 | 2 |
+----+----+---+---+---+
| <  | <  | - | > | > |
+----+----+---+---+---+
|    |    | + |   |   |
+----+----+---+---+---+
|    |    |   | - |   |
+----+----+---+---+---+
```

3.4. **union-intersection.** In this subsection, the following commands are explained:

- `ui(E,i)`
- `ui_ij(E,i,j)`;
- `UI_inverse_set(E)`;

Following [Ato22a, Corollary 5.3], we defined operators $ui_k, ui_{i,j}$ on extended multi-segments (see [HLL23a, Definitions 3.23, 5.1]) that preserves the associated representations. When applicable, the operator ui_k (resp. $ui_{i,j}$) changes the k-th and $k+1$-th (resp. i-th and j-th) segments by their union and intersection. Here is an example.

```
sage: E=(([2, -2], 2, -1), ([1, -1], 1, 1), ([3, 1], 1,
    -1), ([4, 2], 1, 1))
sage: symbol(E)
+----+----+---+---+---+---+---+
| -2 | -1 | 0 | 1 | 2 | 3 | 4 |
+----+----+---+---+---+---+---+
| <  | <  | - | > | > |   |   |
+----+----+---+---+---+---+---+
|    | <  | + | > |   |   |   |
+----+----+---+---+---+---+---+
|    |    |   | < | - | > |   |
+----+----+---+---+---+---+---+
|    |    |   |   | < | + | > |
+----+----+---+---+---+---+---+
sage: symbol(ui(E,2)) # Apply union-intersection to the
    indices (2,3).
+----+----+---+---+---+---+---+
| -2 | -1 | 0 | 1 | 2 | 3 | 4 |
+----+----+---+---+---+---+---+
| <  | <  | - | > | > |   |   |
+----+----+---+---+---+---+---+
|    | <  | + | > |   |   |   |
+----+----+---+---+---+---+---+
|    |    |   | < | - | + | > |
+----+----+---+---+---+---+---+
|    |    |   |   | < | > |   |
+----+----+---+---+---+---+---+
sage: symbol(ui_ij(ui(E,2),0,2)) # Further apply union-
    intersection to the indices (0,2).
+----+----+---+---+---+---+---+
| -2 | -1 | 0 | 1 | 2 | 3 | 4 |
+----+----+---+---+---+---+---+
| <  | <  | - | + | - | > | > |
+----+----+---+---+---+---+---+
|    | <  | + | > |   |   |   |
+----+----+---+---+---+---+---+
|    |    |   | < | > |   |   |
+----+----+---+---+---+---+---+
|    |    |   |   | < | > |   |
+----+----+---+---+---+---+---+
sage: ui_ij(E,0,3)!=E # Is $ui_{0,3}$ applicable on $E$?
False
```

Union and intersection defines a preorder on the set of multi-segments (see [HLL23a, Definition 5.11]). Specifically, for extended multi-segments $\mathcal{E}, \mathcal{E}'$, we define $\mathcal{E} \leq \mathcal{E}'$ if $\mathcal{E}$ can be obtained from $\mathcal{E}'$ by successively applying $ui_{i,j}$ and row exchange R_k.

In [HLL23a, Algorithm 5.15], we give an algorithm to compute the set

$$UI^{-1}(\mathcal{E}) := \{\mathcal{E}' \mid \mathcal{E} \leq \mathcal{E}'\}/(\text{row exchanges}).$$

A variant of this algorithm is implemented in the command `UI_inverse_set(E)`. It gives a set containing $UI^{-1}(\mathcal{E})$ since it includes $dual \circ ui \circ dual$ of type 3' in Step 2 of [HLL23a, Algorithm 5.15]. Here is an example.

