Package 'kstIO'

April 11, 2026

Version 0.5-1

Date 2026-04-11

Title Knowledge Space Theory Input/Output

Description Knowledge space theory by Doignon and Falmagne (1999) <doi:10.1007/978-3-642-58625-5> is a set- and order-theoretical framework which proposes mathematical formalisms to operationalize knowledge structures in a particular domain. The 'kstIO' package provides basic functionalities to read and write KST data from/to files to be used together with the 'kst', 'kstMatrix', 'pks', or 'DAKS' packages.

License GPL (>= 3)

Depends R (>= 4.4.0), sets, kstMatrix (>= 2.3-0), openxlsx2, readODS, tools

Author Cord Hockemeyer [aut, cre]

Maintainer Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

NeedsCompilation no

Repository CRAN

Encoding UTF-8

Date/Publication 2026-04-11 12:30:02 UTC

Contents

`kstIO-package` `kstIO` *File formats*

Description

Knowledge space theory by Doignon and Falmagne (1985, 1999) is a set- and order-theoretical framework, which proposes mathematical formalisms to operationalize knowledge structures in a particular domain. The 'kstIO' package provides basic functionalities to read and write KST data from/to files.

Details

This page focuses on the different file formats that can be used with the `kstIO` functions.

File Formats

Over time and in different research groups with knowledge space theory, different file formats have evolved.

Matrix Format: The probably simplest and most direct approach is to store the information in a binary ASCII matrix where a "1" in row i and column j means that item j is element of state/response pattern i.

There is no separating character between the columns, and there should be no trailing whitespace at the end of the line. The last line of the matrix must carry an EndOfLine - in most editors (except vi) this means an empty line after the matrix.

KST Tools Format: This format (Hockemeyer, 2001) extends the matrix format by two preceding header lines containing the number of items and the number of states/response patterns, respectively.

SRBT Tools Format: This format (Poetzi & Wesiak, 2001) extends the KST tools format by yet another preceding header line with format and content metadata. This new header line has the format

```
#SRBT v2.0 <struct> ASCII <comment>
```

where `<struct>` specifies the type of data stored in the file and `<comment>` is an optional arbitrary comment.

Furthermore, SRBT files may contain an arbitrary number of comment lines after the number of states/patterns.

The following data types are supported by the respective `kstIO` functions:

- basis
- data
- relation
- space
- structure

For `kbase` and surmise `relation` files, the encoding information `"ASCII"` is missing because these files are always in ASCII format.

Spreadsheet Formats: Starting with version 0.5-0, `kstIO` supports various spreadsheet formats (CSV, XLSX, and ODS). The automatic format detection in the `read...()` functions detects these formats based on the filename extension. As explicit format specification, the following names are recognised: CSV, XLSX, and ODS.

The spreadsheet filers themselves contain one sheet only (the `read...()` functions read only the first sheet of the file) containing an optional (but default) header row with item names and the matrix (or data frame in case of surmise functions).

Special File/Data Types

Base Files: Base files are not availablein KST tools format.

Their matrix part differs from the other files in that it contains "0", "1", and "2". A "1" means that the state is minimal for the item and a "2" means that it is not (but contains the item). A "0" stands (as always) for the state not containing the item.

Surmise Relation Files: Surmise relation files are not available in KST Tools format, either.

Their matrices are somewhat transposed in comparison to all the other formats. Row i and column j is equal to "1" if knowing i can be surmised from knowing j, and equal to "0" otherwise. Thus, column j describes the minimal state for item j.

Surmise Function Files: Surmise function files are available only in spreadsheet formats.

The matrix is here preceded by a row which denotes the item for which the respective state (i.e. row) is a clause.

Example for an SRBT file

```
#SRBT v2.0 structure ASCII
3
5
# Some comment or item information
000
100
110
101
111
```

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Doignon, J.-P. & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies, 23,* 175–196.

Doignon, J.-P. & Falmagne, J.-C. (1999). *Knowledge Spaces.* Springer Verlag, Berlin.

Hockemeyer, C. (2001). KST Tools User Manual (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf.

