

Package ‘orbweaver’

June 13, 2024

Title Fast and Efficient Graph Data Structures

Version 0.10.2

Description Empower your data analysis with 'orbweaver', an R package designed for effortless construction and analysis of graph data structures. With 'orbweaver', you can seamlessly build and manipulate graph structures, leveraging its high-performance methods for filtering, joining, and mutating data within the R environment. Drawing inspiration from the efficiency of the 'data.table' package, 'orbweaver' ensures that mutations and changes to the graph are performed in place, streamlining your workflow for optimal productivity.

URL <https://github.com/ixpantia/orbweaver-r>

BugReports <https://github.com/ixpantia/orbweaver-r/issues>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Config/rextendr/version 0.3.1.9000

SystemRequirements Cargo (Rust's package manager), rustc

Imports glue, rlang

Suggests testthat (>= 3.0.0), tibble

Config/testthat/edition 3

NeedsCompilation yes

Author ixpantia, SRL [cph],
Andres Quintero [aut, cre],
The authors of the dependency Rust crates [ctb] (see inst/AUTHORS file
for details)

Maintainer Andres Quintero <andres@ixpantia.com>

Repository CRAN

Date/Publication 2024-06-13 21:50:02 UTC

Contents

add_edge	2
add_path	3
build_acyclic	3
build_directed	4
children	4
find_all_paths	5
find_path	5
get_all_leaves	6
get_all_roots	6
get_leaves_under	7
get_roots_over	7
graph_builder	8
graph_from_bin	8
graph_to_bin	9
has_children	9
has_parents	10
least_common_parents	10
nodes	11
parents	11
populate_edges	12

Index	13
--------------	-----------

add_edge	<i>Add an edge to a graph builder</i>
----------	---------------------------------------

Description

Adds an edge from one node to another in a directed graph builder.

Usage

```
add_edge(graph_builder, from, to)
```

Arguments

graph_builder	A graph builder object
from	The from node.
to	The to node.

Value

The updated graph builder object

add_path	<i>Add a path to a graph</i>
----------	------------------------------

Description

Adds all of the edges that make up the given path to the graph.

Usage

```
add_path(graph_builder, path)
```

Arguments

graph_builder	A graph builder_object
path	A character vector that describes the path

Value

The updated graph builder object

build_acyclic	<i>Build a DirectedAcyclicGraph from a builder</i>
---------------	----------------------------------------------------

Description

Builds a graph builder into a new DirectedAcyclicGraph object.

NOTE: This will consume the builder. It will leave an empty builder in its place.

Usage

```
build_acyclic(graph_builder)
```

Arguments

graph_builder	A graph builder object
---------------	------------------------

Value

A DirectedAcyclicGraph Object

build_directed	<i>Build a DirectedGraph from a builder</i>
----------------	---------------------------------------------

Description

Builds a graph builder into a new DirectedGraph object.

NOTE: This will consume the builder. It will leave an empty builder in its place.

Usage

```
build_directed(graph_builder)
```

Arguments

graph_builder A graph builder object

Value

A DirectedGraph Object

children	<i>Get the children on a node</i>
----------	-----------------------------------

Description

Get a list of the node ids of the children of the provided node.

Usage

```
children(graph, nodes)
```

Arguments

graph A graph object
nodes A character vector of nodes to find children for

Value

A character vector

find_all_paths	<i>Find all paths between two nodes</i>
----------------	-----------------------------------------

Description

Find all the paths between two nodes in a graph.

Not all graphs support this function. Currently only DirectedAcyclicGraph supports this.

Usage

```
find_all_paths(graph, from, to)
```

Arguments

graph	A graph object
from	The starting node of the path
to	The ending node of the path

Value

A list of character vectors

find_path	<i>Find a path between two nodes</i>
-----------	--------------------------------------

Description

Finds a path between two nodes in a graph.

Different types of graphs use different algorithms to find the paths. a DirectedGraph uses breadth-first search while an DirectedAcyclicGraph uses topological sort.

The path is represented as a character vector with the node ids of the nodes that make up the path.

