Package 'mathgraph'

March 24, 2004

Version 0.9-4

Date 2004-03-17

Title Directed and undirected graphs

Author Original S code by Patrick J. Burns. Ported to R by Nick Efthymiou

Description Simple tools for constructing and manipulating objects of class mathgraph from the book "S Poetry".

License This software may be re-distributed freely and used for any non-commercial purpose.

Maintainer Nick Efthymiou <Nick.Efthymiou@Schwab.com>

Depends R (>= 1.4.1)

Built R 1.4.1; i386-pc-sco3.2v4.2; Sat Mar 20 08:29:23 PST 2004

R topics documented:

adjamat . . . 2
alldirected . . . 3
build.mathgraph . . . 4
commontail . . . 4
.find.I.of . . . 5
getpath . . . 6
incidmat . . . 7
justify . . . 9
mathgraph . . . 10
names.mathgraph . . . 11
plot.mathgraph . . . 12
print.mathgraph . . . 13
sort.mathgraph . . . 14
stable.apply . . . 15
unique.mathgraph . . . 16

`adjamat` *Adjacency Matrix of a Mathematical Graph*

Description

Returns an object of class `"adjamat"` which is the adjacency matrix of a numbered graph.

Usage

```
adjamat(x, ...)
adjamat.mathgraph(x, general=FALSE, ...)

is.adjamat(x)
```

Arguments

`x`	an object of class `"mathgraph"`.
`general`	logical flag, if `TRUE`, then multiple edges or arcs between the same nodes are counted; otherwise, there is a `1` no matter how many edges or arcs there are.
`...`	other arguments for generic function.

Details

`adjamat` is a generic function with a method for class `"mathgraph"`.

Value

An object of class `"adjamat"` which is a square matrix with as many rows and columns as there are nodes in the numbered graph.
The `i`,`j` element is an indicator of an arc from node `i` to node `j`. An undirected edge between nodes `i` and `j` contributes a `1` to both the `i`,`j` element and the `j`,`i` element.

`is.adjamat` is the membership function for this class.

BUGS

The `general` argument to `adjamat.mathgraph` is not functional.

Note

S Poetry, Patrick J. Burns, http://www.burns-stat.com/pages/spoetry.html

Author(s)

Nick Efthymiou

References

Chachra, V., Ghare, P. M. and Moore, J. M. (1979). Applications of Graph Theory Algorithms. Elvesier North Holland, New York.

Harary, Frank (1969). Graph Theory, p. 158. Addison Wesley.

See Also

mathgraph, incidmat, getpath.adjamat

Examples

```
adjamat(mathgraph(~ 1:3 * 3:5, dir=TRUE))
```

`alldirected` *Transform to Directed Graph*

Description

Returns a `"mathgraph"` object which has all edges directed.

Usage

```
alldirected(x, ...)
```

Arguments

`x`	an object representing a mathematical graph.
`...`	other arguments for derived functions.

Details

The default method merely creates an error.

Value

A `"mathgraph"` object with any undirected edges in the input split into two arcs.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 13.3, p. 305

See Also

mathgraph

Examples

```
my.graph <- mathgraph(~ 1:3 / 2:4) # undirected graph with 3 edges
alldirected(my.graph) # directed graph with 6 arcs
```

build.mathgraph *Internal Function for Mathematical Graphs*

Description

This is an internal function. It is not meant for direct use.

Usage

```
build.mathgraph(formula, data)
```

Arguments

formula	see mathgraph
data	see mathgraph

Note

Temporary objects may be left over when I() expressions are used. The following command cleans them:
rm(list=ls(pattern="Build.mathgraphI"))

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns

See Also

mathgraph

commontail *Common Strings in Tail*

Description

Takes a list of character vectors and returns the longest vector of strings that is common to the ends of all of the components in the list.

Usage

```
commontail(x)
```

Arguments

x	list of character vectors.

Details

This is useful to get the class that is common to a number of objects.

Value

a character vector containing the common elements of the tails of all the components in `x`. The result is `NULL` if there are no common elements.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns

See Also

`inherits`, `intersect`

Examples

```
commontail(list(c("subA", "cls1"), c("subB", "subA", "cls1")))
commontail(list(c("subA", "cls2"), c("subB", "subA", "cls1")))
```

`.find.I.of` *Find limits of I() in a String*

Description

Returns either `NULL` or a two-column matrix where each row gives the first and last character of a call to `I`.