`read_kbase` *Read base file*

Description

Read a base from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kbase(filename,
           format = "auto",
           as.letters = TRUE,
           header = TRUE,
           sep = ',',
           enforce = TRUE)
```

Arguments

`filename`	A character string specifying the name of the base file.
`format`	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
`as.letters`	logical, should the elements of the sets be letters or numbers?
`header`	For spreadsheet files only: does the file contain a header row?
`sep`	For CSV files only: character separating cells within a row.
`enforce`	Should we enforce that the result is really a basis, i.e. should we run kmbasis on it?

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package. The values `"CSV"`, `"ODS"`, and `"XLSX"` refer to the respective spreadsheet file formats. The value `"auto"` (default) requests an automatic detection of the format.

If `as.letters` is `TRUE` the item names are letters, otherwise numbers. If the data are read from a spreadsheet file (CSV, XLSX, or ODS) containing a header row, item names are taken from there. If they are read from a spreadsheet file without header, the reading function automatically assigns item names from `"V1"` to `"Vn"` where `n` is the number of items.

Value

A list with the following elements:

`matrix`	the read structure/data as binary matrix
`sets`	the read structure as object of class `kbase`

Note

In automatic format detection, the distinction between `"matrix"` and `"KST"` formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

`kbase`, `kmbasis`, kstIO-package

Examples

```
# Produce a base file
library(kstIO)
d <- getwd()
setwd(tempdir())
write_kbase(phsg$basis, "phsg.bas", "KST") # (Old) KST format
# Read file
read_kbase("phsg.bas") # Automatic format detection
read_kbase("phsg.bas", "KST") # Explicit format specification
setwd(d)
```

`read_kdata` *Read a response patterns file*

Description

Read a set of response patterns from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kdata(filename,
           format = "auto",
           as.letters = TRUE,
           header = TRUE,
           sep = ',')
```

Arguments

filename	A character string specifying the name of the data file.
format	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
as.letters	logical, should the elements of the sets be letters or numbers?
header	For spreadsheet files only: does the file contain a header row?
sep	For CSV files only: character separating cells within a row.

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package. The value "auto" (default) requests an automatic detection of the format by thje read_XXX function.

If as.letters is TRUE the item names are letters, otherwise numbers. If the data are read from a spreadsheet file (CSV, XLSX, or ODS) containing a header row, item names are taken from there. If they are read from a spreadsheet file without header, the reading function automatically assigns item names from "V1" to "Vn" where n is the number of items.

Value

A binary matrix with the response patterns.

Note

In automatic format detection, the distinction between "matrix" and "KST" formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstIO-package

Examples

```
# Produce a data file
d <- getwd()
setwd(tempdir())
space <- kmunionclosure(phsg$basis)
data <- kmsimulate(space, 10, 0.1, 0.05)
write_kdata(data, "phsg.ods", "ODS") # matrix format (without any headers)
# Read file; only the first ten rows of the matrix are printed by default
read_kdata("phsg.ods") # Automatic format detection (default)
data <- read_kdata("phsg.ods", "ODS") # Explicit format specification
data
dim(data)
setwd(d)
```

`read_kfamset` *Read a family of sets from file*

Description

Read a family of sets from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kfamset(filename,
             format = "auto",
             as.letters = TRUE,
             header = TRUE,
             sep = ',',
             enforce = TRUE)
```

Arguments

`filename`	A character string specifying the name of the space file.
`format`	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
`as.letters`	logical, should the elements of the sets be letters or numbers?
`header`	Do we have a header row in spreadsheet format?
`sep`	Cell separator for CSV filkes
`enforce`	Shall we enforce famset properties, i.e. uniqueness of rows?

Details

The `format` values `"SRBT"`, `"KST"`, and `"matrix"` refer to the different generations of file formats described in kstIO-package. The value `"auto"` (default) requests an automatic detection of the format by thje `read_XXX` function.

`read_kfamset()` reads any knowledge space file (space, structure, basis) and ignores any file type infor in SRBT file headers.

If `as.letters` is `TRUE` the elements of the sets are letters, otherwise numbers.

Value

A list with the following elements:

`matrix`	the read structure/data as binary matrix
`sets`	the read structure as `kspace`.

Note

In automatic format detection, the distinction between `"matrix"` and `"KST"` formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

There exists no `write_kfamset` function on purpose.

Author(s)

Cord Hockemeyer `<cord.hockemeyer@uni-graz.at>`

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

space_property, kstIO-package

Examples

