Katarzyna Macioszek

Uniwersytet Warszawski

Young tableaux and cell decomposition of the loop groups

Praca semestralna nr 1

(semestr letni 2009/10)

Opiekun pracy: Andrzej Weber

Young tableaux and cell decomposition of the loop groups

Katarzyna Macioszek
km@mimuw.edu.pl

Faculty of Mathematics, Informatics and Mechanics
University of Warsaw

Abstract

We show that a space of algebraic loops on a unitary group has a nice cell structure, similar in construction to the cell structure of Grassmann manifolds. We provide a description of this decomposition using the notion of Young tableaux and we find a subgroup of loops on unitary group whose action preserves decomposition.

1 Preliminaries

Let $n \in \mathbb{N}$ be fixed throughout this paper. Consider a space $\Lambda U(n)$ of smooth loops on a unitary group $U(n)$, that is smooth maps from a circle $S^1 \subset \mathbb{C}$ to $U(n)$. This space is itself a Lie group (with multiplication $(fg)(z) = f(z)g(z)$). By $\Omega U(n)$ we will denote a subgroup of $\Lambda U(n)$ consisting of loops which preserve a base point. Another subgroups of our interest will be spaces of algebraic loops:

$$\Lambda_{alg} U(n) = \{f \in \Lambda U(n) | \ \exists m \ f(z) = \sum_{i=-m}^{m} A_i z^i, \ A_i \in M_{n\times n}(\mathbb{C})\}$$

$$\Omega_{alg} U(n) = \{f \in \Lambda_{alg} U(n) | \ f(1) = 1\}$$

$\Omega_{alg} U(n)$ is known to be homotopy equivalent to $\Omega U(n)$ (see Preposition 8.6.6. in [PressSeg]).

We will also need a notation for integer sequences. Let $\underline{k}$ denote a sequence of n integers: $\underline{k} = (k_1, \ldots, k_n)$.

In section (2) we will show the cell structure on the spaces $\Omega_{alg} U(n)$ and $\Omega_{alg} SU(n)$, where $SU(n)$ denotes a special unitary group) In section (3) we will introduce a nice description of this decompositions using integer sequences and Young tableaux.

2 Cell decomposition

We will describe a cell decomposition of $\Omega_{alg}U(n)$ from the paper of Pressley [Press80]. Some proofs will be omitted.

Let H be a Hibert space of maps from unit circle in $\mathbb{C}$ to $\mathbb{C}^n$: $H = L^2(S^1, \mathbb{C}^n)$ and let $(e_1, ..., e_n)$ be the standard basis of $\mathbb{C}^n$. Let H_i and $H_{p,i}$ denote the subspaces of H:

$$H_p = span\{z^i e_j |\ 1 \leq j \leq n, i \geq p\},$$

$$H_{p,i} = span(H_{p+1} \cup \{z^p e_j |\ 1 \leq j \leq i\}).$$

Next, let X_m denote a set of subspaces $V \subseteq H$ such that:

— $zV \subseteq V$,

— $H_m \subseteq V \subseteq H_{-m}$.

Of course we have $X_0 \subseteq X_1 \subseteq \cdots \subseteq \bigcup_{m \geq 0} X_m$. We will denote the direct limit of X_m by X. If V is an element of X, then $V \ominus zV$ denotes orthogonal complementation of zV in V.

From this construction one can see that X_m is in fact a subspace of a Grassmannian $G(\mathbb{C}^{2mn})$. The space $G_m(H) = \{V \subseteq H |\ H_m \subseteq V \subseteq H_{-m}\}$ can be identified with $G(\mathbb{C}^{2mn})$, because if $H_m \subseteq V \subseteq H_{-m}$, then $V/H_m \subseteq H_{-m}/H_m \cong \mathbb{C}^{2mn}$.

