DirectedHyperedge-class

Class DirectedHyperedge

Description

This class represents directed hyperedges in a Hypergraph-class. A directed hyperedge consists of two disjoint sets of nodes, those in the tail and those in the head of the hyperedge. Directed hyperedges are sometimes called hyperarcs.

Objects from the Class

Objects can be created by calls of the form `new("DirectedHyperedge", head, tail, label)`. You can also use the convenience function `DirectedHyperedge`.

Slots

- `tail`: Character vector of nodes in the tail of the hyperedge
- `head`: Character vector of nodes in the head of the hyperedge
- `label`: Character string describing the directed hyperedge

Extends

Class "Hyperedge", directly.

Methods

- `head signature(.Object = "DirectedHyperedge")`: Return a vector containing the nodes in the head of the hyperedge
- `tail signature(.Object = "DirectedHyperedge")`: Return a vector containing the nodes in the tail of the hyperedge
- `initialize signature(.Object = "DirectedHyperedge")`: Create a new instance.
- `nodes signature(object = "DirectedHyperedge")`: Return a vector containing all nodes present in the hyperedge.
- `show signature(object = "DirectedHyperedge")`: Print me
- `toUndirected signature(.Object = "DirectedHyperedge")`: Return a Hyperedge-class object that results from coercing to an undirected hyperedge.
Author(s)
Seth Falcon

See Also
DirectedHyperedge Hyperedge Hyperedge-class Hypergraph-class

Examples
head <- LETTERS[1:4]
tail <- LETTERS[19:21]
label <- "Directed hyperedge"
dhe <- new("DirectedHyperedge", head=head, tail=tail, label=label)

Description
A convenience constructor for DirectedHyperedge-class objects

Usage
DirectedHyperedge(head, tail, label = "")

Arguments
head Character vector of nodes that are part of the head of the hyperedge
tail Character vector of nodes that part of the tail of the hyperedge
label A character string describing the directed hyperedge

Value
An object of class DirectedHyperedge-class

Author(s)
Seth Falcon

See Also
DirectedHyperedge-class Hyperedge-class Hypergraph-class
A Hyperedge object represents a hyperedge in a hypergraph, that is, a subset of the nodes of a hypergraph.

Objects from the Class

Objects can be created by calls of the form `new("Hyperedge", nodes, label)`.
You can also use the convenience function `Hyperedge` to create instances.
This is especially useful for creating a list of `Hyperedge` instances using `lapply`.

Slots

- **head**: A vector of mode "character" containing the node labels that are a part of the hyperedge
- **label**: An arbitrary "character" string describing this hyperedge

Methods

- **initialize** signature(.Object = "Hyperedge"):
  Create an instance
- **label** signature(object = "Hyperedge"):
  Return the value of the label slot
- **label<-** signature(object = "Hyperedge", value = "character"):
  Set the label slot.
- **nodes** signature(object = "Hyperedge"):
  Return a vector containing the nodes in the hyperedge
- **show** signature(object = "Hyperedge"):
  Print a textual summary of the hyperedge

Author(s)

Seth Falcon

See Also

Hyperedge Hypergraph-class DirectedHyperedge-class

Examples

```r
nodes <- LETTERS[1:4]
label <- "Simple hyperedge"
## Use the convenience constructor
he <- Hyperedge(nodes, label)
```
Hyperedge

Constructor for Hyperedge objects

Description
A convenience constructor for Hyperedge-class objects

Usage
Hyperedge(nodes, label = "")

Arguments

nodes Character vector of nodes that are part of the hyperedge
label A character string describing the hyperedge

Value
An object of class Hyperedge-class

Author(s)
Seth Falcon

See Also
Hyperedge-class Hypergraph-class

Hypergraph-class
Class Hypergraph

Description
A hypergraph consists of a set of nodes and a set of hyperedges. Each hyperedge is a subset of the node set. This class provides a representation of a hypergraph that is (hopefully) useful for computing.

Objects from the Class
Objects can be created by calls of the form new("Hypergraph", nodes, hyperedges). You can also use the convenience function Hypergraph. The nodes argument should be a character vector of distinct labels representing the nodes of the hypergraph. The hyperedges argument must be a list of Hyperedge-class objects.

Slots

nodes: A "character" vector specifying the nodes
hyperedges: A "list" of Hyperedge-class objects
Hypergraph

Methods

- **hyperedges** signature(.Object = "Hypergraph"): Return the list of Hyperedge objects
- **hyperedgeLabels** signature(.Object = "Hypergraph"): Return a character vector of labels for the Hyperedge objects in the hypergraph.
- **inciMat** signature(.Object = "Hypergraph"): Return the incidence matrix representation of this hypergraph
- **initialize** signature(.Object = "Hypergraph"): Create an instance
- **nodes** signature(object = "Hypergraph"): Return the vector of nodes (character vector)
- **numNodes** signature(object = "Hypergraph"): Return the number of nodes in the hypergraph
- **toGraphNEL** signature(.Object = "Hypergraph"): Return the graphNEL representation of the hypergraph (a bipartite graph)

Author(s)

Seth Falcon

See Also

Hyperedge-class DirectedHyperedge-class graphNEL-class

Examples

```r
nodes <- LETTERS[1:4]
hEdges <- lapply(list("A", LETTERS[1:2], LETTERS[3:4]), "Hyperedge")
hg <- new("Hypergraph", nodes=nodes, hyperedges=hEdges)
```

Hypergraph

Constructor for Hypergraph objects

Description

A convenience constructor for link(Hypergraph-class) objects

Usage

Hypergraph(nodes, hyperedges)

Arguments

- **nodes** A vector of nodes (character)
- **hyperedges** A list of Hyperedge-class objects

Value

An object of class Hypergraph-class
kCoresHypergraph

Description
Find all the k-cores in a hypergraph

Usage
kCoresHypergraph(hg)

Arguments
hg an instance of the Hypergraph class

Details
A k-core in a hypergraph is a maximal subhypergraph where (a) no hyperedge is contained in another, and (b) each node is adjacent to at least k hyperedges in the subgraph.
The implementation is based on the algorithm by E. Ramadan, A. Tarafdar, A. Pothen, 2004.