```
sage: E=(([2,0],0,-1),([1,1],0,-1),([3,3],0,-1))
sage: P=UI_inverse_set(E) # The inverse image of $E$ under
    various ui.
sage: len(P) # The cardinality of $P$.
3
sage: for E1 in P:
....:     symbol(E1)
....:
+---+---+---+---+
| 0 | 1 | 2 | 3 |
+---+---+---+---+
| - | + | - |   |
+---+---+---+---+
|   | - |   |   |
+---+---+---+---+
|   |   |   | - |
+---+---+---+---+
+---+---+---+---+
| 0 | 1 | 2 | 3 |
+---+---+---+---+
| - | + |   |   |
+---+---+---+---+
|   | < | > |   |
+---+---+---+---+
|   |   |   | - |
+---+---+---+---+
+---+---+---+---+
| 0 | 1 | 2 | 3 |
+---+---+---+---+
| - |   |   |   |
+---+---+---+---+
|   | + |   |   |
+---+---+---+---+
|   | < | > |   |
+---+---+---+---+
|   |   |   | - |
+---+---+---+---+
```

3.5. **Partial dual.** In this subsection, the following commands are explained:

- `partialdual(E,k)`: The partial dual $dual_k^+(\mathcal{E})$ of an extended multi-segment $\mathcal{E}$.

The last operator we need is called the partial dual ([HLL23a, Definition 6.5]). This operator only applies to an extended multi-segment $\mathcal{E}$ if $B_k = \pm\frac{1}{2}$ for some index k. Roughly, its effect sends B_i to $-B_i$. When $B_k = \frac{1}{2}$, we denote the effect of the operator by $dual_k^+(\mathcal{E})$. When $B_k = \frac{-1}{2}$, we denote the effect of the operator by $dual_k^-(\mathcal{E})$. By [HLL23a, Proposition 6.8], if $\pi(\mathcal{E}) \neq 0$ and $dual_k^+$ (resp. $dual_k^-$) is applicable on $\mathcal{E}$, then $\pi(dual^+(\mathcal{E})) = \pi(\mathcal{E})$ (resp. $\pi(dual_k^-(\mathcal{E})) = \pi(\mathcal{E})$). The operator $dual_k^+(\mathcal{E})$ is implemented in the code as follows.

```
sage: E=(([5/2 , 1/2], 0, -1),)
sage: symbol(E)
+-----+-----+-----+
| 1/2 | 3/2 | 5/2 |
+-----+-----+-----+
|  -  |  +  |  -  |
+-----+-----+-----+
sage: partialdual(E,0) # The partial dual $dual_0^+(E)$.
[([5/2, -1/2], 0, 1)]
sage: symbol(partialdual(E,0))
+------+-----+-----+-----+
| -1/2 | 1/2 | 3/2 | 5/2 |
+------+-----+-----+-----+
|  +   |  -  |  +  |  -  |
+------+-----+-----+-----+
```

We remark that one can compute $dual_k^-(\mathcal{E})$ via `dual(partialdual(dual(E),len(E)-k-1))` (although it is not applicable on $\mathcal{E}$ in the above example).

3.6. **Local Arthur packets containing a representation.** In this subsection, the following commands are explained:

- `E_can(E)`: the canonical form of $\mathcal{E}$;
- `Packets_E(E)`: the set $\Psi(\mathcal{E}) := \{\mathcal{E}' \mid \pi(\mathcal{E}') = \pi(\mathcal{E})\}/(\text{row exchanges})$.

In [HLL23a, §7], from an extended multi-segment $\mathcal{E}$, we constructed another extended multi-segment $\mathcal{E}_{can}$ by the operators introduced in §3.2-3.5, called the canonical form of $\mathcal{E}$. It has the property that $\pi(\mathcal{E}) \cong \pi(\mathcal{E}')$ if and only if $\mathcal{E}_{can} = \mathcal{E}'_{can}$ ([HLL23a, Corollary 7.3]). By reversing the construction of $\mathcal{E}_{can}$, we obtain an algorithm to exhaust the set

$$\Psi(\mathcal{E}) = \{\mathcal{E}' \mid \pi(\mathcal{E}') \cong \pi(\mathcal{E})\}/(\text{row exchanges}).$$

For example:

```
sage: E=(([1/2,1/2],0,-1),([1/2,1/2],0,-1),([3/2,3/2],0,1)
    )
sage: symbol(E)
+-----+-----+
| 1/2 | 3/2 |
+-----+-----+
|  -  |     |
```

```
+-----+-----+
|  -  |     |
+-----+-----+
|     |  +  |
+-----+-----+
sage: symbol(E_can(E)) # The pictograph associated to the
    canonical form of $E$.
+-----+-----+
| 1/2 | 3/2 |
+-----+-----+
|  -  |     |
+-----+-----+
|  -  |  +  |
+-----+-----+
sage: P=Packets_E(E) # The set $\Psi(E)$.
sage: len(P) # The cardinality of $P$.
4
sage: for E1 in P:
....:     symbol(E1)
....:     rep(E1)==rep(E) # Is $\pi(E1)=\pi(E)$?
....:     E_can(E1)==E_can(E) # Do the canonical forms of
    $E1$ and $E$ agree?
....:
+-----+-----+
| 1/2 | 3/2 |
+-----+-----+
|  -  |     |
+-----+-----+
|  -  |  +  |
+-----+-----+
True
True
+-----+-----+
| 1/2 | 3/2 |
+-----+-----+
|  -  |     |
+-----+-----+
|  -  |     |
+-----+-----+
|     |  +  |
+-----+-----+
True
True
+------+-----+-----+
| -1/2 | 1/2 | 3/2 |
+------+-----+-----+
|  +   |  -  |     |
```

