

Package: syt (via r-universe)

August 23, 2024

Type Package

Title Young Tableaux

Version 0.5.0

Date 2024-07-22

Author Stéphane Laurent

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Description Deals with Young tableaux (field of combinatorics). For standard Young tableaux, performs enumeration, counting, random generation, the Robinson-Schensted correspondence, and conversion to and from paths on the Young lattice. Also performs enumeration and counting of semistandard Young tableaux, enumeration of skew semistandard Young tableaux, enumeration of Gelfand-Tsetlin patterns, and computation of Kostka numbers.

License GPL-3

URL <https://github.com/stla/syt>

BugReports <https://github.com/stla/syt/issues>

Imports Matrix, partitions, utils

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Repository CRAN

Date/Publication 2024-07-23 20:10:01 UTC

Contents

all_ssSkewTableaux	2
all_ssytx	3
all_sytx	4

ballot2syt	4
count_ssytX	5
count_sytX	6
dualSkewTableau	6
dualsyt	7
dualTableau	7
firstsyt	8
GelfandTsetlinPatterns	8
gprocess2syt	9
hooklengths	10
hooks	10
isSemistandardSkewTableau	11
isSkewTableau	11
isSSYT	12
isStandardSkewTableau	12
isSYT	13
KostkaNumber	13
KostkaNumbers	14
KostkaNumbersWithGivenLambda	15
KostkaNumbersWithGivenMu	15
matrix2syt	16
nextsyt	17
prettyGT	17
prettySkewTableau	18
prettyTableau	18
rgprocess	19
RS	19
rsyt	20
skewGelfandTsetlinPatterns	20
skewKostkaNumbers	21
skewTableauxWithGivenShapeAndWeight	22
ssytX_withGivenShapeAndWeight	22
syt2ballot	23
syt2gprocess	24
syt2matrix	24
tableau2matrix	25
tableauShape	26
Index	27

all_ssSkewTableaux	<i>Semistandard skew tableaux</i>
--------------------	-----------------------------------

Description

Enumeration of all semistandard skew tableaux with given shape and given maximum entry.

Usage

```
all_ssSkewTableaux(lambda, mu, n)
```

Arguments

lambda, mu integer partitions defining the skew partition: lambda is the outer partition and mu is the inner partition (so mu must be a subpartition of lambda)

n a positive integer, the maximum entry of the skew tableaux

Value

The list of all semistandard skew tableaux whose shape is the skew partition defined by lambda and mu and with maximum entry n.

See Also

[all_ssytx](#), [skewTableauxWithGivenShapeAndWeight](#).

Examples

```
ssstx <- all_ssSkewTableaux(c(4, 3, 1), c(2, 2), 2)
lapply(ssstx, prettySkewTableau)
```

all_ssytx

Enumeration of semistandard Young tableaux

Description

Generates all semistandard Young tableaux of a given shape and filled with integers between 1 and a given n.

Usage

```
all_ssytx(lambda, n)
```

Arguments

lambda an integer partition, the shape

n an integer, the maximum value of the entries (the minimum value is 1)

Value

List of all semistandard Young tableaux with shape lambda and filled with integers between 1 and n.

See Also

[ssytx_withGivenShapeAndWeight](#).

Examples

```
ssytx <- all_ssytx(c(2, 1), 3)
lapply(ssytx, prettyTableau)
```

all_sytx	<i>Enumeration of standard Young tableaux</i>
----------	---

Description

Generates all standard Young tableaux of a given shape.

Usage

```
all_sytx(lambda)
```

Arguments

lambda the shape, an integer partition

Value

A list of standard Young tableaux.

Examples

```
syttx <- all_syttx(c(5, 2))
lapply(syttx, prettyTableau)
```

ballot2syt	<i>Tableau as ballot sequence</i>
------------	-----------------------------------

Description

Converts a ballot sequence to its corresponding standard Young tableau.

Usage

```
ballot2syt(a)
```

Arguments

a ballot sequence

Value

A standard Young tableau.

See Also[syt2ballot](#)**Examples**

```
a <- c(1,1,2,3,2,1)
ballot2syt(a)
```

count_ssytx	<i>Number of semistandard Young tableaux</i>
-------------	--

Description

Number of semistandard Young tableaux of a given shape and filled with integers between 1 and a given n.

