Package 'rPorta'

February 15, 2012

Version 0.1-9

Date 2009-03-26

Title R/PORTA interface

Author Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>, Silke Straatmann, Simone Wenzel

Maintainer Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

Depends R (>= 2.6.0)

Imports methods

Description An R interface to a modified version of PORTA. For more information on PORTA see http://www.zib.de/Optimization/Software/Porta/.

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-03-30 18:25:22

R topics documented:

`as.ieq` *Coerce to ieq object*

Description

`as.ieq` attempts to turn its argument into an `ieq` object.

Usage

```
as.ieq(v, sign=NULL)
```

Arguments

`v`	argument that should be turned into an `ieq` object
`sign`	vector of signs (-1 for <=, 0 for ==, and 1 for >=). Default is >=

Details

Handles the same inputs as `as.poi` but the input has to be a matrix with at least two columns.

Value

Returns an `ieq` object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

`as.poi`, "`ieq`"

Examples

```
# x1,x2,x3>=0
as.ieq(cbind(diag(3),rep(0,3)))

# x1,x2,x3==0
as.ieq(cbind(diag(3),rep(0,3)),sign=rep(0,3))
```

`as.ieqFile` *Coerce to ieqFile object*

Description

`as.ieqFile` attempts to turn its argument into an `ieqFile` object.

Usage

```
as.ieqFile(v, sign=NULL)
```

Arguments

`v`	argument that should be turned into an `ieqFile` object
`sign`	vector of signs (-1 for <=, 0 for ==, and 1 for >=). Default is >=

Details

Constructs an `ieqFile` object consisting only of the inequalities with the help of `as.ieq`.

Value

Returns an `ieqFile` object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

`as.ieq`, `"ieqFile"`

Examples

```
# x1,x2,x3>=0
as.ieqFile(cbind(diag(3),rep(0,3)))

# x1,x2,x3==0
as.ieqFile(cbind(diag(3),rep(0,3)),sign=rep(0,3))
```

as.poi *Coerce to poi object*

Description

as.poi attempts to turn its argument into a poi object.

Usage

```
as.poi(v)
```

Arguments

v	argument that should be turned into a poi object

Details

Matrices with numeric values are transformed to rational values. This is done in a way that assumes that the numeric values in v represent rational values, e.g. as.poi(0.8)=4/5, but as.poi(1/3)=333333333/1e+09. For exact transformation, one possibility is to use the R package gmp, which gives the exact rational values, e.g.
as.bigq(0.8)=3602879701896397/4503599627370496
as.bigq(1/3)=6004799503160661/18014398509481984
as represented by the computer.

as.poi also transforms strings like "1/3" to poi objects.

Value

Returns a poi object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"poi"

Examples

```
# Convert the numeric value 0.8 to a poi object
as.poi(0.8)

# Convert the character value "1/3" to a poi object
as.poi("1/3")

# Convert a matrix to a poi object
as.poi(matrix(1:16/4,ncol=4))
```

`as.poiFile` *Coerce to poiFile object*

Description

`as.poiFile` attempts to turn its argument into an `poiFile` object.

Usage

```
as.poiFile(v, hull=TRUE)
```

Arguments

`v`	argument that should be turned into an `poiFile` object
`hull`	determines whether the convex_hull or the convex_cone slot should be used

Details

Constructs an `poiFile` object consisting only of the points of the convex cone or the convex hull with the help of `as.poi`.

Value

Returns an `poiFile` object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

`as.poi`, "`poiFile`"

Examples

```
# (1,0,0), (0,1,0), and (0,0,1) as convex hull points
as.poiFile(diag(3))

# # (1,0,0), (0,1,0), and (0,0,1) as convex cone points
as.poiFile(diag(3),hull=FALSE)
```

convertToString-method *Method "convertToString"*

Description

Converts some of the objects provided by rPorta to character strings, namely `poi`, `ieq`, `poiFile`, and `ieqFile`. These character strings correspond to the format used by Porta.

