

Package ‘petrinetR’

July 23, 2025

Type Package

Title Building, Visualizing, Exporting and Replaying Petri Nets

Version 0.3.0

Date 2023-04-01

Description Functions for the construction of Petri Nets. Petri Nets can be replayed by firing enabled transitions.
Silent transitions will be hidden by the execution handler. Also includes functionalities for the visualization of Petri Nets and
export of Petri Nets to PNML (Petri Net Markup Language) files.

License GPL-3

Depends R(>= 3.0.0)

Imports dplyr, visNetwork, DiagrammeR, xml2, purrr, lifecycle

RoxygenNote 7.2.3

Encoding UTF-8

URL <https://bupar.net>, <https://github.com/bupaverse/petrinetR>

BugReports <https://github.com/bupaverse/petrinetR/issues>

NeedsCompilation no

Author Gert Janssenswillen [aut, cre],
Ivan Esin [ctb]

Maintainer Gert Janssenswillen <gert.janssenswillen@uhasselt.be>

Repository CRAN

Date/Publication 2023-04-03 13:20:02 UTC

Contents

create_marked_PN	2
create_PN	3
enabled	3
enabled_transition	4
execute	4

final_marking	5
flows	5
initial_marking	6
is_node	6
is_place	7
is_transition	7
marked_petrinet	8
marking	8
nodes	8
n_places	9
parsel_trace	10
parse_trace	10
part_of	11
petrinet	11
petrinetR	11
places	12
post_set	12
pre_set	13
read_PN	13
render_PN	14
transitions	14
visNetwork_from_PN	15

Index 16

create_marked_PN	<i>Create marked Petri Net</i>
------------------	--------------------------------

Description

Function to create a [marked_petrinet](#), consisting of a [petrinet](#), an initial marking, and a final marking.

Usage

```
create_marked_PN(PN, initial_marking, final_marking)
```

Arguments

PN [petrinet](#): Object of class [petrinet](#).
initial_marking [character](#): A vector with place ids representing the initial marking.
final_marking [character](#): A vector with place ids representing the final marking.

Value

A [marked_petrinet](#)

create_PN

Create Petri Net

Description

Function to create a [petrinet](#) by specifying places, transitions and flows.

Usage

```
create_PN(places, transitions, flows)
```

Arguments

places	data.frame or tibble of places, with columns id and label. Both columns should be characters.
transitions	data.frame or tibble of transitions, with columns id and label. Both columns should be characters.
flows	data.frame or tibble of flows, with columns named "from" and "to", referring to ids of places and transitions. Both columns should be characters.

Value

A [petrinet](#)

Examples

```
library(dplyr)
create_PN(tibble(id = "p1", label = "place_1"),
  tibble(id = "t1", label = "transition_1"),
  tibble(from = "t1", to = "p1"))
```

enabled

Enabled transitions

Description

List the enabled transitions in a marked Petri Net. Silent transitions, i.e. starting with "inv_" or "tau" are assumed to be able to fire silently, thereby possibly enabling other transitions.

Usage

```
enabled(PN)
```

Arguments

PN	A Petri Net
----	-------------

enabled_transition	<i>Enabled Transition</i>
--------------------	---------------------------

Description

Check if a transition is currently enabled

Usage

```
enabled_transition(PN, transition)
```

Arguments

PN	A Petri Net
transition	A Transition

execute	<i>Execute</i>
---------	----------------

Description

Executes (fire) an enabled transition and returns the Petri Net with the New marking. If the transition is enabled via the firing of silent transition (i.e. starting with "inv_" or "tau"), it will fire these first. If the transition is not enabled, it will return FALSE.

Usage

```
execute(PN, transition)
```

Arguments

PN	A Petri Net
transition	The transition to be fired

final_marking	<i>Final Marking</i>
---------------	----------------------

Description

Get the final marking of a [marked_petrinet](#)

Usage

final_marking(PN)

Arguments

PN A [marked_petrinet](#)

flows	<i>Flows</i>
-------	--------------

Description

Extracts the flows from a (marked) Petri Net

Usage

```
flows(PN)

## S3 method for class 'petrinet'
flows(PN)

## S3 method for class 'marked_petrinet'
flows(PN)
```

Arguments

PN [petrinet](#) or [marked_petrinet](#)

Value

A data.frame containing the flows of the petri net.