```
+------+-----+-----+
|      |  -  |  +  |
+------+-----+-----+
True
True
+------+-----+-----+
| -1/2 | 1/2 | 3/2 |
+------+-----+-----+
|  +   |  -  |     |
+------+-----+-----+
|      |  -  |     |
+------+-----+-----+
|      |     |  +  |
+------+-----+-----+
True
True
```

3.7. **Determination of Arthur type representations.** In this subsection, the following commands are explained:

- `Atype_rep(pi)`: determine whether π is of Arthur type;
- `Psi_pi_1(pi)`, compute $\Psi(\pi) = \{\mathcal{E} \mid \pi(\mathcal{E}) = \pi\}$ using [HLL23a, Algorithm 7.6];
- `Psi_pi_2(pi)`, compute $\Psi(\pi)$ using [HLLZ22, Algorithm 5.12];

In [HLL23a, Theorem 7.7], we showed that the canonical form $\mathcal{E}_{can}$ carries the most information of derivatives among all the other members in $\Psi(\mathcal{E})$. This leads to [HLL23a, Algorithm 7.9] that determines whether a representation is of Arthur type or not. In our code, the command `Atype_rep(pi)` outputs the canonical form of among $\{\mathcal{E} \mid \pi(\mathcal{E}) = \pi\}$ if π is of Arthur type. Combining with `Packets_E(E)`, the command `Psi_pi_1(pi)` computes the set

$$\Psi(\pi) = \{\mathcal{E} \mid \pi(\mathcal{E}) = \pi\}/(\text{row exchanges}).$$

We gave a second algorithm in [HLLZ22, Algorithm 5.12] that serves the same function. This is implemented in the command `Psi_pi_2(pi)`. We give an example that demonstrate these two algorithms.

```
sage: phi=((1, -1), (1, 0), (1, 1), (2, -5/2), (2, 5/2), (3, 0), (5, 0))
sage: for pi in Lpacket(phi):
....:     pi
....:     print("---")
....:     Atype_rep(pi) # Output the canonical form if pi is of Arthur type, and False otherwise.
....:     print("---")
....:     Psi_pi_1(pi) # The set $\Psi(\pi)$ using the first algorithm.
....:     Psi_pi_2(pi) # The set $\Psi(\pi)$ using the second algorithm.
....:     print("----------------------")
```

```
....:
(([-2, -3], [-1, -1]), ([0, 1], [1, 1], [2, 1]))
---
False
---
[]
[]
-----------------------
(([-2, -3], [-1, -1]), ([0, -1], [1, -1], [2, 1]))
---
False
---
[]
[]
-----------------------
(([-2, -3], [-1, -1]), ([0, -1], [1, 1], [2, -1]))
---
(([3, -2], 2, -1), ([1, 1], 0, -1))
---
[(([3, -2], 2, -1), ([1, 1], 0, -1)),
 (([3, -3], 3, -1), ([2, -2], 2, 1), ([1, 1], 0, -1))]
[(([3, -2], 2, -1), ([1, 1], 0, -1)),
 (([3, -3], 3, -1), ([2, -2], 2, 1), ([1, 1], 0, -1))]
-----------------------
(([-2, -3], [-1, -1]), ([0, 1], [1, -1], [2, -1]))
---
False
---
[]
[]
-----------------------
```

3.8. ***L*-packet of Arthur type.** In this subsection, the following commands are explained:

- `in_Lpacket(E)`: determine whether $\pi(\mathcal{E}) \in \Pi_{\phi_{\psi_{\mathcal{E}}}}$.

In [HLL23a, §8], we gave a criterion on an extended multi-segment $\mathcal{E}$ to determine whether $\pi(\mathcal{E})$ lies in the L-packet associated to the local Arthur parameter $\psi_{\mathcal{E}}$. It is implemented in the command `in_Lpacket(E)` as follows. Note that `E` is exactly $\mathcal{E}_2$ in [HLL23a, Example 9.7].