Usage

```
.find.I.of(string, nesting.ok = FALSE)
```

Arguments

`string`	a single character string.
`nesting.ok`	logical value: if `TRUE`, then all occurrences are given. If `FALSE`, then calls to I within other calls to it are ignored.

Value

a numeric matrix with two columns, or `NULL`. Each row represents one call to I.
The first column is the number of the character within the string that starts the call (i.e., the "I").
The second column is the number of the character within the string that ends the call (the ")").

BUGS

Parentheses inside quotes in the call to I will confuse it.

One or more spaces after the "I" will cause the call to be unrecognized. However, if the string is parsed and then deparsed, the space will go away.

Note

This is used by mathgraph functions, and is not meant for direct use.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 11.2, p. 280

See Also

build.mathgraph

Examples

```
.find.I.of("~ x + I(x^2)")
.find.I.of("~ x + I(x^2 + I(y^3))")

.find.I.of("~ x + I(x^2 + I(y^3))", TRUE)
```

---

getpath *Find a Path in a Mathematical Graph*

Description

Returns a path, if it exists, from the `start` to the `end`.

Usage

```
getpath(x, start, end, ...)
getpath.mathgraph(x, start, end, ...)
```

Arguments

`x`	an object representing a mathematical graph.
`start`	character string or integer giving the starting node.
`end`	character string or integer giving the ending node.
`...`	generic arguments.

Details

`getpath` is a generic function with methods for `"mathgraph"`, `"incidmat"` and `"adjamat"`. The default method converts `x` to class `"incidmat"`.

`getpath.adjamat` is an implementation of algorithm 2.2 in Chachra, Ghare and Moore (1979) and `getpath.incidmat` is an implementation of their algorithm 2.3.

The distinction between non-existent paths and paths of length zero may be useful in some situations.

Value

When at least one path exists, a `"mathgraph"` object containing the edges within the first path found; this may be an empty mathgraph if `start` and `end` are equal.
When no path exists, returns `NULL`.

Note

S Poetry, Patrick J. Burns, Section 13.3

Author(s)

Nick Efthymiou

References

Chachra, V., Ghare, P. M. and Moore, J. M. (1979). Applications of Graph Theory Algorithms. Elvesier North Holland, New York.

See Also

`mathgraph`, `incidmat`, `adjamat`

Examples

```
getpath(mathgraph(~ 1:3 / 3:5), 1, 5) # returns a path
getpath(mathgraph(~ 1:3 / 3:5), 1, 4) # no path, returns NULL
getpath(mathgraph(~ 1:3 / 3:5), 1, 1) # returns mathgraph()
```

`incidmat` *Incidence Matrix for a Mathematical Graph*

Description

Returns an object of class `"incidmat"` which is a matrix indicating the start and end node for each edge in the graph.

Usage

```
incidmat(x, ...)
incidmat.mathgraph(x, expand=TRUE, general=FALSE, ...)

is.incidmat(x)
```

Arguments

x	object representing a mathematical graph.
expand	logical flag: if TRUE, then undirected edges are represented by two columns in the output. If FALSE, then both non-zero elements of an undirected edge are positive.
general	logical flag: if TRUE, then there is a non-zero entry in a column representing a loop.
...	generic arguments.

Details

The incidmat function is generic, with a method for class "mathgraph".

is.incidmat is the membership function for this class.

Value

an object of class "incidmat" which is a matrix with rows representing nodes and columns representing edges.
Generally speaking, there is a 1 in the location where an edge begins and a -1 in the location where it ends.
Frank Harary defines incidence matrix with expand = FALSE.

Note

From S Poetry, Patrick J. Burns, Section 13.3, p. 312 http://www.burns-stat.com/pages/spoetry.html

Author(s)

Nick Efthymiou

References

Chachra, V., Ghare, P. M. and Moore, J. M. (1979). Applications of Graph Theory Algorithms. Elvesier North Holland, New York.

Harary, Frank (1969). Graph Theory, p. 160. Addison Wesley.