Methods (by class)

- flows(petrinet): Flow of petrinet
- flows(marked_petrinet): Flow of marked petrinet

initial_marking	<i>Initial Marking</i>
-----------------	------------------------

Description

Get the initial marking of a [marked_petrinet](#)

Usage

initial_marking(PN)

Arguments

PN A [marked_petrinet](#)

is_node	<i>Is node</i>
---------	----------------

Description

Check if a node is part of a petri net

Usage

is_node(node, PN)

Arguments

node [character](#) of length one: the node id to check.
PN [petrinet](#) or [marked_petrinet](#)

Value

logical that indicates whether node is a node in PN

*is_place**Is place*

Description

Check if a place is part of a petri net.

Usage

```
is_place(place, PN)
```

Arguments

place [character](#) of length one: the place id to check.

PN [petrinet](#) or [marked_petrinet](#)

Value

logical that indicates whether place is a place in PN

*is_transition**Is transition*

Description

Check if a transition is part of a petri net.

Usage

```
is_transition(transition, PN)
```

Arguments

transition [character](#) of length one: the transition id to check.

PN [petrinet](#) or [marked_petrinet](#)

Value

logical that indicates whether transition is a transition in PN

marked_petrinet	<i>Marked petrinet</i>
-----------------	------------------------

Description

Object consisting of a petrinet, initial marking, and final marking

marking	<i>Marking</i>
---------	----------------

Description

Get the current marking of a Petri Net

Usage

marking(PN)

Arguments

PN A Petri Net

nodes	<i>Get nodes from (marked) petrinet</i>
-------	---

Description

Get nodes from (marked) petrinet

Usage

nodes(PN)

Arguments

PN [petrinet](#) or [marked_petrinet](#)

n_places	Utils
----------	-------

Description

Several auxilliary functions for Petri Net objects.

Usage

n_places(PN)

n_transitions(PN)

n_flows(PN)

n_nodes(PN)

rename_transitions(PN, .f, ...)

rename_places(PN, .f, ...)

add_places(PN, places)

add_transitions(PN, transitions)

add_flows(PN, flows)

Arguments

PN A petri net

.f A function name to apply for renaming

... Additional arguments

places data.frame or tibble of places, with columns id and label. Both columns should be characters.

transitions data.frame or tibble of transitions, with columns id and label. Both columns should be characters.

flows data.frame or tibble of flows, with columns named "from" and "to", referring to ids of places and transitions. Both columns should be characters.

parsel_trace	<i>Parse (logical)</i>
--------------	------------------------

Description

Tests whether a sequence of transitions can be fired by a Petri Net. If so returns TRUE, otherwise FALSE.

Usage

parsel_trace(PN, trace)

Arguments

- | | |
|-------|--|
| PN | A Petri Net |
| trace | A sequence of transitions, stored in a vector. |

parse_trace	<i>Parse</i>
-------------	--------------

Description

Parses a sequence of transitions. If possible returns the Petri Net with the updated marking. Otherwise returns FALSE

Usage

parse_trace(PN, trace)

Arguments

- | | |
|-------|--|
| PN | A Petri Net |
| trace | A sequence of transitions, stored in a vector. |

part_of	<i>Part of</i>
---------	----------------

Description

Check if a node is part of a petri net

Usage

part_of(node, PN)

Arguments

node	A node
PN	A Petri Net

petrinet	<i>Ppetrinet</i>
----------	------------------

Description

Object consisting of places, transitions and flows that denote a petri net

petrinetR	<i>petrinetR - Building, visualizing, exporting and replaying Petri Nets</i>
-----------	--