Usage

```
count_ssytx(lambda, n)
```

Arguments

lambda	an integer partition, the shape
n	an integer, the maximum value of the entries (the minimum value is 1)

Value

The number of semistandard Young tableaux with shape lambda and filled with integers between 1 and n.

See Also[KostkaNumber](#).**Examples**

```
count_ssytx(c(4, 3, 3, 2), 5)
```

count_sytx	<i>Number of standard Young tableaux</i>
------------	--

Description

Number of standard Young tableaux of a given shape.

Usage

```
count_sytx(lambda)
```

Arguments

lambda an integer partition, the shape

Value

An integer, the number of standard Young tableaux of shape lambda.

See Also

[all_sytx](#).

Examples

```
count_sytx(c(5, 4, 1))
length(all_sytx(c(5, 4, 1)))
```

dualSkewTableau	<i>Dual skew tableau</i>
-----------------	--------------------------

Description

Returns the dual (skew) tableau of a skew tableau.

Usage

```
dualSkewTableau(skewTableau)
```

Arguments

skewTableau a skew tableau

Value

A skew tableau.

Examples

```
tbl <- list(c(NA, NA, 1, 1), c(NA, 1), c(1, 2))
dtbl <- dualSkewTableau(tbl)
prettySkewTableau(dtbl)
```

dualsyt

Dual tableau

Description

The dual standard Young tableau of a standard Young tableau.

Usage

```
dualsyt(syt)
```

Arguments

syt standard Young tableau

Value

A standard Young tableau.

Examples

```
syt <- list(c(1,2,6), c(3,5), 4)
dualsyt(syt)
```

dualTableau

Dual tableau

Description

The dual tableau of a tableau (mirror image to the main diagonal).

Usage

```
dualTableau(tableau)
```

Arguments

tableau a tableau

Value

A tableau.

Examples

```
tbl <- list(c("a", "s", "e", "f"), c("f", "o"), c("u"))
dualTableau(tbl)
```

firstsynt	<i>First tableau of a given shape</i>
-----------	---------------------------------------

Description

Returns the "first" standard Young tableau of a given shape.

Usage

```
firstsynt(lambda)
```

Arguments

lambda the shape, an integer partition

Value

A standard Young tableau.

Examples

```
firstsynt(c(4, 2, 1))
```

GelfandTsetlinPatterns	<i>Gelfand-Tsetlin patterns</i>
------------------------	---------------------------------

Description

Enumeration of Gelfand-Tsetlin patterns defined by a given integer partition and a given weight.

Usage

```
GelfandTsetlinPatterns(lambda, weight)
```

Arguments

lambda integer partition; up to trailing zeros, this will be the top diagonal of the generated Gelfand-Tsetlin patterns

weight integer vector; the partial sums of this vector will be the diagonal sums of the generated Gelfand-Tsetlin patterns

Value

A list of Gelfand-Tsetlin patterns. A Gelfand-Tsetlin pattern is a triangular array of non-negative integers, and it is represented by the list of its rows. Hence the first element of this list is an integer, the second element is an integer vector of length two, and so on. The length of this list is the length of weight.

See Also

[skewGelfandTsetlinPatterns](#).

Examples

```
GTpatterns <- GelfandTsetlinPatterns(c(3, 1), c(1, 1, 1, 1))
lapply(GTpatterns, prettyGT)
```

gprocess2syt

Growth process to tableau

Description

Converts a growth process of integer partitions to its corresponding standard Young tableau.

Usage

```
gprocess2syt(path)
```

Arguments

path a path of the Young graph from the root vertex, given as a list of integer partitions

Value

A standard Young tableau.

See Also

[syt2gprocess](#).

Examples

```
path <- list(1, 2, c(2,1), c(3,1), c(3,1,1))
gprocess2syt(path)
```

hooklengths

Hook lengths

Description

Hook lengths of a given integer partition.

Usage

```
hooklengths(lambda)
```

Arguments

lambda an integer partition

Value

The hook lengths of the partition, given in a list.

See Also

[hooks](#).

Examples

```
hooklengths(c(4, 2))
```

hooks

Hooks

Description

Hooks of a given integer partition.

Usage

```
hooks(lambda)
```

Arguments

lambda integer partition

Value

The hooks of the partition in a list.

See Also

[hooklengths](#).