```
# Produce a space file
d <- getwd()
setwd(tempdir())
write_kfamset(phsg$basis, "phsg.fs") # Write in (default) SRBT format
# Read file
read_kfamset("phsg.fs") # Automatic format detection (default)
read_kfamset("phsg.fs", "SRBT") # Explicit format specification
setwd(d)
```

read_kspace *Read a knowledge space file*

Description

Read a knowledge space from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kspace(filename,
            format = "auto",
          as.letters = TRUE,
          header = TRUE,
          sep = ',',
          enforce = TRUE)
```

Arguments

filename	A character string specifying the name of the space file.
format	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", or "auto" (default).
as.letters	logical, should the elements of the sets be letters or numbers?
header	For spreadsheet files only: does the file contain a header row [default TRUE]?
sep	Cell separator for CSVC files
enforce	Do we enforce knowledge space properties, i.e. do we run kmunionclosure?

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package. The value "auto" (default) requests an automatic detection of the format by thje read_XXX function.

If as.letters is TRUE the elements of the sets are letters, otherwise numbers.

If close is TRUE, a closure under union is computed to ensure that the returned knowledge space really is one.

Value

A list with the following elements:

matrix	the read structure/data as binary matrix
sets	the read structure as kspace.

Note

In automatic format detection, the distinction between "matrix" and "KST" formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

space_property, kstIO-package

Examples

```
# Produce a space file
d <- getwd()
setwd(tempdir())
ksp <- kmunionclosure(phsg$basis)
write_kspace(ksp, "phsg.spc", format="SRBT") # Write in SRBT format
# Read file
read_kspace("phsg.spc") # Automatic format detection (default)
read_kspace("phsg.spc", "SRBT") # Explicit format specification
setwd(d)
```

---

read_kstructure *Read a knowledge structure file*

Description

Read a knowledge structure from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_kstructure(filename,
                format = "auto",
                as.letters = TRUE,
                header = TRUE,
                sep = ',',
                enforce = TRUE
               )
```

Arguments

filename	A character string specifying the name of the structure file.
format	Specification of the files format. Can be "SRBT", "KST", "matrix", "CSV", "ODS", "XLSX", or "auto" (default).
as.letters	logical, should the elements of the sets be letters or numbers?
header	For spreadsheet format: Does the file include a header?
sep	Cell separator for CSV files.
enforce	Do we enforce that the resulting structure is a knowldge structure or do we trust the file's correctness?

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package. "CSV", "ODS", and "XLSX" refer to the repsective spreadsheet formats. The value "auto" (default) requests an automatic detection of the format by thje read_XXX function.

If as.letters is TRUE the elements of the sets are letters, otherwise numbers.

Value

A list with the following elements:

matrix	the read structure/data as binary matrix
sets	the read structure as object of class kstructure.

Note

In automatic format detection, the distinction between "matrix" and "KST" formats work somewhat heuristic. In other words, in rare cases the automatic detection might give the wrong result.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstructure, kstIO-package

Examples

```
# Produce a structure file
d <- getwd()
setwd(tempdir())
kst <- kmstructure(phsg$basis)
write_kstructure(kst, "phsg.struct") # Write in (default) SRBT format
write_kstructure(kst, "phsg_struct.ods") # write in ODS format
# Read file
read_kstructure("phsg.struct") # Automatic format detection by heuristics (default)
read_kstructure("phsg_struct.ods") # Automatic format detection by filename extension
setwd(d)
```

---

read_surmisefunction *Read surmise function file*

Description

Read a surmise function from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_surmisefunction(filename)
```

Arguments

filename	A character string specifying the name of the surmise function file.

Details

Surmise function files exist (so far) only in spreadsheet format. The concrete file type is derived from the filname extension.