Usage

```
find_path(graph, from, to)
```

Arguments

graph	A graph object
from	The starting node of the path
to	The ending node of the path

Value

A character vector

get_all_leaves *Get all the leaf nodes of a graph*

Description

Retrieves the nodes in a graph that have no children

Usage

```
get_all_leaves(graph, ...)
```

Arguments

graph	A graph object
...	Unused

Value

A character vector of nodes

get_all_roots *Get the all the root nodes of a graph*

Description

Retrieves the nodes in a graph that have no parents

Usage

```
get_all_roots(graph, ...)
```

Arguments

graph	A graph object
...	Unused

Value

A character vector of nodes

`get_leaves_under` *Get the leaf nodes of a graph under some nodes*

Description

Retrieves the nodes in a graph that have no children under a certain node or group of nodes

Usage

`get_leaves_under(graph, nodes)`

Arguments

`graph` A graph object
`nodes` A character vector of nodes to find leaves for

Value

A character vector of nodes

`get_roots_over` *Get the root nodes of a graph over some nodes*

Description

Retrieves the nodes in a graph that have no parents over a certain node or group of nodes

Usage

`get_roots_over(graph, nodes)`

Arguments

`graph` A graph object
`nodes` A character vector of nodes to find roots for

Value

A character vector of nodes

graph_builder	<i>A new builder for a graph based on the type</i>
---------------	----------------------------------------------------

Description

Object used to build graphs

Usage

```
graph_builder(type = "directed")
```

Arguments

type	The type of graph
------	-------------------

graph_from_bin	<i>Read the graph from a binary blob</i>
----------------	------------------------------------------

Description

Read the graph from a binary blob

Usage

```
graph_from_bin(path, bin, type = c("directed", "dag"))
```

Arguments

path	(Optional) Path to a file containing a graph binary
bin	(Optional) The raw binary of the graph
type	The type of graph the JSON represents

Value

A graph object

graph_to_bin	<i>Save the graph into a binary blob</i>
--------------	------------------------------------------

Description

Save the graph into a binary blob

Usage

```
graph_to_bin(graph, path)
```

Arguments

graph	A graph object
path	Path to a file to save the graph into

Value

Run for its side-effects

has_children	<i>Checks if a node in a graph has children</i>
--------------	-------------------------------------------------

Description

This function validates if the node has an edge pointing to any other node.

Usage

```
has_children(graph, nodes)
```

Arguments

graph	A graph object
nodes	A character vector of nodes to determine

Value

A logical vector with the same length as nodes

has_parents *Checks if a node in a graph has parents*

Description

This function validates if any edge points to the given node.

Usage

```
has_parents(graph, nodes)
```

Arguments

graph	A graph object
nodes	A character vector of nodes to determine

Value

A logical vector with the same length as nodes

least_common_parents *Find the least common parents in a graph*

Description

It finds the nodes that have no parents in the given set.

Usage

```
least_common_parents(graph, selected)
```

Arguments

graph	A graph object
selected	A character vector of node ids

Value

A character vector of node ids

nodes	<i>Get the nodes in the graph</i>
-------	-----------------------------------

Description

Returns the unique nodes in the graph

Usage

```
nodes(graph, ...)
```

Arguments

graph	A directed or directed acyclic graph
...	Reserved for later use

Value

A character vector with the nodes

parents	<i>Get the parents on a node</i>
---------	----------------------------------

Description

Get a list of the node ids of the parents of the provided node.

Usage

```
parents(graph, nodes)
```

Arguments

graph	A graph object
nodes	A character vector of nodes to find parents for

Value

A character vector

populate_edges	<i>Populates the edges of a graph from a tibble</i>
----------------	-----------------------------------------------------

Description

Adds a set of edges from a tibble to a graph

Usage

```
populate_edges(graph_builder, edges_df, parent_col, child_col)
```

Arguments

graph_builder	A graph builder object
edges_df	A tibble with a parent and child variable
parent_col	The name of the column containing the parents
child_col	The name of the column containing the children

Value

The updated graph builder object

Index

add_edge, 2
add_path, 3

build_acyclic, 3
build_directed, 4

children, 4

find_all_paths, 5
find_path, 5

get_all_leaves, 6
get_all_roots, 6
get_leaves_under, 7
get_roots_over, 7
graph_builder, 8
graph_from_bin, 8
graph_to_bin, 9

has_children, 9
has_parents, 10

least_common_parents, 10

nodes, 11

parents, 11
populate_edges, 12