In algebraic loops $\Lambda_{alg}U(n)$ we have subspaces $\Lambda^m_{alg}U(n)$: f is an element of $\Lambda^m_{alg}U(n)$ if it is of the form

$$f(z) = \sum_{i=-m}^{m} A_i z^i.$$

Clearly, $\Lambda_{alg}U(n) = \bigcup_m \Lambda^m_{alg}U(n)$. In the same way $\Omega^m_{alg}U(n)$ is a set of loops $f \in \Lambda^m_{alg}U(n)$ such that $f(1) = 1$.

Group $\Lambda_{alg}U(n)$ acts on H by unitary transformations: if $f \in \Lambda_{alg}U(n)$ and $v \in H$, then $(fv)(z) = f(z)v(z)$.

It is easy to check that it also acts on X: Let $f \in \Lambda^m_{alg}U(n)$ (i.e. $f(z) = \sum_{i=-m}^{m} A_i z^i$) and $V \in X_p$ (i.e. $zV \subseteq V$ and $H_p \subseteq V \subseteq H_{-p}$). Then $fV \subseteq H_{-p-m}$. On the other hand, $f(z)^{-1} = \sum_{i=-m}^{m} A_i^* z^{-i}$, so we have also that $f^{-1}H_{p+m} \subseteq H_p \subseteq V$, thus $H_{p+m} \subseteq fV$. Furthermore $zfV = f(zV) \subseteq fV$. Therefore $fV \in X_{p+m} \subseteq X$.

Proposition 2.1. *For every $w \in S^1$ i $V \in X$ an evaluation $ev_w \colon V \ominus zV \to \mathbb{C}^n$ is isometric and isomorphism.*

Proposition 2.2. $\Lambda_{alg}U(n)$ *acts transitively on X and the stabilizer group of the subspace H_0 is subgroup containing constant loops $U(n)$.*

For proofs see [Press80].

Conclusion 2.3. $X \cong \Lambda_{alg}U(n)/U(n) \cong \Omega_{alg}U(n)$.

Let us note also that if $f \in \Lambda^m_{alg}U(n)$, then $fH_0 \in X_m$. Therefore we can identify $\Omega^m_{alg}U(n)$ with X_m.

The action of $\Lambda_{alg}U(n)$ on X extends to the action of the group of all algebraic loops $S^1 \to GL(n,\mathbb{C})$, whose inverses are also algebraic loops. In other words, this group is $GL(n,\mathbb{C}[z,z^{-1}])$. We will denote it by $M_{\mathbb{C}}$. The stabilizer of H_0 in action of $M_{\mathbb{C}}$ on X is a subgroup $P \subseteq M_{\mathbb{C}}$ containing the loops

$$\left\{ \sum_{i=0}^{m} A_i z^i \in M_{\mathbb{C}} |\ A_0 \in GL(n,\mathbb{C}) \right\}.$$

Equivalently, $P = GL(n,\mathbb{C}[z])$.

Conclusion 2.4. $\Lambda_{alg}U(n)/U(n) \cong M_{\mathbb{C}}/P$.

The more convenient objects to work with will be formal series, i.e. ones that can have infinitely many terms with positive exponents. Let $\overline{M}_{\mathbb{C}}$ denote $\mathbb{C}[z^{-1},z]]$, where

$$f \in \mathbb{C}[z^{-1},z]] \Leftrightarrow f(z) = \sum_{i=-m}^{\infty} a_i z^i.$$

and let $\overline{P}$ denote $\mathbb{C}[[z]]$ (i.e. series $\sum_{i=0}^{\infty} a_i z^i$).

We will show, that $\overline{M}_{\mathbb{C}}/\overline{P} = M_{\mathbb{C}}/P$.

Consider a coset $g\overline{P} \in \overline{M}_{\mathbb{C}}/\overline{P}$ ($g \in \overline{M}_{\mathbb{C}}$). The following algorithm finds a representative g' of coset $g\overline{P}$ such that $g' \in M_{\mathbb{C}}$. We give this algorithm in details, because it is essential to understanding some of the further constructions.

1. We choose an element g_{ij} of matrix g, which begins with the lowest exponent. If there are more than one such elements, then we choose the one in the highest row. We switch the column with the chosen element with the first one.

