Package ‘quhomology’

May 10, 2018

Type Package

Title Calculation of Homology of Quandles, Racks, Biquandles and Biracks

Version 1.1.1

Date 2018-05-01

Description Calculates the Quandle, Rack and Degenerate Homology groups of Racks and Biracks (as well as Quandles and Biquandles). In addition, a test is provided to ascertain if a given set with one or two given functions is indeed a biquandle or not.

License GPL (>= 3)

Imports MASS, numbers

Depends R(>= 3.0.0)

ByteCompile yes

Suggests testthat

NeedsCompilation no

Author Ansgar Wenzel [aut, cre]

Maintainer Ansgar Wenzel <ansgar.wenzel+quhomology@gmail.com>

Repository CRAN

Date/Publication 2018-05-10 12:16:39 UTC

R topics documented:

---

quhomology-package *Calculation of Homology groups of a rack/birack*

Description

This package provides the functionality to calculate the rack, quandle and degenerate Homology groups of a given rack or birack.

Details

Package:	quhomology
Type:	Package
Version:	1.0
Date:	2014-10-10
License:	GPL v3+

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Ansgar Wenzel

Maintainer: <ansgar.wenzel+quhomology@gmail.com>

References

http://www.maths.sussex.ac.uk/Staff/RAF/Maths/homo.pdf

Examples

```
#Using the up and down action as provided for the dihedral quandle, we can then calculate:

#$H_3^R(R_3)$ by
homology(3,3,FALSE)


#$H_3^Q(R_3)$ by
homology(3,3,TRUE)
```

```
#$H_3^D(R_3)$ by
degenerate_homology(3,3)
```

`boundary_matrix` *This function calculates a boundary matrix.*

Description

This function calculates the boundary matrix for rack/birack for both the quandle and rack homology case. In particular, this is a representation of the boundary function in the simplicial complex of the rack/birack.

Usage

```
boundary_matrix(degree, k, degenerate = FALSE)
```

Arguments

`degree`	This is the degree of the Homology group, that is, if one wants to calculate \$H_3\$, then degree=3. A positive integer.
`k`	This describes the order of the underlying rack or birack. A positive integer.
`degenerate`	If degenerate=TRUE, this calculates the boundary matrix for the quandle homology. If FALSE, the boundary matrix for the rack homology case is returned.

Details

This functions takes all words (or just the non-degenerate ones) of length \$degree\$ in the rack/biquandle (which are represented by \$Z_k\$) and then calculates their boundary via the following equation. For this, let \$x=(x_i)_0^degree-1\$ be an element of the rack/birack and let \$n:=degree-1\$. \$\$partial(x) = Sum_i=0^n (-1)^i ((x_0...(^x_i)...x_n)-(x_0^x_ix_1^x_i...x_i-1^x_ix_i+1_x_i...x_n_x_i))\$\$, where ^x_i means except x_i. If this is a rack rather than a birack, remember that \$f_a()=Id\$.

Value

A Matrix.

References

http://www.maths.sussex.ac.uk/Staff/RAF/Maths/homo.pdf

See Also

`link{boundary_matrix_degenerate}`

Examples

```
boundary_matrix(3,3,TRUE)
```

boundary_matrix_degenerate *Calculation of boundary matrix for degenerate Homology.*

Description

This function returns the boundary matrix of a rack/birack necessary to calculate the degenerate Homology of the same. In particular, this is a representation of the boundary function in the simplicial complex of the rack/birack.

Usage

```
boundary_matrix_degenerate(degree, k)
```

Arguments

degree	This is the degree of the Homology group, that is, if one wants to calculate H_3, then degree=3. A positive integer.
k	This describes the order of the underlying rack or birack. A positive integer.

Details

This functions takes all degenerate words of length $degree$ in the rack/biquandle (which are represented by Z_k) and then calculates their boundary via the followi ng equation. For this, let $x=(x_i)_0^degree-1$ be an element of the rack/birack and let $n:=degree-1$. $$partial(x) = Sum_i=0^n (-1)^i ((x_0...(^x_i)...x_n)-(x_0^x_ix_1^x_i...x_i-1^x_ix_i+1_x_i...x_n_x_i))$$, where ^x_i means except x_i. If this is a rack rather than a birack, remember that $f_a()=Id$.

Value

A matrix.