Examples

```
hooks(c(4, 2))
```

```
isSemistandardSkewTableau
```

Check whether a skew tableau is semistandard

Description

Check whether a skew tableau is a semistandard skew tableau.

Usage

```
isSemistandardSkewTableau(skewTableau)
```

Arguments

skewTableau a skew tableau

Value

A Boolean value.

Examples

```
tbl <- list(c(NA, NA, 1, 1), c(NA, 1), c(1, 2))
isSemistandardSkewTableau(tbl)
```

```
isSkewTableau
```

Check whether a tableau is a skew tableau

Description

Check whether a tableau is a skew tableau.

Usage

```
isSkewTableau(tableau)
```

Arguments

tableau a tableau

Value

A Boolean value.

Examples

```
tbl <- list(c(NA, NA, 1, 1), c(NA, 1), c(1, 2))
isSkewTableau(tbl)
```

isSSYT

Checks whether a tableau is semistandard

Description

Checks whether a tableau is a semistandard Young tableau.

Usage

```
isSSYT(tableau)
```

Arguments

tableau a tableau

Value

A Boolean value.

Examples

```
tbl <- list(c(1, 2, 6), c(5, 5), 7)
isSSYT(tbl)
```

isStandardSkewTableau

Check whether a skew tableau is standard

Description

Check whether a skew tableau is a standard skew tableau.

Usage

```
isStandardSkewTableau(skewTableau)
```

Arguments

skewTableau a skew tableau

Value

A Boolean value.

Examples

```
tbl <- list(c(NA, NA, 1, 1), c(NA, 1), c(1, 2))
isStandardSkewTableau(tbl)
```

isSYT	<i>Checks whether a tableau is standard</i>
-------	---

Description

Checks whether a tableau is a standard Young tableau.

Usage

```
isSYT(tableau)
```

Arguments

tableau a tableau

Value

A Boolean value.

Examples

```
tbl <- list(c(1, 2, 6), c(3, 5), 4)
isSYT(tbl)
```

KostkaNumber	<i>Kostka number</i>
--------------	----------------------

Description

Computes a Kostka number.

Usage

```
KostkaNumber(lambda, mu)
```

Arguments

lambda an integer partition
mu an integer vector whose sum equals the weight (i.e. the sum) of lambda

Details

The Kostka number $K(\lambda, \mu)$ is the number of semistandard Young tableaux with shape λ and weight μ . It does not depend on the order of the elements of μ (so one can always take an integer partition for μ). The *weight* is the vector whose i -th element is the number of occurrences of i in the tableau.

Value

The Kostka number corresponding to lambda and mu.

See Also

[KostkaNumbers](#), [KostkaNumbersWithGivenMu](#), [KostkaNumbersWithGivenLambda](#), [skewKostkaNumbers](#).

Examples

```
KostkaNumber(c(3,2), c(1,1,1,2))
KostkaNumber(c(3,2), c(1,1,2,1))
KostkaNumber(c(3,2), c(1,2,1,1))
KostkaNumber(c(3,2), c(2,1,1,1))
lambda <- c(4, 3, 1)
mu <- rep(1, sum(lambda))
KostkaNumber(lambda, mu) == count_sytz(lambda) # should be TRUE
```

KostkaNumbers

Kostka numbers for all partitions of a given weight

Description

Computes the Kostka numbers for all integer partitions of a given weight.

Usage

```
KostkaNumbers(n)
```

Arguments

n positive integer, the weight of the partitions

Value

An integer matrix, whose row names and column names encode the partitions λ and μ and whose entries are the Kostka numbers $K(\lambda, \mu)$.

See Also

[KostkaNumbersWithGivenLambda](#), [KostkaNumbersWithGivenMu](#), [skewKostkaNumbers](#).

Examples

```
KostkaNumbers(4)
```

KostkaNumbersWithGivenLambda
Kostka numbers with given λ

Description

Lists all positive Kostka numbers $K(\lambda, \mu)$ with a given partition λ .

Usage

`KostkaNumbersWithGivenLambda(lambda, output = "vector")`

Arguments

<code>lambda</code>	integer partition
<code>output</code>	the format of the output, either "vector" or "list"

Value

If `output="vector"`, this function returns a named vector. This vector is made of the non-zero (i.e. positive) Kostka numbers $K(\lambda, \mu)$, which are integers, and its names encode the partitions μ . If `output="list"`, this function returns a list of lists. Each of these lists has two elements. The first one is named `mu` and is an integer partition, and the second one is named `value` and is a positive integer, the Kostka number $K(\lambda, \mu)$. It is faster to compute the Kostka numbers with this function than computing the individual Kostka numbers with the function [KostkaNumber](#).