Usage

```
## S4 method for all signatures:
convertToString(object)
```

Arguments

`object`	Object of one of the supported classes (`poi`, `ieq`, `poiFile`, or `ieqFile`)

Value

Returns a character string in PORTA's format.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieq", "poi", "ieqFile", "poiFile"

Examples

```
# Returns a String representation of a poiFile object
convertToString(example.poi())

# Returns a String representation of an ieqFile object
convertToString(example.ieq())
```

example.ieq *Creates an object representing PORTA's example.ieq*

Description

Creates an object representing one of PORTA's example files (example.ieq).

Usage

```
example.ieq()
```

Value

Returns an `ieqFile` object encapsulating this text file:

```
DIM = 5
VALID
3 3 0 2 3

LOWER_BOUNDS
0 1 2 2 2
UPPER_BOUNDS
2 2 2 5 5

ELIMINATION_ORDER
2 0 1 0 3

INEQUALITIES_SECTION
(  1) +27x1-28x2+57x4-37x5 == 0
(  2)           -  x4+  x5 == 1

(  1) + x2   -2x5 <=  -3
(  2)     -x3     <=   0
-2x2   + x5 <=   0
-4/15x2-1/15x5 >=  -1

END
```

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile"

example.poi *Creates an object representing PORTA's example.poi*

Description

Creates an object representing one of PORTA's example files (example.poi).

Usage

```
example.poi()
```

Value

Returns an `poiFile` object encapsulating this text file:

```
DIM = 3

CONV_SECTION
 3 3 0
 5/3 1 0
 1 5/2 0
CONE_SECTION
 0 0 2/3

END
```

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"poiFile"

failureRegions *Determination of unfeasible regions*

Description

Determines unfeasible regions inside a given parameter space based on emerging failure parts, i.e. missing values, in earlier experiments.

Usage

```
failureRegions(experiments, parameterspace, fail)
```

Arguments

`experiments`	An object of class `poi`. The rows contain the points with known results, e.g. already solved experiments.
`parameterspace`	An object of class `poi` representing the parameter space as a grid. The rows should contain the points of the grid.
`fail`	A logical vector indicating which of the points of `experiments` are failure points. TRUE for failure points, FALSE for non-failure points.

Details

The unfeasible regions are determined as follows. Suppose the design space is covered roughly with a set of experiments specified in `experiments` and the results contain some failure points and some non-failure points which is indicated by `fail`. Assuming the feasible area is convex, the space lying, from the viewpoint of the non-failure point, behind the failure points has to be part of the failure region. For each non-failure point a polyhedral convex cone is spanned over all non-failure points. Thereby the minimum needed polyhedral cone is constructed, i.e. only the extreme rays are used. Then each cone is mirrored to the opposite side of the failure points. Those points from `parameterspace` laying inside one of the mirrored cones then belong to the unfeasible region.

Value

`failureRegions` returns an object of class `parameterrestriction`.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

References

N. Henkenjohann, R. G\"obel, M. Kleiner, and J. Kunert (2005). An Adaptive Sequential Procedure for Efficient Optimization of the Sheet Metal Spinning Process. *Quality and Reliability Engineering International.* **21** (5), 439–455.

See Also

"paramspacerestriction", "poi"

Examples

```
# Construct a 3-dimensional grid with values 0.0,0.2,...,1.0
parameterspaceNum<-matrix(nrow=216,ncol=3)
for (i in(0:215)) parameterspaceNum[i+1,]<-c(i%/%6^2,i%/%6%%6,i%%6)
parameterspace<-new("poi",num=parameterspaceNum,den=matrix(5,nrow=216,ncol=3))

# Construct a poi object with 10 experiments
experimentsNum<-matrix(c(
4,1,4,
2,1,0,
1,1,5,
1,4,4,
```

```
5,1,1,
1,1,2,
4,4,4,
4,4,1,
1,1,1,
1,4,1
),byrow=TRUE,nrow=10,ncol=3)
experiments<-new("poi",num=experimentsNum,den=matrix(5,nrow=10,ncol=3))

# Declare which experiments succeeded
fail <- c(0,1,0,0,0,1,0,0,1,0)

# Start
result<-failureRegions(experiments,parameterspace,fail)
show(result)

# Show summary
getParamspaceInfo(result)
```

`fctp-method` *Method "fctp"*

Description

Checks inequalities for facet inducing property.

Usage

```
## S4 method for signature 'ieqFile':
fctp(object,poiObject)
```

Arguments

`object`	Object of class ieqFile
`poiObject`	Object of class poiFile

Details

fctp performs a check whether the inequalities given in `object` are facet inducing for the polyhedron given by `poiObject`.