2. Suppose the chosen element equals $\sum_{i \geq k_1} a_i z^i$, where $a_{k_1} \neq 0$. It can be written in the form $z^{k_1}h(z)$, where $h \in \overline{P}$ and $h(0) = a_{k_1}$. The series h is invertible in $\mathbb{C}[[z]]$, as $h(0) \neq 0$. Thus we can multiply the first column of g by h^{-1}. After doing this, we have in first column in one row an element z^{k_1}, in rows above there are series with exponents higher than k_1 and in the rows below there are series with exponents higher or equal k_1.

3. In the row with the chosen element on the right there are series with exponents higher or equal k_1. Thus by elementary column operations we can change all those series to 0.

Subsequently, we repeat steps 1-3, i.e. we choose the element with the lowest exponent and in the highest row, we switch its column with the second column, and then by elementary column operations we change this element to monomial z^{k_2} and the series on the right to zeros.

All described operations correspond to multiplication on the right by some element of $\overline{P}$, so we always get a matrix from $g\overline{P}$.

After n repetitions of steps 1-3 we get a matrix in the following form: in the i-th column in some row there is a monomial z^{k_i}. Above this monomial there are series with exponents higher than k_i and below there are series with exponents higher or equal k_i. On the row with monomial z_{k_i}, on the right side of it, there are zeros. Furthermore $k_1 \geq k_2 \geq \ldots k_n$.

Now with successive column operations we can make all series on the left from z^{k_i} to have exponents lower than k_i. Clearly, it is again a multiplication by some element of $\overline{P}$.

Eventually we get a matrix $g' \in g\overline{P}$ which has no series with infinitely many terms, i.e. $g' \in M_{\mathbb{C}}$. Note that after multiplying g' by any element of $\overline{P}$ other than identity we get a matrix that is not in the form described above.

Conclusion 2.5. *Given a coset $g\overline{P} \in \overline{M}_{\mathbb{C}}/\overline{P}$ there exists exactly one element $g' \in M_{\mathbb{C}}$ such that $g' \in g\overline{P}$ and there exist integers $k_1, \ldots, k_n \in \mathbb{Z}$ and permutation $m_1, \ldots, m_n$ of $\{1, \ldots, n\}$ such that:*

- *the element of g' in m_i-th row and i-th column equals z^{k_i} ($g'_{m_i,i} = z^{k_i}$),*
- *in the i-th column above that element there are polynomials with exponents higher than k_i, and below there are polynomials with exponents higher or equal k_i,*
- *in the m_i-th row, on the right of i-th element (i.e. z^{k_1}) there are zeros, and on the left there are polynomials with exponents lower than k_i,*
- *exponents k_i form a nondecreasing sequence,*
- *if $k_i = k_{i+1}$, then z^{k_i} in i-th column is in higher row than z^{k_i} in $(i+1)$-th column.*

Matrix of this form will be referred to as reduced matrix.

Example 2.6. *An example of reduced matrix:*

$$\begin{pmatrix} z^0 & 0 & 0 & 0 \\ \bullet + \bullet z & \bullet z & z^2 & 0 \\ \bullet + \bullet z + \bullet z^2 & \bullet z + \bullet z^2 & \bullet z^2 & z^3 \\ 0 & z^0 & 0 & 0 \end{pmatrix}$$

(A dot $\bullet$ stands for any complex number). In this case we have $k_1 = 0$, $k_2 = 0$, $k_3 = 2$, $k_4 = 3$, $m_1 = 1$, $m_2 = 4$, $m_3 = 2$, $m_4 = 3$.

We have shown that the map $M_{\mathbb{C}}/P \to \overline{M}_{\mathbb{C}}/\overline{P}$ given by $gP \mapsto g\overline{P}$ is „onto".

We will now show that it is also injective. Lets assume that $g, h \in M_{\mathbb{C}}$ and $g\overline{P} = h\overline{P}$. Then we have $h^{-1}g \in \overline{P}$, and also $h^{-1}g \in P$. Therefore $gP = hP$. This finishes the proof of $\overline{M}_{\mathbb{C}}/\overline{P} = M_{\mathbb{C}}/P$.