See Also

[KostkaNumber](#), [KostkaNumbers](#), [KostkaNumbersWithGivenMu](#).

Examples

`KostkaNumbersWithGivenLambda(c(2, 1, 1))`

KostkaNumbersWithGivenMu
Kostka numbers with given μ

Description

Lists all positive Kostka numbers $K(\lambda, \mu)$ with a given partition μ .

Usage

`KostkaNumbersWithGivenMu(mu, output = "vector")`

Arguments

mu integer partition
 output the format of the output, either "vector" or "list"

Value

If output="vector", this function returns a named vector. This vector is made of the positive Kostka numbers $K(\lambda, \mu)$ and its names encode the partitions λ . If output="list", this function returns a list of lists. Each of these lists has two elements. The first one is named lambda and is an integer partition, and the second one is named value and is a positive integer, the Kostka number $K(\lambda, \mu)$. It is faster to compute the Kostka numbers with this function than computing the individual Kostka numbers with the function [KostkaNumber](#).

See Also

[KostkaNumber](#), [KostkaNumbers](#), [KostkaNumbersWithGivenLambda](#).

Examples

```
KostkaNumbersWithGivenMu(c(2, 1, 1))
```

```
matrix2syt
```

```
Standard Young tableau from a matrix
```

Description

Converts a matrix to a standard Young tableau.

Usage

```
matrix2syt(M)
```

Arguments

M a matrix

Value

A standard Young tableau.

See Also

[syt2matrix](#).

Examples

```
M <- rbind(c(1,2,6), c(3,5,0), c(4,0,0))
matrix2syt(M)
```

nextsyt	<i>Next tableau</i>
---------	---------------------

Description

Given a standard Young tableau, returns the "next" one having the same shape.

Usage

```
nextsyt(syt)
```

Arguments

syt a standard Young tableau

Value

A standard Young tableau of the same shape as syt, or NULL if syt is the last standard Young tableau of this shape.

Examples

```
syt <- firstsyt(c(4, 2, 1))
nextsyt(syt)
```

prettyGT	<i>Pretty Gelfand-Tsetlin pattern</i>
----------	---------------------------------------

Description

Pretty form of a Gelfand-Tsetlin pattern.

Usage

```
prettyGT(GT)
```

Arguments

GT a Gelfand-Tsetlin pattern

Value

A 'noquote' character matrix.

prettySkewTableau *Pretty skew tableau*

Description

Pretty form of a skew tableau.

Usage

```
prettySkewTableau(skewTableau)
```

Arguments

skewTableau a skew tableau

Value

A 'noquote' character matrix.

Examples

```
tbl <- list(c(NA, NA, 1, 1), c(NA, 1), c(1, 2))
prettySkewTableau(tbl)
```

prettyTableau *Pretty tableau*

Description

Pretty form of a tableau.

Usage

```
prettyTableau(tableau)
```

Arguments

tableau a tableau

Value

A 'noquote' character matrix.

Examples

```
tbl <- list(c(0, 2, 1, 1), c(4, 1), c(1, 2))
prettyTableau(tbl)
```

rgprocess	<i>Plancherel growth process</i>
-----------	----------------------------------

Description

Samples a path of the Young graph according to the Plancherel growth process.

Usage

```
rgprocess(n)
```

Arguments

`n` the size of the path to be sampled

Value

The path as a list, starting from the root vertex 1.

See Also

[gprocess2syt](#) and [syt2gprocess](#) to convert a Young path to a standard Young tableau and conversely.

Examples

```
rgprocess(7)
```

RS	<i>Robinson-Schensted correspondence</i>
----	--

Description

Pair of standard Young tableaux given from a permutation by the Robinson-Schensted correspondence.

Usage

```
RS(sigma)
```

Arguments

`sigma` a permutation given as a vector of integers

Value

A list of two standard Young tableaux.

Examples

```
RS(c(1, 3, 6, 4, 7, 5, 2))
```

```
rsyt
```

```
Random standard Young tableau
```

Description

Uniform sampling of a standard Young tableau of a given shape.