See Also

adjamat, mathgraph, getpath.incidmat

Examples

```
incidmat(mathgraph(~ 1:3 / 3:5, dir=TRUE))
incidmat(mathgraph(~ 1:3 / 3:5, dir=FALSE))
incidmat(mathgraph(~ 1:3 / 3:5, dir=FALSE), expand=FALSE)
```

`justify` *Justify Elements of a Vector*

Description

Returns a vector like the input, but each string may have added blank spaces at the start and/or end.

Usage

```
justify(x, type = "r")
```

Arguments

`x`	a character vector.
`type`	a string giving the type of justification. This may be an abbreviation of one of ”right”, ”left”, ”center”.

Value

a character vector like x, except all elements have the same number of characters, and the text is lined up along one edge, or centered.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns

See Also

`format`, `substring`, `paste`

Examples

```
data(freeny)
as.matrix(justify(dimnames(freeny.x)[[2]], "r"))
as.matrix(justify(dimnames(freeny.x)[[2]], "l"))
as.matrix(justify(dimnames(freeny.x)[[2]], "c"))
```

`mathgraph` *Create Mathematical Graph*

Description

Create an object of class `"mathgraph"` which represents a mathematical graph.

Usage

```
mathgraph(formula, directed = FALSE, data = sys.parent())
length.mathgraph(x)
c.mathgraph(...)

is.mathgraph(x)
```

Arguments

`formula`	a formula containing just the right-side. Special operators in the formula are `+` which separates terms, `/` which puts an edge between corresponding elements of the two vectors on which it is operating, and `*` which puts an edge between every pair of elements in the two vectors on which it is operating.
`directed`	logical flag: if `TRUE`, then all edges that are created are directed, otherwise they are undirected.
`data`	the frame in which to find objects referenced in the formula. This can be either the number of a memory frame, or a list or data frame containing the data.
`...`	objects to be concatenated.
`x`	object of class `"mathgraph"`.

Details

Mathematical graphs consist of a set of nodes (vertices) and edges. Edges go between two nodes. An edge that is directed is often called an arc.

Terms in the formula (delimited by `+`) may be either calls to `*` or `/`, or objects that are already of class `"mathgraph"`.

Two other representations of graphs are adjacency matrices and incidence matrices. The functions to convert `"mathgraph"` objects to these are `adjamat` and `incidmat`, respectively. Most algorithms for mathematical graphs are in terms of incidence matrices or adjacency matrices.

The generic functions that have a method for class `"mathgraph"` include: `[`, `c`, `length`, `names`, `plot`, `print`, `unique`.

`is.mathgraph` is the membership function for this class.

Value

an object of class `mathgraph` which is a two-column matrix of nodes along with an additional attribute called `"directed"` which is a logical vector stating whether or not each edge is directed.
An edge (row of the matrix) that is directed goes from the node in the first column to the node in the second column.

Note

S Poetry, Patrick J. Burns, http://www.burns-stat.com/pages/spoetry.html

Author(s)

Nick Efthymiou

References

Chachra, V., Ghare, P. M. and Moore, J. M. (1979). Applications of Graph Theory Algorithms. Elvesier North Holland, New York.

See Also

adjamat, incidmat, getpath

Examples

```
mathgraph(~ 1:3 / 2:4) # graph with 3 edges
mathgraph(~ 1:3 * 2:4) # graph with 9 edges

mathgraph(~ 1:3 / 2:4, dir=TRUE) # directed graph with 3 edges

# graph with some edges directed, some not
c(mathgraph(~ 1:3 * 2:4), mathgraph(~ c(3,1) / c(2,4), dir=TRUE))
```

---

names.mathgraph *Edge Names in a Mathematical Graph*

Description

Sets or returns the names (corresponding to the edges) of a mathematical graph represented by a `mathgraph` object.

Usage

```
names.mathgraph(x)
names.mathgraph(x) <- value
```

Arguments

`x`	an object inheriting from `mathgraph`.
`value`	a value to be assigned to the names of the `mathgraph`.

Details

In the assignment form, the names are created or changed.