```
sage: psi=((1, 3), (3, 3), (5, 1))
sage: P=Packet(psi,1)
sage: len(P)
5
sage: for E in P:
....:     symbol(E)
```

```
....:      in_Lpacket(E) # Is $rep(E)$ in the L-packet
   associated to $psi_E$?
....:    Lpar_rep(rep(E))==phi_psi(psi) # Is the L-
   parameter of $rep(E)$ the same as $phi_psi(psi)$?
....:
+----+---+---+---+
| -1 | 0 | 1 | 2 |
+----+---+---+---+
| <  | + | > |   |
+----+---+---+---+
|    | < | + | > |
+----+---+---+---+
|    |   |   | + |
+----+---+---+---+
True
True
+----+---+---+---+
| -1 | 0 | 1 | 2 |
+----+---+---+---+
| <  | - | > |   |
+----+---+---+---+
|    | < | - | > |
+----+---+---+---+
|    |   |   | + |
+----+---+---+---+
True
True
+----+---+---+---+
| -1 | 0 | 1 | 2 |
+----+---+---+---+
| <  | - | > |   |
+----+---+---+---+
|    | < | + | > |
+----+---+---+---+
|    |   |   | - |
+----+---+---+---+
True
True
+----+---+---+---+
| -1 | 0 | 1 | 2 |
+----+---+---+---+
| <  | + | > |   |
+----+---+---+---+
|    | < | - | > |
+----+---+---+---+
|    |   |   | - |
+----+---+---+---+
```

```
True
True
+----+---+---+---+
| -1 | 0 | 1 | 2 |
+----+---+---+---+
| <  | - | > |   |
+----+---+---+---+
|    | - | + | - |
+----+---+---+---+
|    |   |   | - |
+----+---+---+---+
False
False
```

3.9. $\psi^{max}(\pi)$ **and** $\psi^{min}(\pi)$**.** In this subsection, the following commands are explained:

- `E_max(E)`: the absolutely maximal member in $\Psi(\mathcal{E})$;
- `E_min(E)`: the absolutely minimal member in $\Psi(\mathcal{E})$.

In [HLL23a, §10], we defined distinguished elements of $\Psi(\mathcal{E})$, $\mathcal{E}^{|max|}$ and $\mathcal{E}^{|min|}$, which are the unique maximal and minimal elements in $\Psi(\mathcal{E})$ with respect to many orderings (see [HLL23a, Theorem 1.20]; note that orderings on $\Psi(\pi)$ are equivalent to orderings on $\Psi(\mathcal{E})$). $\mathcal{E}^{|max|}$ and $\mathcal{E}^{|min|}$ are computed by the commands `E_max(E)` and `E_min(E)` respectively as follows. The following code is complementary to [HLL23a, Example 10.14(1)].

```
sage: E=(([3 , -3], 3, -1),([1,0],0,1))
sage: symbol(E)
+----+----+----+---+---+---+---+
| -3 | -2 | -1 | 0 | 1 | 2 | 3 |
+----+----+----+---+---+---+---+
| <  | <  | <  | - | > | > | > |
+----+----+----+---+---+---+---+
|    |    |    | + | - |   |   |
+----+----+----+---+---+---+---+
sage: E_max(E)
(([3, -3], 3, -1), ([0, 0], 0, 1), ([1, 1], 0, -1))
sage: symbol(E_max(E))
+----+----+----+---+---+---+---+
| -3 | -2 | -1 | 0 | 1 | 2 | 3 |
+----+----+----+---+---+---+---+
| <  | <  | <  | - | > | > | > |
+----+----+----+---+---+---+---+
|    |    |    | + |   |   |   |
+----+----+----+---+---+---+---+
|    |    |    |   | - |   |   |
+----+----+----+---+---+---+---+
sage: E_min(E)
```