Usage

```
rsyt(lambda)
```

Arguments

lambda shape, an integer partition

Value

A standard Young tableau of shape lambda.

Examples

```
rsyt(c(7, 3, 1))
```

```
skewGelfandTsetlinPatterns
```

```
Skew Gelfand-Tsetlin patterns
```

Description

Enumeration of Gelfand-Tsetlin patterns defined by a given skew partition and a given weight.

Usage

```
skewGelfandTsetlinPatterns(lambda, mu, weight)
```

Arguments

lambda, mu integer partitions defining the skew partition: lambda is the outer partition and mu is the inner partition (so mu must be a subpartition of lambda); lambda will be the last row of the generated Gelfand-Tsetlin patterns and mu will be their first row

weight integer vector; this vector will be the differences of the row sums of the generated Gelfand-Tsetlin patterns; consequently, there will be no generated Gelfand-Tsetlin pattern unless the sum of weight equals the difference between the sum of lambda and the sum of mu

Value

A list of matrices with non-negative integer entries. The number of columns of these matrices is the length of lambda and the number of rows of these matrices is one plus the length of weight.

See Also

[GelfandTsetlinPatterns](#).

Examples

```
skewGelfandTsetlinPatterns(c(3, 1, 1), c(2), c(1, 1, 1))
```

skewKostkaNumbers	<i>Skew Kostka numbers</i>
-------------------	----------------------------

Description

Skew Kostka numbers associated to a given skew partition.

Usage

```
skewKostkaNumbers(lambda, mu, output = "vector")
```

Arguments

lambda, mu	integer partitions defining the skew partition: lambda is the outer partition and mu is the inner partition (so mu must be a subpartition of lambda)
output	the format of the output, either "vector" or "list"

Details

The skew Kostka number $K_{\lambda/\mu, \nu}$ is the number of skew semistandard Young tableaux with shape λ/μ and weight ν . The *weight* of a Young tableau is the vector whose i -th element is the number of occurrences of i in this tableau.

Value

If output="vector", the function returns a named vector. This vector is made of the positive skew Kostka numbers $K_{\lambda/\mu, \nu}$ and its names encode the partitions ν . If output="list", the function returns a list. Each element of this list is a named list with two elements: an integer partition ν in the field named "nu", and the corresponding skew Kostka number $K_{\lambda/\mu, \nu}$ in the field named "value". Only the non-null skew Kostka numbers are provided by this list.

See Also

[KostkaNumber](#), [KostkaNumbersWithGivenMu](#).

Examples

```
skewKostkaNumbers(c(4,2,2), c(2,2))
```

```
skewTableauxWithGivenShapeAndWeight
```

Skew semistandard tableaux with given shape and weight

Description

Enumeration of all skew semistandard tableaux with a given shape and a given weight. The *weight* of a tableau is the vector whose i -th element is the number of occurrences of i in this tableau.

Usage

```
skewTableauxWithGivenShapeAndWeight(lambda, mu, weight)
```

Arguments

lambda, mu	integer partitions defining the skew partition: lambda is the outer partition and mu is the inner partition (so mu must be a subpartition of lambda)
weight	integer vector, the weight

Value

List of all skew semistandard tableaux whose shape is the skew partition defined by lambda and mu and whose weight is weight.

Examples

```
ssstx <- skewTableauxWithGivenShapeAndWeight(c(3, 1, 1), c(2), c(1, 1, 1))
lapply(ssstx, prettySkewTableau)
```

```
ssytx_withGivenShapeAndWeight
```

Semistandard Young tableaux with given shape and weight

Description

Enumeration of all semistandard Young tableaux with a given shape and a given weight. The *weight* of a tableau is the vector whose i -th element is the number of occurrences of i in this tableau.

Usage

```
ssytx_withGivenShapeAndWeight(lambda, weight)
```

Arguments

lambda integer partition, the shape
weight integer vector, the weight

Value

List of all semistandard Young tableaux with shape lambda and weight weight.

See Also

[all_ssytx](#).

Examples

```
ssytx <- ssytx_withGivenShapeAndWeight(c(4, 1), c(0, 2, 1, 1, 1))  
lapply(ssytx, prettyTableau)
```

syt2ballot	<i>Tableau as ballot sequence</i>
------------	-----------------------------------

Description

Converts a standard Young tableau to its corresponding ballot sequence.