For all inequalities `fctp` does the following:

In a first step fctp checks if the inequality is valid. If this is not the case `fctp` returns the points and rays which are not valid.

In a second step `fctp` computes those valid points and rays which satisfy the inequality with equality and - if there are any - returns them.

Value

Returns one or more objects of class `poiFile`.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile", "poiFile"

Examples

```
# A very simple example
ieqExample<-as.ieqFile(cbind(diag(3),rep(0,3)))
poiExample<-as.poiFile(rbind(diag(3),rep(-1,3)))
fctp(ieqExample,poiExample)
```

---

fmel-method *Method "fmel"*

Description

Projection of linear system on subspaces xi = 0.

Usage

```
## S4 method for signature 'ieqFile':
fmel(object,chernikov_rule_off=FALSE,long_arithmetic=FALSE)
```

Arguments

`object`	Object of class ieqFile
`chernikov_rule_off`	Fourier-Motzkin elimination without using the rule of Chernikov
`long_arithmetic`	Use long integers for intermediate results.

Details

`fmel` takes a system of linear inequalities as input and eliminates choosen variables. That is, `fmel` projects the given system to the subspace given by xi = 0, for i is contained in I, where I is the index set of the variables that should be eliminated.

The set I and the elimination order are given in the input object in the slot `elimination_order`, containing exactly dim integers, where dim is the dimension of the problem.

A '0' as the i-th entry of `elimination_order` indicates that the i-th variable should not be eliminated, that is, i is not in I. An entry 'j', for 0 < j < dim, as the i-th entry of `elimination_order` indicates that the i-th variable should be eliminated in the j-th iteration. (All nonzero numbers must be different and it must be possible to put them into an order 1,2,3,4...)

Value

Returns the resulting inequalities as an `ieqFile` object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile"

Examples

```
# Fourier-Motzkin elimination of an ieqFile
fmel(example.ieq())
```

---

ieq-class *Class "ieq"*

Description

Encapsulates inequalities in PORTA's ieq format.

Objects from the Class

An `ieq` object holds three slots.

Slots

`num`: Matrix containing the numerators of the inequalities

`den`: Matrix containing the denominators of the inequalities

`sign`: Vector containing either -1,0, or 1 for each inequality, corresponding to '<=', '==', and '>='

Methods

convertToString Convert to character string in PORTA's format

getNumerator Returns this objects numerator matrix

getDenominator Returns this objects denominator matrix

index May be used to obtain subsets of the object

Extends

Class "poi", directly.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile", "poi", "poiFile"

---

ieqFile-class *Class "ieqFile"*

Description

Encapsulates PORTA files describing polyhedra in PORTA's ieq file format.

Objects from the Class

An ieqFile object holds six slots.

Slots

valid: May be used to specify a valid point for method traf

lower_bounds: May be used to specify lower bounds for method vint

upper_bounds: May be used to specify upper bounds for method vint

elimination_order: May be used to specify the elimination order for method fmel

inequalities: ieq object describing the inequalities

strong_validity: Only used for return values containing a strong validity table

Methods

convertToString Convert to character string in PORTA's format

Extends

Class "portaFile", directly.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieq", "poi", "poiFile"

Examples

```
# Show the definition of example.ieq to provide an example
example.ieq

# A second example
ieqFileExample <- new("ieqFile", inequalities=new("ieq", num=matrix(1:100, ncol=10),
den=matrix(rep(c(1,3,5,6),25), ncol=10), sign=rep(c(1,0,-1,0,1),2)))
```

`iespo-method` *Method "iespo"*

Description

Enumeration of equations and inequalities that are valid for a convex cone and a convex hull.

Usage

```
## S4 method for signature 'ieqFile':
iespo(object,poiObject,validity_table_out = FALSE)
```

Arguments

`object`	Object of class ieqFile
`poiObject`	Object of class poiFile
`validity_table_out`	Include a table in the output which indicates strong validity.

Details

`iespo` is a simple enumeration routine which enumerates the subset of equations and inequalities in `object` which are valid (not necessarily facet inducing) for the polyhedron given by the `poiObject`.