Let B denote a subgroup of P consisting of matrices $\sum_{i=0}^{m} A_i z^i$ such that A_0 is lower triangular. Group B acts on the left on $M_{\mathbb{C}}/P$. Let $\mu_{\underline{k}}$ denote a reduced matrix corresponding to the sequence $\underline{k}$ and with all terms equal 0 except one z^{k_i} in i-th row. Multiplying $\mu_{\underline{k}}$ on the left by an element of B gives us an element of coset whose reduced matrix has also monomial z^{k_i} on the i-th row. What is more, we get in this way every reduced matrix of this form.

We denote the orbit of action of B on $\mu_{\underline{k}}P$ by $Q_{\underline{k}}$:

$$Q_{\underline{k}} = B\mu_{\underline{k}}P/P.$$

The above construction gives us a cell decomposition of X.

Proposition 2.7. *Dimension of the cell $Q_{\underline{k}}$ equals*

$$\sum_{i<j} |k_i - k_j| - v(\underline{k}),$$

where $v(\underline{k})$ is the number of inversions in sequence $\underline{k}$, i.e. the number of pairs i, j such that $i < j$ i $k_i > k_j$.

Our final preposition will be:

Proposition 2.8. *The loop space $\Omega_{alg}U(n)$ consists of $\mathbb{Z}$ connected components. The i-th connected component consists of loops f with degree of determinant $\det f$ equal to i. What is more, if $f \in Q_{\underline{k}}$, then degree of determinant $\det f$ equals $\sum_i k_i$. Therefore*

$$\Omega_{alg}U(n) = \coprod_p \left(\bigcup_{\sum_i k_i = p} Q_{\underline{k}} \right).$$

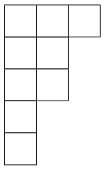
It follows immediately that:

Conclusion 2.9. $\Omega_{alg}SU(n) = \bigcup_{\sum_i k_i = 0} Q_{\underline{k}}$.

3 Cells of $\Omega_{alg}U(n)$ as Young tableaux

We have described decomposition of $\Omega_{alg}U(n)$ into cells $Q_{\underline{k}}$ indexed with sequences of integers $\underline{k}$. We will now show a geometric interpretation of this decomposition.

First let us take a reduced matrix from the previous section and for every column let us write every dot from this column as a box of Young tableau. For the cell $Q_{(0,2,3,0)}$ from example (2.6) we get:



This construction will become more natural (especially if one is familiar with the cell decomposition of Grassmann manifolds) if we see it in the following way.

Consider a subspace $V \in X$. We know, that $zV \subseteq V$ and that $V \ominus zV$ is isomorphic with $\mathbb{C}^n$ (see Preposition 2.1). Then there exist n linearly independent vectors $(v_1, v_2, \ldots, v_n)$ such that $V = span\{z^i v_j |\ i \geq 0\}$. We can write the vectors v_j in infinite matrix:

z^3		z^2				z^1				z^0			
$\ldots$	e_1	e_4	e_3	e_2	e_1	e_4	e_3	e_2	e_1	e_4	e_3	e_2	e_1
$\ldots$	$v_{1,11}$	$v_{1,10}$	$v_{1,9}$	$v_{1,8}$	$v_{1,7}$	$v_{1,6}$	$v_{1,5}$	$v_{1,4}$	$v_{1,3}$	$v_{1,2}$	$v_{1,1}$		
$\ldots$	$v_{2,13}$	$v_{2,12}$	$v_{2,11}$	$v_{2,10}$	$v_{2,9}$	$v_{2,8}$	$v_{2,7}$	$v_{2,6}$	$v_{2,5}$	$v_{2,4}$	$v_{2,3}$	$v_{2,2}$	$v_{2,1}$
$\ldots$	$v_{3,3}$	$v_{3,2}$	$v_{3,1}$										
$\ldots$	$v_{4,7}$	$v_{4,6}$	$v_{4,5}$	$v_{4,4}$	$v_{4,3}$	$v_{4,2}$	$v_{4,1}$						

By elementary row operations, analogous to ones used in cell decomposition of Grassmannians, we can make this matrix into some kind of echelon form, But in our case we have additional condition that $zV \subseteq V$, so there will be additional zeros in every column shifted to the left from the rightmost ones by multiple of n.