```
(([3, -3], 3, -1), ([1, -1], 1, 1), ([0, 0], 0, -1))
sage: symbol(E_min(E))
+----+----+----+---+---+---+---+
| -3 | -2 | -1 | 0 | 1 | 2 | 3 |
+----+----+----+---+---+---+---+
| <  | <  | <  | - | > | > | > |
+----+----+----+---+---+---+---+
|    |    | <  | + | > |   |   |
+----+----+----+---+---+---+---+
|    |    |    | - |   |   |   |
+----+----+----+---+---+---+---+
sage: E_can(E) # Notice that E_can, E_max, and E_min are
   all different.
(([3, -3], 3, -1), ([1, 0], 0, 1))
sage: symbol(E_can(E))
+----+----+----+---+---+---+---+
| -3 | -2 | -1 | 0 | 1 | 2 | 3 |
+----+----+----+---+---+---+---+
| <  | <  | <  | - | > | > | > |
+----+----+----+---+---+---+---+
|    |    |    | + | - |   |   |
+----+----+----+---+---+---+---+
sage: len(Packets_E(E)) # E_can(E)=E, E_max(E), E_min(E),
   are the only 3 symbols giving the same representation
3
```

If $\pi = \pi(\mathcal{E})$, for an extended multi-segment $\mathcal{E}$, then $\psi^{max}(\pi)$ and $\psi^{min}(\pi)$ are the local Arthur parameters attached to $\mathcal{E}^{|max|}$ and $\mathcal{E}^{|min|}$, respectively. These can be computed by the commands `psi_E(E_max(E))` and `psi_E(E_min(E))`, respectively.

```
sage: E=(([3 , -3], 3, -1),([1,0],0,1))
sage: rep(E) # The representation associated to E
(([-3, -3],), ([0, -1], [1, 1], [2, -1]))
sage: psi_E(E_max(E))
((1, 1), (1, 7), (3, 1))
sage: psi_E(E_min(E))
((1, 1), (1, 3), (1, 7))
```

We note that $\psi^{max}(\pi)$ is called the local Arthur parameter for π ([HLL23a, §10]).

References

[Art13] J. Arthur, *The endoscopic classification of representations: Orthogonal and Symplectic groups.* Colloquium Publication Vol. **61**, 2013, American Mathematical Society. 1

[Ato20a] H. Atobe, On an algorithm to compute derivatives. arXiv:2006.02638. 6

[Ato22a] H. Atobe, Construction of local A-packets. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, to appear, arXiv:2012.07232. 1, 7, 8, 10, 12

[Ato22b] H. Atobe, The set of local A-packets containing a given representation. arXiv:2201.00952. 1

[Aub95] A. Aubert, Dualité dans le groupe de Grothendieck de la catégorie des représentations lisses de longueur finie d'un groupe réductif p-adique. *Trans. Amer. Math. Soc.* **347**, 2179-2189 (1995). 12

[HLL23a] A. Hazeltine, B. Liu, C. Lo, On the intersection of local Arthur packets for classical groups and applications. (2023), Preprint. 1, 12, 13, 14, 15, 17, 18, 20, 21

[HLLZ22] A. Hazeltine, B. Liu, C. Lo, and Q. Zhang, The closure ordering conjecture on local L-parameters in local Arthur packets of classical groups. (2022), Preprint. arXiv:2209.03816. 17

[Mœ06a] C. Mœglin, Paquets d'Arthur pour les groupes classiques; point de vue combinatoire. arXiv:math/0610189v1. 1

[Mœ06b] C. Mœglin, Sur certains paquets d'Arthur et involution d'Aubert-Schneider-Stuhler généralisée. Represent. Theory **10**, (2006), 86-129. 1

[Moe09a] C. Mœglin, Paquets d'Arthur discrets pour un groupe classique p-adique. *Automorphic forms and L-functions II. Local aspects*, 179-257, Contemp. Math., **489**, Israel Math. Conf. Proc., *Amer. Math. Soc., Providence, RI*, 2009. 1

[Mœ10] C. Mœglin, Holomorphie des opérateurs d'entrelacement normalisés à l'aide des paramètres d'Arthur. Canad. J. Math. **62** (2010), no. 6, 1340-1386. 1

[Mœ11a] C. Mœglin, Multiplicité 1 dans les paquets d'Arthur aux places p-adiques. On certain L-functions, 333-374, Clay Math. Proc., **13**, Amer. Math. Soc., Providence, RI, 2011. 1

[Mœ11b] C. Mœglin, Image des opérateurs d'entrelacements normalisés et pôles des séries d'Eisenstein. *Adv. Math.* **228** (2011), 1068-1134.

[Xu21b] B. Xu, A combinatorial solution to Mœglin's parametrization of Arthur packets for p-adic quasisplit Sp(N) and O(N). *J. Inst. Math. Jussieu.* **20**, 1091-1204 (2021). 10

Department of Mathematics, Purdue University, West Lafayette, IN, 47907, USA

Email address: ahazelti@purdue.edu

Department of Mathematics, Purdue University, West Lafayette, IN, 47907, USA

Email address: liu2053@purdue.edu

Department of Mathematics, Purdue University, West Lafayette, IN, 47907, USA

Email address: lo93@purdue.edu