References

http://www.maths.sussex.ac.uk/Staff/RAF/Maths/homo.pdf

See Also

boundary_matrix

Examples

```
boundary_matrix_degenerate(3,3)
```

`boundary_names` *Calculation of boundary elements for quandle and rack boundary matrix*

Description

This functions calculates the row and column names for both the quandle and the rack boundary matrix.

Usage

```
boundary_names(degree, k, degenerate)
```

Arguments

`degree`	Length of elements to be calculated. A positive integer.
`k`	Order of underlying rack/birack. This will be passed on to up/down action, if necessary. A positive integer.
`degenerate`	If true, remove degenerate entries (and hence calculate the names for the quandle boundary matrix). TRUE/FALSE.

Details

This calculates all possible permutations of elements in Z_k of length $degree$. If degenerate is true, it loops through all of them, removing the degenerate ones (that is, those where $x_i=x_i+1$, for an element $x=(x_i)_0^degree)$).

Value

A matrix with $degree$ columns.

See Also

`boundary_names_degenerate`, `boundary_matrix`

Examples

```
boundary_names(3,3,TRUE)
```

`boundary_names_degenerate` *Calculation of degenerate boundary elements for boundary matrix*

Description

This functions calculates the row and column names for the degenerate boundary matrix.

Usage

```
boundary_names_degenerate(degree, k)
```

Arguments

`degree`	Length of elements to be calculated. A positive integer
`k`	Order of underlying rack/birack. This will be passed on to up/down action, if necessary. A positive integer.

Details

This calculates all possible permutations of elements in Z_k of length $degree$. If degenerate is true, it loops through all of them, removing the non-degenerate ones (that is, those where $x_i =/= x_i+1$ for all $i=0,...,degree-1$, for an element $x=(x_i)_0^degree)$).

Value

A matrix, where the rows represent the elements.

See Also

`boundary_matrix_degenerate`, `boundary_names`

Examples

```
boundary_names_degenerate(3,3)
```

degenerate_homology *Calculates the degenerate Homology for a rack/birack.*

Description

This function calculates the degenerate homology group of a given rack or birack.

Usage

```
degenerate_homology(degree, k, return_values = FALSE)
```

Arguments

degree	This is the degree of the Homology group, that is, if one wants to calculate H^D_3, then degree=3.
k	This describes the order of the underlying rack or birack.
return_values	If return_values = TRUE, the functions returns the diagonal of the Smith Normal Form. If FALSE (the default), the function calls output_results instead which prints the homology group to the screen.

Details

This function is based on the algorithm described in the references below. It should be sufficient for most users to change the up/down action functions according to their requirements and then run the calculation.

Value

NULL if return_values is FALSE, the diagonal of the Smith Normal Form if return_values is TRUE.

References

http://www.maths.sussex.ac.uk/Staff/RAF/Maths/homo.pdf

See Also

homology output_results

Examples

```
degenerate_homology(3,3)
```

down_action *The down action for a birack or biquandle.*

Description

This functions defines the down action for a birack or biquandle. In the case of a quandle or rack, it is the identity. The definition of this functions is $f_b(a)$, that is, b acting on a from below.

Usage

```
down_action(a, b, k)
```

Arguments

a	This is the elements that is acted upon. An integer.
b	This is the element that acts. An integer.
k	This is the order of the biquandle. It is not always required, but passed on nevertheless. An integer.

Details

This can (and should) be changed by the user if s/he requires a different down action. It could be implemented as a matrix lookup, a function or some other way. Examples for the first two options are below.

Value

An integer, representing an element in the birack or rack.

References

http://en.wikipedia.org/wiki/Biquandle http://en.wikipedia.org/wiki/Racks_and_quandles

See Also

up_action

Examples

```
## Example for version with function (for a dihedral quandle)
down_action <- function (a, b, k){

    result <- (2 * b - a)%%k
    return(as.integer(result))
}


##Example for matrix lookup (for commutative quandle over S_3, in which case k = 6)
```

```
down_action <- function (a, b, k){
    #first define the action matrix
    action_matrix <- rbind(c(0,0,0,0,0,0),c(1,1,5,5,2,2),c(2,5,2,1,5,1),
    c(3,4,4,3,4,4),c(4,3,3,3,4,3),c(5,2,1,2,1,5))
    result <-action_matrix[a + 1, b + 1]
    return(as.integer(result))
}

##example for quandles/racks
down_action <- function (a, b, k){

    return(a)
}
```