Value
A vector of the core numbers for all the nodes in g.

Author(s)
Li Long <li.long@isb-sib.ch>

References

Examples
# to turn the snacoreex.gxl (from RBGL package) graph to a hypergraph
# this is a rough example
kc_hg_e <- lapply(kc_hg_e, "Hyperedge")
kc_hg_he <- lapply(kc_hg_e, "Hyperedge")
kc_hg <- new("Hypergraph", nodes=kc_hg_n, hyperedges=kc_hg_he)
kCoresHypergraph(kc_hg)
12hel

Create lists of Hyperedge objects

Description
Conveniently create lists of Hyperedge-class instances.

Usage
l2hel(e)

Arguments
e
A list of character vectors. Each element of the list represents a hyperedge and the character vector value specifies the nodes of the hypergraph that are part of the hyperedge. The names of the list elements, if found, will be used as the label for the corresponding Hyperedge object.

Value
A list of Hyperedge-class objects. If the list e did not have names, the labels of the Hyperedges will be set to its index in the list coerced to character.

Author(s)
Seth Falcon

See Also
Hyperedge-class Hypergraph-class

Examples
edges <- list("e1"="A", "e2"=c("A", "B"), "e3"=c("C", "D"))
hEdgeList <- l2hel(edges)

vCoverHypergraph
Approximate minimum weight vertex cover in a hypergraph

Description
Approximate minimum weight vertex cover in a hypergraph

Usage
vCoverHypergraph(hg, vW=rep(1, numNodes(hg)))

Arguments
hg
an instance of the Hypergraph class
vW
vertex weights
Details

Hypergraph $g$ has non-negative weights on its vertices. The minimum weight vertex cover problem is to find a subset of vertices $C$ such that $C$ includes at least one vertex from each hyperedge and the sum of the weights of the vertices in $C$ is minimum. This problem is NP-hard.

We implement the greedy algorithm to approximate near-optimal solution, proposed by E. Ramadan, A. Tarafdar, A. Pothen, 2004.

Value

A list of vertices from hypergraph $g$.

Author(s)

Li Long <li.long@isb-sib.ch>

References


Examples

```r
# to turn the snacorex.gxl graph (from RBGL package) to a hypergraph
# this is a rough example
kc_hg_e <- list(c("A", "C"), c("B", "C"), c("C", "E"), c("C", "F"), c("E", "D"), c("E", "F"), c("D", "G"), c("D", "H"), ...
kc_hg_he <- lapply(kc_hg_e, "Hyperedge")
kc_hg <- new("Hypergraph", nodes=kc_hg_n, hyperedges=kc_hg_he)

vCoverHypergraph(kc_hg)
```
Index

*Topic classes*
  DirectedHyperedge, 2
  DirectedHyperedge-class, 1
  Hyperedge, 4
  Hyperedge-class, 3
  Hypergraph, 5
  Hypergraph-class, 4
  l2hel, 7

*Topic models*
  kCoresHypergraph, 6
  vCoverHypergraph, 7

DirectedHyperedge, 1, 2, 2
DirectedHyperedge-class, 2, 3, 5, 6
DirectedHyperedge-class, 1

graphNEL-class, 5

head(DirectedHyperedge-class), 1
head, DirectedHyperedge-method
  (DirectedHyperedge-class), 1

Hyperedge, 2, 3, 4
Hyperedge-class, 2, 4–7
Hyperedge-class, 1, 3, 4
hyperedgeLabels
  (Hypergraph-class), 4
hyperedgeLabels, Hypergraph-method
  (Hypergraph-class), 4
hyperedges (Hypergraph-class), 4
hyperedges, Hypergraph-method
  (Hypergraph-class), 4

Hypergraph, 5
Hypergraph-class, 1–7
Hypergraph-class, 4

inciMat (Hypergraph-class), 4
inciMat, Hypergraph-method
  (Hypergraph-class), 4
initialize, DirectedHyperedge-method
  (DirectedHyperedge-class), 1
initialize, Hyperedge-method
  (Hyperedge-class), 3

kCoresHypergraph, 6
l2hel, 7
label (Hyperedge-class), 3
label, Hyperedge-method
  (Hyperedge-class), 3
label<-(Hyperedge-class), 3
label<-, Hyperedge, character-method
  (Hyperedge-class), 3
lapply, 3

nodes, DirectedHyperedge-method
  (DirectedHyperedge-class), 1
nodes, Hyperedge-method
  (Hyperedge-class), 3
nodes, Hypergraph-method
  (Hypergraph-class), 4
numNodes, Hypergraph-method
  (Hypergraph-class), 4

show, DirectedHyperedge-method
  (DirectedHyperedge-class), 1
show, Hyperedge-method
  (Hyperedge-class), 3

tail (DirectedHyperedge-class), 1
tail, DirectedHyperedge-method
  (DirectedHyperedge-class), 1
toGraphNEL (Hypergraph-class), 4
toGraphNEL, Hypergraph-method
  (Hypergraph-class), 4
toUndirected
  (DirectedHyperedge-class), 1
toUndirected, DirectedHyperedge-method
  (DirectedHyperedge-class), 1

vCoverHypergraph, 7