Description

Functions for the construction of Petri Nets. Petri Nets can be replayed by firing enabled transitions. Silent transitions will be hidden by the execution handler. Also includes functionalities for the visualization of Petri Nets and export of Petri Nets to PNML-files.

places

Places

Description

Extracts the places from a Petri Net

Usage

```
places(PN)
```

```
## S3 method for class 'petrinet'
places(PN)
```

```
## S3 method for class 'marked_petrinet'
places(PN)
```

Arguments

PN [petrinet](#) or [marked_petrinet](#)

Methods (by class)

- `places(petrinet)`: Places of petrinet
- `places(marked_petrinet)`: Places of marked petrinet

post_set

Postset

Description

Get the postset of a transition or place in a Petri Net

Usage

```
post_set(PN, node)
```

Arguments

PN [petrinet](#) or [marked_petrinet](#)

node [character](#) of length one: the node id for which to get the postset.

pre_set	<i>Preset</i>
---------	---------------

Description

Get the preset of a transition or place in a Petri Net

Usage

```
pre_set(PN, node)
```

Arguments

PN	petrinet or marked_petrinet
node	character of length one: the node id for which to get the postset.

read_PN	<i>Read .PNML file</i>
---------	------------------------

Description

Read .PNML file

Usage

```
read_PN(file, add_final_marking = TRUE)
```

Arguments

file	Path to .pnml file
add_final_marking	logical (default: TRUE): add final marking. If TRUE, all places without outgoing flows are considered part of a single final marking. Overwrite with <code>set_final_marking()</code> if needed. If FALSE, final_marking is set to NULL

Value

A `marked_petrinet`

render_PN	<i>Render Petri Net</i>
-----------	-------------------------

Description

Visualize Petri Net with bipartite graph.

Usage

render_PN(PN)

Arguments

PN [petrinet](#) or [marked_petrinet](#)

transitions	<i>Transitions</i>
-------------	--------------------

Description

Extracts the transitions from a Petri Net

Usage

```
transitions(PN)

## S3 method for class 'petrinet'
transitions(PN)

## S3 method for class 'marked_petrinet'
transitions(PN)
```

Arguments

PN [petrinet](#) or [marked_petrinet](#)

Methods (by class)

- transitions(petrinet): Transitions of petrinet
- transitions(marked_petrinet): Transitions of marked petrinet

visNetwork_from_PN	<i>VisNetwork from PN</i>
--------------------	---------------------------

Description

Visualize a Petri Net with an interactive network

Usage

```
visNetwork_from_PN(PN)
```

Arguments

PN [petrinet](#) or [marked_petrinet](#)

Index

`add_flows (n_places)`, [9](#)
`add_places (n_places)`, [9](#)
`add_transitions (n_places)`, [9](#)

`character`, [2](#), [6](#), [7](#), [12](#), [13](#)
`create_marked_PN`, [2](#)
`create_PN`, [3](#)

`enabled`, [3](#)
`enabled_transition`, [4](#)
`execute`, [4](#)

`final_marking`, [5](#)
`flows`, [5](#)

`initial_marking`, [6](#)
`is_node`, [6](#)
`is_place`, [7](#)
`is_transition`, [7](#)

`logical`, [13](#)

`marked_petrinet`, [2](#), [5–8](#), [8](#), [12–15](#)
`marking`, [8](#)

`n_flows (n_places)`, [9](#)
`n_nodes (n_places)`, [9](#)
`n_places`, [9](#)
`n_transitions (n_places)`, [9](#)
`nodes`, [8](#)

`parse_trace`, [10](#)
`parsel_trace`, [10](#)
`part_of`, [11](#)
`petrinet`, [2](#), [3](#), [5–8](#), [11](#), [12–15](#)
`petrinetR`, [11](#)
`places`, [12](#)
`post_set`, [12](#)
`pre_set`, [13](#)

`read_PN`, [13](#)

`rename_places (n_places)`, [9](#)
`rename_transitions (n_places)`, [9](#)
`render_PN`, [14](#)

`transitions`, [14](#)

`visNetwork_from_PN`, [15](#)