`GaussianElimination` *Calculation of Gaussian Form of a matrix.*

Description

This function calculates the Gaussian Form of a Matrix as well as the "row change" multiplication matrix, in short, both N (the Gaussian Form) and X for a matrix G of the form: $$ N = X G Y $$

Usage

```
GaussianElimination(A, B, tol = sqrt(.Machine$double.eps),
verbose = FALSE, fractions = FALSE)
```

Arguments

`A`	A Matrix to be turned into Gaussian Form.
`B`	An identity matrix, which will be returned as the row change multiplication matrix.
`tol`	Tolerance for checking for 0 pivot.
`verbose`	If TRUE, print intermediate steps.
`fractions`	If true, try to express nonintegers as rational numbers.

Value

A matrix

Author(s)

John Fox

References

http://socserv.mcmaster.ca/jfox/Courses/R-course-Berkeley/

See Also

rref

Examples

```
test_mat <- matrix(c(2,4,4, -6,6,12, 10,-4,-16), nrow=3, ncol=3, byrow=TRUE)
identity_mat <- diag(3)
GaussianElimination(test_mat,identity_mat)
```

`homology` *Calculation of quandle and rack homology groups of a rack / birack.*

Description

This function calculates the quandle and rack homology groups of a given rack or birack.

Usage

```
homology(degree, k, quandle = TRUE, return_values = FALSE)
```

Arguments

degree	This is the degree of the Homology group, that is, if one wants to calculate H_3, then degree=3.
k	This describes the order of the underlying rack or birack.
quandle	If quandle=TRUE, this calculates the quandle homology group. If FALSE, the rack homology is calculated.
return_values	If return_values = TRUE, the functions returns the diagonal of the Smith Normal Form. If FALSE (the default), the function calls output_results instead which prints the homology group to the screen.

Details

This function is based on the algorithm described in the references below. It should be sufficient for most users to change the up/down action functions according to their requirements and then run the calculation.

Value

NULL if return_values is FALSE, the diagonal of the Smith Normal Form if return_values is TRUE.

Note

Note that the rack/birack is determined by not only k, but also by the up and down actions in up_action and down_action

References

http://www.maths.sussex.ac.uk/Staff/RAF/Maths/homo.pdf

See Also

degenerate_homology down_action up_action output_results

Examples

```
homology(3,3,TRUE)
homology(3,3,FALSE)
```

---

matrix_rank   *Calculates the rank of a matrix.*

Description

This function calculates the rank of a matrix, using Gaussian elimination.

Usage

```
matrix_rank(A)
```

Arguments

A	A matrix, the rank of which one wants to know.

Value

An integer, the rank of the matrix.

See Also

GaussianElimination

Examples

```
test_mat <- matrix(c(2,4,4, -6,6,12, 10,-4,-16), nrow=3, ncol=3, byrow=TRUE)
matrix_rank(test_mat)
#output:
# 2
```

`output_results` *Function that prints the calculated homology group to the screen*

Description

This functions takes the diagonal of the Smith Normal Form of the homology representation and from this prints the homology groups.

Usage

```
output_results(hom_type, Delta, degree, k)
```

Arguments

`hom_type`	This is the type of homology group, one of degenerate (if called from degenerate_homology), quandle (if called from homology(quandle=TRUE)) and rack (if called from homology(quandle=FALSE)).
`Delta`	This is the diagonal of the Smith Normal Form of the homology representation.
`degree`	This is the degree of the Homology group, that is, if one wants to calculate H_3, then degree=3.
`k`	This describes the order of the underlying rack or birack.

Details

This function prints the specified homology group of the given biquandle from the diagonal of the Smith Normal Form of the representation.

In particular, all 1 give nothing, all zeros give a Z each and every other integer n gives a Z_n.

Value

This function does return 0. Otherwise, it is only used for printing output to the screen.