Value

Returns an `ieqFile` object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile", "poiFile"

paramspacerestriction-class
Class "paramspacerestriction"

Description

Encapsulates information returned by `failureRegions`.

Objects from the Class

An `paramspacerestriction` object holds five slots.

Slots

`parameterspace`**:** The original `poi` object representing the parameter space.

`feasiblepoints`**:** Set of indices describing all points from parameterspace that do not lie inside one of the polyhedral convex cones, i.e. the feasible area.

`excludingcone`**:** Set of indices describing which cone led to the exclusion of points from parameterspace

`cones`**:** Set of the polyhedral convex cones constructed by `failureRegions`

`conesources`**:** Set of indices describing the points the cones are based on

Methods

convertToString Convert to character string

getUnfeasiblePoints Returns the set of unfeasible points

getFeasiblePoints Returns the set of feasible points

getParamspaceInfo Returns a summary of the parameter space restriction. The first columns contain the points of the parameter space, the column named with PCC gives the set of indices in which cone the point is included, where 0 means that the point is not included in any cone. The last column PCC-Source gives the point which is the cone end.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

`failureRegions`

poi-class *Class "poi"*

Description

Encapsulates points in PORTA's poi format.

Objects from the Class

A poi object holds two slots.

Slots

num: Matrix containing the numerators of the points

den: Matrix containing the denominators of the points

Methods

as.matrix Converts the values to a numeric matrix.

convertToString Convert to character string in PORTA's format

getNumerator Returns this objects numerator matrix

getDenominator Returns this objects denominator matrix

index May be used to obtain subsets of the object

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"poiFile", "ieq", "ieqFile", as.poi

Examples

```
# Construct a 3-dimensional grid with values 0.0,0.2,...,1.0
parameterspaceNum<-matrix(nrow=216,ncol=3)
for (i in(0:215)) parameterspaceNum[i+1,]<-c(i%/%6^2,i%/%6%%6,i%%6)
parameterspace<-new("poi",num=parameterspaceNum,den=matrix(5,nrow=216,ncol=3))
```

poiFile-class *Class "poiFile"*

Description

Encapsulates PORTA files describing polyhedra in PORTA's poi file format.

Objects from the Class

A `poiFile` object holds three slots.

Slots

`convex_hull`**:** poi object containing a set of points describing the polyhedron

`convex_cone`**:** poi object containing a set of vectors spanning a convex cone

`strong_validity`**:** Only used for return values containing a strong validity table

Methods

convertToString Convert to character string in PORTA's format

dim `dim` computes the dimension of the convex hull and the convex cone of a given set of points by using a gaussian elimination algorithm. Moreover, in the case that the input system is not full dimensional, `dim` displays the equations satisfied by the system.

Extends

Class `"portaFile"`, directly.

Author(s)

Robin Nunkesser `<Robin.Nunkesser@tu-dortmund.de>`

See Also

`"poi"`, `"ieq"`, `"ieqFile"`

Examples

```
# Show the definition of example.poi to provide an example
example.poi

# A second example
hull <-  as.poi(matrix(1:30,ncol=3))
cone <- new("poi", num=matrix(c(0,0,2), nrow=1,ncol=3), den=matrix(c(1,1,3),nrow=1,ncol=3))
poiFileExample<-new("poiFile", convex_hull=hull, convex_cone=cone)
```

portaFile-class *Class "portaFile"*

Description

Encapsulates PORTA files in PORTA's ieq and poi file format. Is the abstract superclass of "ieqFile" and "poiFile".

Methods

convertToString Convert to character string in PORTA's format

writeToFile Writes the character string to a file

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile", "poiFile"

portsort-method *Method "portsort"*

Description

Sort inequality or point systems.

Usage

```
## S4 method for signature 'portaFile':
portsort(object)
```

Arguments

object	Object of class portaFile

Details

portsort puts the points or inequalities into an increasing order according to the following criteria:

1. right hand sides of inequalities or equations
2. frequency of the values -5 .. -1, 1 .. 5
3. lexicographical order

Value

Returns a sorted version of the input object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"portaFile"

Examples

```
# Sorting of a poiFile object
portsort(example.poi())

# Sorting of an ieqFile object
portsort(example.ieq())
```

---

posie-method *Method "posie"*

Description

Enumeration of points that are valid for a linear system.