Value

Character vector of the names.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 13.3 Mathematical Graphs, http://www.burns-stat.com/pages/spoetry.html

See Also

mathgraph

Examples

```
jjm <- mathgraph(~ 1:3 * 2:4)
jjm
names(jjm) <- letters[1:length(jjm)]
jjm
names(jjm)
```

---

plot.mathgraph *Plot a Mathematical Graph*

Description

Very crude plotting method for `mathgraph` class.

Usage

```
plot.mathgraph(x, ...)
```

Arguments

`x`	an object that inherits from `mathgraph`.
`...`	graphics parameters may be given.

Details

A representation of the mathematical graph is produced on the current graphics device.

BUGS

Needs to be smarter, and allow the user some control.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 13.3

See Also

mathgraph, plot, par

Examples

```
plot(mathgraph(~ 1:3 * 2:4), main="Graph K4")
```

`print.mathgraph` *Print a Mathematical Graph*

Description

Prints a representation of the graph.

Usage

```
print.mathgraph(x, prefix.node = if (is.character(xu)) "" else "node", ...)
```

Arguments

`x`	an object inheriting from mathgraph which represents a mathematical graph.
`prefix.node`	a string to put in front of each node named. The default is an empty string if the nodes are character and the string "node" if they are not.
`...`	other arguments to print may be given, but are not used.

Details

The object is printed.
A '–' between nodes means an undirected edge, while a single arrow means a directed edge.

Value

the input x is returned invisibly.

Note

The format is consistent with the dot graph language.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 13.3, Mathematical Graphs

See Also

mathgraph, names.mathgraph

Examples

```
mathgraph(~ 1:3 / 2:4)
mathgraph(~ 1:3 / 2:4, dir=TRUE)
jjm <- mathgraph(~ letters[1:3] * letters[2:4])
jjm
names(jjm) <- LETTERS[1:9]
jjm
```

`sort.mathgraph` *Sort a Mathematical Graph*

Description

Sorts nodes within undirected edges and/or edges by nodes.

Usage

```
sort.mathgraph(x, nodes = TRUE, edges = TRUE)
```

Arguments

`x`	an object that inherits from `mathgraph`.
`nodes`	logical value; if `TRUE`, then the nodes within undirected edges are sorted.
`edges`	logical value; if `TRUE`, then the edges are sorted by the first node with ties broken by the second node.

Value

an object that represents the same graph as the input, but with some rearrangement.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 13.3; http://www.burns-stat.com/pages/spoetry.html

See Also

`mathgraph`

Examples

```
jjmg <- c(mathgraph(~ 4:2 * 1:3), mathgraph(~ 3:5 / 1:3))
sort.mathgraph(jjmg)
sort.mathgraph(jjmg, node=FALSE)
sort.mathgraph(jjmg, edge=FALSE)
```

`stable.apply` *Apply with Stable Dimensions*

Description

Does the same thing as `apply` except that when the function returns a vector, the dimensions are put back the way they started.

Usage

```
stable.apply(X, MARGIN, FUN, ...)
```

Arguments

`X`	same as in `apply`.
`MARGIN`	same as in `apply`.
`FUN`	same as in `apply`.
`...`	same as in `apply`.

Value

When `FUN` returns a scalar or when `MARGIN` has a length that is not one less than the number of dimensions in `X`, then the same as `apply`.

Otherwise, an array similar to the result of `apply`, but with the dimensions permuted to correspond to the dimensions of `X`.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, http://www.burns-stat.com/pages/spoetry.html

See Also

`apply`, `aperm`

Examples

```
data(freeny)
stable.apply(freeny.x, 1, sort)
# compare to:
apply(freeny.x, 1, sort)
```

`unique.mathgraph` *Unique Edges of a Mathematical Graph*

Description

Returns a `mathgraph` object that may have fewer edges than the input.

Usage

```
unique.mathgraph(x, incomparables = FALSE, ...)
```

Arguments

`x`	an object that inherits from `mathgraph`.
`incomparables`	a vector of values that cannot be compared. The only possible value is `FALSE`, meaning that all values can be compared.
`...`	other arguments for generic function.

Value

An object that is the same class as the input `x`, but redundant edges are removed.

Author(s)

Nick Efthymiou

References

S Poetry, Patrick J. Burns, Section 13.3

See Also

`mathgraph`

Examples

```
unique( c(mathgraph(~ 1:2*2:3), mathgraph(~ 1/3)) )
```