We finally get a matrix of this form:

	z^3				z^2				z^1				z^0			
$\ldots$	e_4	e_3	e_2	e_1	e_4	e_3	e_2	e_1	e_4	e_3	e_2	e_1	e_4	e_3	e_2	e_1
$\ldots$	0	0	0	0	0	$\bullet$	0	0	0	$\bullet$	$\bullet$	0	0	$\bullet$	$\bullet$	1
$\ldots$	0	0	0	0	0	$\bullet$	0	0	0	$\bullet$	$\bullet$	0	1			
$\ldots$	0	0	0	0	0	$\bullet$	1									
$\ldots$	0	1														

In rows there are vectors $(v'_1, v'_2, \ldots, v'_n)$ such that $V = span\{z^i v'_j |\ i \geq 0\}$. Number of dots in the matrix equals the complex dimension of $Q_{\underline{k}}$.

The matrix in our example corresponds to the cell $Q_{(0,2,3,0)}$.

We can think of this infinite matrix as a flatten reduced matrix from the algorithm described earlier. Then it is easy to see that this operations corresponds to successive steps of the algorithm and that dots form the same Young tableau (only transposed).

Let us denote by $Y(\underline{k})$ the Young tableau corresponding to the cell $Q_{\underline{k}}$.

We will now describe some properties of cells of $\Omega_{alg}U(n)$ and corresponding Young tableaux. Some proofs will be omitted.

Proposition 3.1. *Consider the nondecreasing sequence $\underline{k}$ and some non-constant permutation $\underline{k'}$ of $\underline{k}$. Then cell $Q_{\underline{k}}$ is of higher dimension than $Q_{\underline{k'}}$ and $Q_{\underline{k'}}$ lies in the closure of $Q_{\underline{k}}$.*

Proposition 3.2. *Consider the sequence $\underline{k}$ and an inversion s_i which switches k_i and k_{i+1}. Let us assume that k_i and k_{i+1} are respectively the m_i-th and m_{i+1}-th terms of corresponding sorted sequence. Then:*

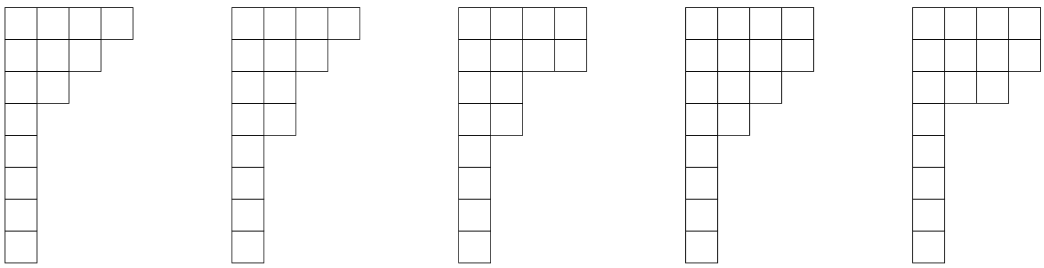
- *if $k_i < k_{i+1}$, then tableau $Y(s_i\underline{k})$ is $Y(\underline{k})$ without one box in m_i-th column,*
- *if $k_i > k_{i+1}$, then tableau $Y(s_i\underline{k})$ is $Y(\underline{k})$ with one additional box in m_{i+1}-th column,*

In other words, inversion s_i adds or removes one box from the column corresponding to the lesser of k_i and k_{i+1}. The box is added iff inversion decreases the total number of inversions in sequence $\underline{k}$.