Usage

```
## S4 method for signature 'ieqFile':
posie(object,poiObject)
```

Arguments

object	Object of class ieqFile
poiObject	Object of class poiFile

Details

posie is a simple enumeration routine which determines the number of the points and direction vectors in poiObject which are valid for the linear system in object.

Value

Returns the valid points in an poiFile object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile", "poiFile"

---

read.portaFile *Parse Porta files*

Description

read.portaFile attempts to turn the .poi or .ieq file given by its argument into a portaFile object.

Usage

```
read.portaFile(file)
```

Arguments

file	the name of the file which should be turned into a portaFile object

Value

Returns a poiFile or ieqFile object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"portaFile", "poiFile", "ieqFile"

Examples

```
## Not run:
# Could be used for one of the example files provided by PORTA
read.portaFile("example.poi")

## End(Not run)
```

traf-method *Method "traf"*

Description

Transformation of polyhedron representations.

Usage

```
## S4 method for signature 'portaFile':
traf(object, opt_elim = FALSE, chernikov_rule_off = FALSE,
validity_table_out = FALSE,long_arithmetic=FALSE)
```

Arguments

object	Object of class portaFile
opt_elim	Use a heuristic to eliminate that variable next, for which the number of new inequalities is minimal (local criterion).
chernikov_rule_off	Fourier-Motzkin elimination without using the rule of Chernikov
validity_table_out	Include a table in the output which indicates strong validity.
long_arithmetic	Use long integers for intermediate results.

Details

traf transforms polyhedra between the representations poiFile (convex hull of points + convex cone of vectors) and ieqFile (system of linear equations and inequalities). The direction of transformation is determined by the class of object. All computations are carried out in rational arithmetic to have guaranteed correct numerical results. Rational arithmetic uses only integer operations.

The computation of the ieq-representation is performed using Gaussian and Fourier-Motzkin elimination. In the output inequalities the right hand sides are 0, or determined by the smallest integer value for which the coefficients of the inequality are integral. If this is not possible with system integer arithmetic

, the right hand sides are 0 or 1 or -1 and the values are reduced as far as possible. The resulting inequalities are all facet-defining for your polyhedron and give together with equations a minimal linear description of your polyhedron.

If an 'ieq'-representation is given as input and if 0 is not valid for the linear system, traf needs a valid point. Such a valid point may be specified in the slot valid of objects of class ieqFile. traf transforms the ieq representation to the poi-representation, after elimination of equations and 0-centering, by applying the 'poi'-to-'ieq' direction to the polar polyhedron.

Hint: If you give a valid point or if 0 is valid, then this vector may appear again in the resulting system, even if this vector might be redundant in a minimal description. (All other vectors are non-redundant.)

Value

Returns an object of class `poiFile` or `ieqFile` depending on the direction of transformation.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

References

K. Fukuda and A. Prodon (1996). Double Description Method Revisited. In: *Combinatorics and Computer Science.* Vol. 1120 of Lecture Notes in Computer Science. Springer, London, 91–111.

See Also

"portaFile", "poiFile", "ieqFile"

Examples

```
# Convert a poiFile object to an ieqFile object
traf(example.poi())

# Convert an ieqFile object to a poiFile object
traf(example.ieq())
```

`vint-method` *Method "vint"*

Description

enumeration of integral inner points of a linear system

Usage

```
## S4 method for signature 'ieqFile':
vint(object)
```

Arguments

`object`	Object of class ieqFile

Details

`vint` enumerates all integral points within given bounds that are valid for a linear system.

In `object` lower and upper bounds for each component must be given. The correponding slots are `lower_bounds` and `upper_bounds`. They have to contain exactly dim integers. The i-th entry of such a line gives the upper resp. lower bound for the i-th component.

Value

Returns the found integral points in a `poiFile` object.

Author(s)

Robin Nunkesser <Robin.Nunkesser@tu-dortmund.de>

See Also

"ieqFile"

Examples

```
# A very simple example
example<-as.ieqFile(cbind(diag(3),rep(0,3)))
example@lower_bounds<-c(-1,-1,-1)
example@upper_bounds<-c(1,1,1)
vint(example)
```

Index