See Also

homology degenerate_homology

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
##H_2^Q(R_3):
output_results("quandle",c(1,1,1,1),2,3)
```

row_space — *Row Space of a Matrix.*

Description

This calculates the space spanned by the rows of a matrix, or, more precisely, a basis for it. This is done via calculation of the Hermite Normal Form of said matrix.

Usage

```
row_space(B)
```

Arguments

B	The matrix whose row space one wants to know.

Details

Calculates row space of a matrix via its hermite normal form.

Value

A Matrix, consisting of the basis of the space spanned by the rows, plus potentially rows of zeros, so the dimensions of this matrix are the same as of the matrix B.

See Also

hermiteNF

Examples

```
test_mat <- matrix(c(2,4,4, -6,6,12, 10,-4,-16), nrow=3, ncol=3, byrow=TRUE)
row_space(test_mat)
```

rref — *Reduced Row Echelon Form of a matrix*

Description

Function calculates the Reduced Row Echelon Form of a matrix.

Usage

```
rref(A, tol = sqrt(.Machine$double.eps), verbose = FALSE, fractions = FALSE)
```

Arguments

`A`	Matrix to be turned into Gaussian Form.
`tol`	Tolerance for checking for 0 pivot.
`verbose`	If TRUE, print intermediate steps.
`fractions`	If true, try to express nonintegers as rational numbers.

Value

A matrix

Author(s)

John Fox

References

http://socserv.mcmaster.ca/jfox/Courses/R-course-Berkeley/

See Also

`GaussianElimination`

Examples

```
test_mat <- matrix(c(2,4,4, -6,6,12, 10,-4,-16), nrow=3, ncol=3, byrow=TRUE)
rref(test_mat)
```

`smith` *Smith Normal Form of a matrix.*

Description

This calculates the Smith Normal Form of a Matrix.

Usage

```
smith(S)
```

Arguments

`S`	A matrix of which one wants to calculate the Smith Normal Form.

Details

This calculates the Smith Normal Form of a Matrix based on repeated calculation of the Hermite Normal Form of the matrix and its transpose.

Value

A matrix.

See Also

check_more_push, push_down, hermiteNF

Examples

```
test_mat <- matrix(c(2,4,4, -6,6,12, 10,-4,-16), nrow=3, ncol=3, byrow=TRUE)
smith(test_mat)
#######
#output:

#  2    0    0
#  0    6    0
#  0    0   12
```

S_test — *Testing of possible quandle/biquandle actions*

Description

This functions tests if a given set with given operations is a biquandle (or quandle), or not.

Usage

```
S_test(k, return_result = FALSE)
```

Arguments

k	Order of set, a positive integer.
return_result	This variable specifies if the results of the tests should be returned (as a list, if TRUE) or if the result of the tests should be printed to the screen (if FALSE, the default).

Details

The test requires the user to define their own up and down actions. The different tests confirm two facts, namely, the bijectivity of the two functions f, g is considered, as well as the bijectivity of the switch map S, via their permutations. Furthermore, via the Yang-Baxter Check, it confirm whether the Yang-Baxter equation holds for the given up and down functions or not.

Value

A vector with 4 boolean entries for the permutation tests for S, f and g, respectively as well as a check that Yang-Baxter holds.

References

add in thesis.

See Also

up_action, down_action

Examples

```
###Using the provided up/down action functions.
S_test(3)
##Output:
"The permutation checks hold that S is TRUE, f is TRUE
and g is TRUE and that the Yang-Baxter check holds TRUE."
```

---

up_action *The up action for a birack or biquandle.*

Description

This function defines the up action for a birack or biquandle. In the case of a quandle or rack, it is the rack or quandle action. The definition of this functions is $f^b(a)$, that is, b acting on a from above.

Usage

```
up_action(a, b, k)
```

Arguments

a	This is the elements that is acted upon. An integer.
b	This is the element that acts. An integer.
k	This is the order of the biquandle. It is not always required, but passed on nevertheless. An integer.

Details

This can (and should) be changed by the user if s/he requires a different up action. It could be implemented as a matrix lookup, a function or some other way. Examples for the first two options are below.

Value

An integer, representing an element in the birack or rack.

References

http://en.wikipedia.org/wiki/Biquandle http://en.wikipedia.org/wiki/Racks_and_quandles

See Also

down_action

Examples

```
## Example for version with function (for a dihedral quandle)
up_action <- function (a, b, k){

    result <- (2 * b - a)%%k
    return(as.integer(result))
}


##Example for matrix lookup (for commutative quandle over S_3, in which case k = 6)
up_action <- function (a, b, k){
    #first define the action matrix
    action_matrix <- rbind(c(0,0,0,0,0,0),c(1,1,5,5,2,2),c(2,5,2,1,5,1),
    c(3,4,4,3,4,4),c(4,3,3,3,4,3),c(5,2,1,2,1,5))
    result <-action_matrix[a + 1, b + 1]
    return(as.integer(result))
}
```

Index