Example 3.3. *Let us take the sequence $\underline{k} = (5, 2, 3, 2, 0)$ and apply to it inversions s_1, s_2, s_3, s_1. We get:*

$$(5,2,3,2,0) \xrightarrow{s_1} (2,5,3,2,0) \xrightarrow{s_2} (2,3,5,2,0) \xrightarrow{s_3} (2,3,2,5,0) \xrightarrow{s_1} (3,2,2,5,0)$$

$$(8,3,2,1) \qquad (8,4,2,1) \qquad (8,4,2,2) \qquad (8,4,3,2) \qquad (8,3,3,2)$$



In section (4) there is a Python program that takes a sequence $\underline{k}$ and a sequence of inversions and prints successive Young tableaux.

Proposition 3.4. *Consider a Young tableau of a given shape. Then in every connected component of $\Omega_{alg}U(n)$ there exists exactly one cell corresponding to this tableau. Equivalently, for every $p \in \mathbb{Z}$ there exists exactly one cell $Q_{\underline{k}}$ corresponding to this tableau and for which $\sum_i k_i = p$.*

Proof. From the process of creating the Young tableau from a sequence described in this section it is easy to see that if we fix the place of the rightmost one in the matrix, then there is a deterministic way to put all the other ones, zeros and dots into matrix (as a illustration, there is a Python program in section (4) that takes a shape of a Young tableau and compute a corresponding sequence $\underline{k}$ such that $k_1 = 0$). What is more, there are n ones and they lie in columns $e_1, \ldots, e_n$, so there is a one in column e_1 and a one in column e_n. Therefore moving all ones, zeros and dots one column left (or right) gives us a correct matrix filling with the same Young tableau but with a sum of exponents increased (decreased) by one, because it moves exactly one one from the column $z^k e_n$ to $z^{k+1} e_1$ (or from $z^k e_1$ to $z^{k-1} e_n$). Thus this shifted Young tableau corresponds to the cell in other connected component of $\Omega_{alg} U(n)$. □

Proposition 3.5. *The only elements of $\Omega_{alg} U(n)$ whose actions preserve the cell structure on $\Omega_{alg} U(n)$ are*

$$\left(\begin{array}{c|c} 0 & z^{k+1} I_m \\ \hline z^k I_{n-m} & 0 \end{array}\right)$$

Proof. In the proof of (3.4) we have seen that shifting a Young tableau in the infinite matrix to the left (or right) changes $z^k e_n$ to $z^{k+1} e_1$ (or $z^k e_1$ to $z^{k-1} e_n$), so it changes a sequence $(k_1, k_2, \ldots, k_{n-1}, k_n)$ to $(k_n + 1, k_1, k_2, \ldots, k_{n-1})$ (or to $(k_2, \ldots, k_{n-1}, k_n, k_1 - 1)$). We will denote this operation by s_0. As a element of $\Omega_{alg} U(n)$ s_0 is:

$$\left(\begin{array}{c|c} 0 & z \\ \hline I_{n-1} & 0 \end{array}\right)$$

and z_0^{-1} is:

$$\left(\begin{array}{c|c} 0 & I_{n-1} \\ \hline z^{-1} & \end{array}\right)$$

Composing s_0 gives matrix

$$\left(\begin{array}{c|c} 0 & z^{k+1} I \\ \hline z^k I & 0 \end{array}\right)$$

Consider some other operation preserving cell structure which is not of this form. It must then correspond to some operation on n-sequences from group generated by inversions $s_1,\ldots,s_n$ and by s_0. In particular this operation must be a composition of some number of s_0 and permutations (in any order) and this permutations must preserve cell structure. But this is not possible for any non-constant permutation as there always exists a sequence for which given permutation changes the number of inversions (and therefore the dimension of corresponding cell). □