Value

A data frame containing the surmise function.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

See Also

kstIO-package

Examples

```
d <- getwd()
setwd(tempdir())
sf <- kmsurmisefunction(phsg$basis)
write_surmisefunction(sf, "phsg_sf.xlsx")
read_surmisefunction("phsg_sf.xlsx")
setwd(d)
```

`read_surmiserelation` *Read surmise relation file*

Description

Read a surmise relation from a file. The file formats are described in the kstIO-package information page.

Usage

```
read_surmiserelation(filename,
                     format = "auto",
                     header = TRUE,
                     as.letters = TRUE,
                     sep = ',')
```

Arguments

`filename`	A character string specifying the name of the base file.
`format`	Specification of the files format. Can be "SRBT", "matrix", "CSV", or "auto" (default).
`header`	Whether spreadsheet file contains header row.
`as.letters`	logical, should the elements of the sets be letters or numbers? Defaults to TRUE.
`sep`	Cell separator for CSV files.

Details

The `format` values `"SRBT"` and `"matrix"` refer to the different generations of file formats described in kstIO-package. The value `"auto"` (default) requests an automatic detection of the format by the `read_XXX` function.

If `as.letters` is `TRUE` (default), the elements of the sets are letters, otherwise numbers.

The relation is always closed under reflexivity and transitivity.

Value

The incidence matrix of the surmise relation.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). SRbT Tools User Manual. https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstIO-package

Examples

```
# Produce a relation file
d <- getwd()
setwd(tempdir())
r <- kmsurmiserelation(phsg$basis)
write_surmiserelation(r, "phsg.rel", "SRBT") # SRBT format
# Read file
read_surmiserelation("phsg.rel") # Automatic format detection
read_surmiserelation("phsg.rel", "SRBT") # Explicit format specification
setwd(d)
```

`write_kbase` *Write a base file*

Description

Write a base to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kbase(x,
            filename,
            format = NULL,
            sep = ',',
            enforce = TRUE)
```

Arguments

`x`	The data to be written, either a binary matrix or an object of `kbase` class.
`filename`	A character string specifying the name of the base file.
`format`	Specification of the files format. Can be "SRBT", "KST", "CSV", "matrix" or sprreadsheet formats.
`sep`	Cell separator for CSV files.
`enforce`	Enforce basis porperties? [Default is TRUE]

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kbase, kstIO-package

Examples

```
d <- getwd()
setwd(tempdir())
# Write base to file
write_kbase(phsg$basis, "phsg.bas") # Write in SRBT format
write_kbase(phsg$basis, "phsg.xlsx") # XLSX format
setwd(d)
```

---

write_kdata *Write a knowledge space theory file*

Description

Write a data set to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kdata(x,
            filename,
            format = NULL,
            sep = ',')
```

Arguments

x	The data to be written, as a binary matrix.
filename	A character string specifying the name of the data file.
format	Specification of the files format. Can be "KST" or spreadsheet format.
]	
sep	Cell separator for CSV format.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstIO-package

Examples

```
d <- getwd()
setwd(tempdir())
space <- kmunionclosure(phsg$basis)
df <- kmsimulate(space, 100, 0.1, 0.05)
# Write data to file
write_kdata(df, "phsg.dat") # Write in KST format
write_kdata(df, "phsg_dat.xlsx") # xlsx format
setwd(d)
```

---

write_kfamset *Write a family of sets to a file*

Description

Write a family of sets to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kfamset(x,
              filename,
              format = NULL,
              sep = ',',
              enforce = TRUE)
```

Arguments

x	The data to be written, either a binary matrix as kmfamset object or an object of kfamset class.
filename	A character string specifying the name of the famset file.

format	Specification of the files format. Can be "KST", matrix, "CSV", "ODS", or "XLSX".
sep	Cell eparator for CSV files.
enforce	Enforce Famset properties? [Default is TRUE]

Details

The format values refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
d <- getwd()
setwd(tempdir())
# Write famset to file
write_kfamset(phsg$basis, "phsg.fs") # Write in KST format
write_kfamset(phsg$basis, "phsg.ods") # Write the matrix in ODS format
setwd(d)
```