Usage

```
syt2ballot(syt)
```

Arguments

syt standard Young tableau

Value

A ballot sequence.

See Also

[ballot2syt](#)

Examples

```
syt <- list(c(1,2,6), c(3,5), 4)  
syt2ballot(syt)
```

syt2gprocess	<i>Tableau as growth process</i>
--------------	----------------------------------

Description

Converts a standard Young tableau to its corresponding growth process of partitions.

Usage

```
syt2gprocess(syt)
```

Arguments

syt standard Young tableau

Value

A list of integer partitions, representing a path of the Young graph starting from the root vertex.

See Also

[gprocess2syt](#).

Examples

```
syt <- list(c(1,2,4), 3, 5)
syt2gprocess(syt)
```

syt2matrix	<i>Standard Young tableau as sparse matrix</i>
------------	--

Description

Representation of a standard Young tableau as a sparse matrix.

Usage

```
syt2matrix(syt)
```

Arguments

syt a standard Young tableau

Value

A sparse matrix.

Note

This function is the same as `tableau2matrix` except that in addition it checks that the given tableau is a standard Young tableau.

See Also

`matrix2syt`.

Examples

```
syt <- list(c(1, 2, 6), c(3, 5), 4)
syt2matrix(syt)
```

<code>tableau2matrix</code>	<i>Tableau as sparse matrix</i>
-----------------------------	---------------------------------

Description

Representation of a tableau as a sparse matrix; only for a tableau with numeric or logical entries.

Usage

```
tableau2matrix(tableau)
```

Arguments

`tableau` a tableau with numeric or logical entries

Value

A sparse matrix.

Examples

```
syt <- list(c(1, 2, 6), c(3, 5), 4)
tableau2matrix(syt)
```

tableauShape	<i>Shape of a tableau</i>
--------------	---------------------------

Description

The shape of a tableau.

Usage

```
tableauShape(tableau)
```

Arguments

tableau a tableau (list of vectors having the same mode)

Value

The shape of the tableau. This is an integer partition whose *i*-th part is the number of boxes in the *i*-th row of the tableau.

Examples

```
tableau <- list(c(2, 1, 3), c(5, 2))  
tableauShape(tableau)
```

Index

[all_ssSkewTableaux](#), [2](#)
[all_ssytx](#), [3](#), [3](#), [23](#)
[all_sytx](#), [4](#), [6](#)

[ballot2syt](#), [4](#), [23](#)

[count_ssytx](#), [5](#)
[count_sytx](#), [6](#)

[dualSkewTableau](#), [6](#)
[dualsyt](#), [7](#)
[dualTableau](#), [7](#)

[firstsyt](#), [8](#)

[GelfandTsetlinPatterns](#), [8](#), [21](#)
[gprocess2syt](#), [9](#), [19](#), [24](#)

[hooklengths](#), [10](#), [11](#)
[hooks](#), [10](#), [10](#)

[isSemistandardSkewTableau](#), [11](#)
[isSkewTableau](#), [11](#)
[isSSYT](#), [12](#)
[isStandardSkewTableau](#), [12](#)
[isSYT](#), [13](#)

[KostkaNumber](#), [5](#), [13](#), [15](#), [16](#), [21](#)
[KostkaNumbers](#), [14](#), [14](#), [15](#), [16](#)
[KostkaNumbersWithGivenLambda](#), [14](#), [15](#), [16](#)
[KostkaNumbersWithGivenMu](#), [14](#), [15](#), [15](#), [21](#)

[matrix2syt](#), [16](#), [25](#)

[nextsyt](#), [17](#)

[prettyGT](#), [17](#)
[prettySkewTableau](#), [18](#)
[prettyTableau](#), [18](#)

[rgprocess](#), [19](#)
[RS](#), [19](#)

[rsyt](#), [20](#)

[skewGelfandTsetlinPatterns](#), [9](#), [20](#)
[skewKostkaNumbers](#), [14](#), [21](#)
[skewTableauxWithGivenShapeAndWeight](#), [3](#),
[22](#)
[ssytx_withGivenShapeAndWeight](#), [3](#), [22](#)
[syt2ballot](#), [5](#), [23](#)
[syt2gprocess](#), [9](#), [19](#), [24](#)
[syt2matrix](#), [16](#), [24](#)

[tableau2matrix](#), [25](#), [25](#)
[tableauShape](#), [26](#)