4 Programs

In this section we give two Python programs: youngToSeq.py which changes Young tableau into a sequence and seqToYoung.py which changes sequence into Young tableau.

```
#!/usr/bin/python
# file young.py
# a class Young used in youngToSeq.py and seqToYoung.py

import sys

class Young:
  def __init__ (self, line):
    maxLine = max(line) + 1
    minLine = min(line)
    self.setRange(minLine,maxLine)
    self.setLine(line)

  def setLine(self, line):
    self.line = line
    dots = len(self.line)
    self.young = []
    for r in self.ranges:
      for l in reversed (self.line):
        if r < l:
          self.young.append(dots)
        elif r == l:
          dots -=1
    end = len(self.young)
    for y in reversed (self.young):
      if y == 0:
        end -=1
      else:
        if end < len(self.young):
          self.young = self.young [:end]

  def setRange(self, minimum, maximum):
    self.min = minimum
    self.max = maximum
    self.ranges = range(self.min,self.max)
    self.ranges.reverse()

  def restoreRange(self):
    minimum = min(self.min, min(self.young))
    maximum = max (self.max, max (self.young))
    if minimum < self.min or maximum > self.max:
      self.setRange(minimum, maximum)
      self.setLine(self.line)
```

```
# file young.py

  def printYourself(self):
    print self.line, "sum =",sum(self.young)
    i = 1
    print " ",
    while True:
      s = 0
      for y in self.young:
        if y >= i:
          s += 1
      if s == 0:
        break
      print s,
      i += 1
    print
    for y in self.young:
      print y,
      for i in range(0,y):
        print "*",
      print
    print

  def transform(self, n):
    copy = self.line[:]
    if n > 0:
      copy[n-1] = self.line[n]
      copy[n] = self.line[n-1]
    else:
      copy[n-1] = self.line[n] + 1
      copy[n] = self.line[n-1] - 1
    self.setLine(copy)
    self.restoreRange()
```

```
#!/usr/bin/python
# file youngToSeq.py
# Program takes a shape of a Young tableau
# and gives a corresponding sequence starting with 0
# usage: ./youngToSeq <n> <shape-of-tableau>
# example:
# ./youngToSeq.py 4 1,1,2,2,3
# result: [0, 2, 3, 0]

import sys

def youngToSeq(young, N):
  unusedIndices = range(0,N)
  sequence = list(-1 for i in range(0, N))
  sequence [0] = 0
  unusedIndices.remove(0)
  currentWidth = 1
  currentIndex = 0
  currentColumn = 0
  currentValue = 0
  while currentColumn < len(young):
    print
    while(currentColumn < len(young)) \
      and (young [currentColumn] == currentWidth)):
      currentColumn += 1
      currentIndex = \
        (currentIndex + 1) % len(unusedIndices)
      if currentIndex == 0:
        currentValue += 1
    if currentColumn == len(young):
      for i in unusedIndices:
        sequence [i] = currentValue
      break
    sequence [unusedIndices [currentIndex] ] = currentValue
    del unusedIndices [currentIndex]
    if currentIndex == len(unusedIndices):
      currentValue += 1
    currentIndex = currentIndex % len(unusedIndices)
    currentWidth += 1
  return sequence

n = int(sys.argv [1])
young = map(lambda x: int(x), sys.argv[2].split(’,’))

sequence = youngToSeq(young, n)

print sequence
```

```
#!/usr/bin/python
# file seqToYoung.py
# Program takes a sequence corresponding to some cell
# and a sequence of inversions
# and gives a Young tableaux of all obtained cells
# usage: ./seqToYoung <sequence> <inversions>
# example:
# ./seqToYoung 5,2,3,2,0  1,2,3,1

line = map(lambda x: int(x), sys.argv[1].split(','))
transforms = map(lambda x: int(x), sys.argv[2].split(','))

y = young.Young (line)
y.printYourself()
for t in transforms:
  print t
  y.transform(t)
  y.printYourself()
  print
```

References

[Press80] Andrew Pressley, *Decomposition of the space of loops on a Lie group*, Topology Vol. 19, Pergamon Press Ltd., 1980, s. 65–79

[PressSeg] Andrew Pressley, Graeme Segal, *Loop groups*, Oxford University Press, New York 1988