---

write_kspace *Write a knowledge space file*

Description

Write a knowledge space to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kspace(x,
             filename,
             format = NULL,
             sep = ',',
             enforce = TRUE)
```

Arguments

x	The data to be written, either a binary matrix or an object of kspace class.
filename	A character string specifying the name of the base file.
format	Specification of the files format. Can be "SRBT" (default), "KST", "CSV", or "matrix".
sep	Cell eparator for CSV files.
enforce	ENforce space properties? [Default is true]

Details

The format values "SRBT", "KST", and "matrix" refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

space_property, kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
d <- getwd()
setwd(tempdir())
ksp <- kmunionclosure(phsg$basis)
# Write space to file
write_kspace(ksp, "phsg.spc", format="KST") # Write in KST format
write_kspace(ksp, "phsg.xlsx") # Write the matrix XLSX format
setwd(d)
```

`write_kstructure` *Write a knowledge structure file*

Description

Write a knowledge structure to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_kstructure(x,
                 filename,
                 format = NULL,
                 sep=',',
                 enforce = TRUE)
```

Arguments

`x`	The data to be written, either a binary matrix or an object of `kstructure` class.
`filename`	A character string specifying the name of the base file.
`format`	Specification of the files format. Can be "SRBT" (default), "KST", "matrix", or spreadsheet formats. Default is NULL
`sep`	Cell separator for CSV files - otherwise ignored.
`enforce`	Enforce structure properties? [Default is TRUE]

Details

The `format` values refer to the different generations of file formats described in kstIO-package.

Author(s)

Cord Hockemeyer `<cord.hockemeyer@uni-graz.at>`

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

`kstructure`, kstIO-package

Examples

```
# Obtain data to write from the 'pks' package
d <- getwd()
setwd(tempdir())
kst <-kmunionclosure(phsg$basis)
# Write structure to file
write_kstructure(kst, "phsg.struct") # Write in (default) SRBT format
# Write the matrix in CSV format
write_kstructure(kst, "phsg.csv")
setwd(d)
```

`write_surmisefunction` *Write a surmise function file*

Description

Write a surmise function to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_surmisefunction(x, filename, sep=',')
```

Arguments

`x`	The data to be written, either a quadratic binary matrix or an object of `relation` class.
`filename`	A character string specifying the name of the base file.
`sep`	Cell separator for CSAV files.

Details

Surmise function can (so far) be stored only in spreadsheet formats. The concrete file type (CSV, ODS, or XLSX) is determined from the filename extension.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

See Also

kstIO-package

`write_surmiserelation` *Write a (surmise) relation file*

Description

Write a surmise relation to a file. The file formats are described in the kstIO-package information page.

Usage

```
write_surmiserelation(x, filename, format = NULL, sep=',')
```

Arguments

`x`	The data to be written, either a quadratic binary matrix or an object of `relation` class.
`filename`	A character string specifying the name of the relation file.
`format`	Specification of the files format. Can be "SRBT", "matrix", CSV", "ODS", or "XLSX". If `format` is NULL, the format is taken from the filname extension (if ".csv", ".ods", or ".xlsx"), otherwise it is "SRBT".
`sep`	Cell separator for CSV files.

Details

The `format` values `"SRBT"` and `"matrix"` refer to the different generations of file formats described in kstIO-package. The other formats refer to the respective spreadsheet formats.

Author(s)

Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

References

Hockemeyer, C. (2001). *KST Tools User Manual* (2nd ed.). https://resources.cord-hockemeyer.info/techreports/KST-Tools_TechRep_FWF01.pdf.

Poetzi, S. & Wesiak, G. (2001). *SRbT Tools User Manual.* https://resources.cord-hockemeyer.info/techreports/SRBT-Tools_TechRep_FWF01.pdf

See Also

kstIO-package

Examples

```
d <- getwd()
setwd(tempdir())
r <- kmsurmiserelation(phsg$basis)
# Write surmise relation to file
write_surmiserelation(r, "phsg.bas") # Write in (default) SRBT format
write_surmiserelation(r, "phsg.ods") # ODS format
setwd(d)
```

Index