Package 'toposort'

March 8, 2023

Title Topological Sorting Algorithms

Version 1.0.0

Description Flexible and ergonomic topological sorting implementation for R. Supports a variety of input data encoding (lists of edges or adjacency matrices, graphs edge direction), stable sort variants as well as cycle detection with detailed diagnosis.

License MIT + file LICENSE

URL https://github.com/tzakharko/toposort

BugReports https://github.com/tzakharko/toposort/issues

Imports glue, rlang, vctrs

Suggests cli, testthat (>= 3.0.0), tibble

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Taras Zakharko [aut, cre] (<https://orcid.org/0000-0001-7601-8424>)

Maintainer Taras Zakharko <taras.zakharko@gmail.com>

Repository CRAN

Date/Publication 2023-03-08 15:00:03 UTC

R topics documented:

topological_sort *Topological sorting*

Description

Topological sorting

Usage

```
topological_sort(x, ..., dependency_type, labels = vec_names(x))

stable_topological_sort(x, ..., dependency_type, labels = vec_names(x))
```

Arguments

x	The dependency graph. It must be either a list of integer vectors (where `x[[i]]` are the items `i` depends on) or as a square matrix of integer or logical flags (where `x[i,j] == 1` indicates that `i` depends on `j`). The graph can optionally have names (will be taken from row names for the matrix).
...	These dots are for future extensions and must be empty.
dependency_type	named string argument specifying how to interpret the graph. This must be either "precedes" (parent nodes in the graph come before their children) or "follows" (parent nodes in the graph follow their children). This can also be specified as an attribute of the same name on the graph input `x`
labels	optional named character vector of item labels. If provided, the sorted output will use these labels. The default labels are taken from the names of `x` (row names if it is a matrix), if any are provided. Set to `NULL` to suppress labels.

Details

The dependency structure can be encoded in a number of different ways for flexibility (see examples).

`stable_topological_sort()` guarantees stable sort order (items without mutual dependencies will be sorted in the order of occurrence). `topological_sort()` makes no such guarantees and might offer improved performance in future versions of the package.

An informative error is raised if cycles are detected in the dependency graph. The error condition has the class `toposort/cyclic_depenencies_error` and the element `cycles` of the condition will contain the list of detected cycles

Value

Items in their order of precedence (earlier items first). This is either an integer vector of item indices or a character vector of item labels (if labels were provided).

Examples

```
# the following examples show the different ways to encode the
# dependency structure of four items, where item 1 precedes items 2 and 3,
# item 2 precedes item 4, and item 3 precedes item 2

# list with items encoded by their precedence (i precedes all x[[i]])
x <- list(c(2L, 3L), 3L, 4L, integer())
topological_sort(x, dependency_type = "precedes")
stable_topological_sort(x, dependency_type = "precedes")

# list with items encoded by their antecedence (i follows all x[[i]]))
x <- list(integer(), c(1L, 3L), 1L, 2L)
topological_sort(x, dependency_type = "follows")
stable_topological_sort(x, dependency_type = "follows")

# matrix with items encoded by their precedence
x <- matrix(FALSE, ncol = 4, nrow = 4)
x[1L, c(2L, 3L)] <- TRUE
x[2L, 4L] <- TRUE
x[3L, 2L] <- TRUE
topological_sort(x, dependency_type = "precedes")
stable_topological_sort(x, dependency_type = "precedes")

# matrix with items encoded by their antecedence
x <- matrix(FALSE, ncol = 4, nrow = 4)
x[2L, c(1L, 3L)] <- TRUE
x[3L, 1L] <- TRUE
x[4L, 2L] <- TRUE
topological_sort(x, dependency_type = "follows")
stable_topological_sort(x, dependency_type = "follows")
```

Index