

Package ‘Rdiagnosislist’

July 19, 2021

Title Manipulate SNOMED CT Diagnosis Lists

Version 0.2

Description Functions and methods for manipulating SNOMED CT concepts. The package contains functions for loading the SNOMED CT release into a convenient R environment, selecting SNOMED CT concepts using regular expressions, and navigating the SNOMED CT ontology. It provides the 'SNOMEDconcept' S3 class for a vector of SNOMED CT concepts (stored as 64-bit integers) and the 'SNOMEDcodelist' S3 class for a table of concepts IDs with descriptions. For more information about SNOMED CT visit <<https://www.snomed.org/>>.

License GPL-3

Imports data.table, bit64, methods

Suggests knitr, rmarkdown, testthat

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Author Anoop Shah [aut, cre] (<<https://orcid.org/0000-0002-8907-5724>>)

Maintainer Anoop Shah <anoop@doctors.org.uk>

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-07-19 08:40:10 UTC

R topics documented:

as.data.frame.SNOMEDconcept	2
attrConcept	3
c.SNOMEDconcept	4
createSNOMEDindices	5
description	5
expandSNOMED	6

getSNOMED	7
hasAttributes	7
inactiveIncluded	8
is.SNOMEDcodelist	9
is.SNOMEDconcept	9
loadSNOMED	10
parents	11
print.SNOMEDconcept	11
Rdiagnosislist	12
relatedConcepts	12
sampleSNOMED	13
semanticType	14
simplify	14
SNOMEDcodelist	15
SNOMED_CONCEPT	17
SNOMED_DESCRIPTION	18
SNOMED_RELATIONSHIP	19
union.SNOMEDconcept	20
unique.SNOMEDconcept	21

Index 23

as.data.frame.SNOMEDconcept

Returns the SNOMED CT concept IDs for a set of terms

Description

Carries out an exact or regular expression match to return the concept ID for a set of search terms, or converts a character, integer or integer64 vector to a SNOMEDconcept object.

Usage

```
## S3 method for class 'SNOMEDconcept'
as.data.frame(x, ...)
```

```
## S3 method for class 'SNOMEDconcept'
as.integer64(x)
```

```
SNOMEDconcept(
  x,
  active_only = TRUE,
  exact_match = TRUE,
  unique = TRUE,
  SNOMED = getSNOMED()
)
```

```
as.SNOMEDconcept(x, ...)
```

Arguments

x	character vector of terms to match, or character vector containing SNOMED CT concept IDs, or 64-bit integer vector containing SNOMED CT concept IDs
...	additional arguments to send to grepl if using regular expression matching
active_only	whether or not to include inactive concepts
exact_match	if TRUE, only an exact (case sensitive) match is performed. If FALSE, a regular expression match is performed.
unique	whether to include no more than one instance of each SNOMED CT concept
SNOMED	environment containing SNOMED dictionary. Defaults to an object named 'SNOMED' in the global environment

Value

a SNOMEDconcept object (vector of 64-bit integers) containing unique SNOMED CT concept IDs

See Also

Other SNOMEDconcept functions: [c.SNOMEDconcept\(\)](#), [is.SNOMEDconcept\(\)](#), [print.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#), [unique.SNOMEDconcept\(\)](#)

Other SNOMEDconcept functions: [c.SNOMEDconcept\(\)](#), [is.SNOMEDconcept\(\)](#), [print.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#), [unique.SNOMEDconcept\(\)](#)

Other SNOMEDconcept functions: [c.SNOMEDconcept\(\)](#), [is.SNOMEDconcept\(\)](#), [print.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#), [unique.SNOMEDconcept\(\)](#)

Examples

```
SNOMEDconcept('Heart failure', SNOMED = sampleSNOMED()) -> hf
is.SNOMEDconcept(hf)
SNOMEDconcept('900000000000003001')
as.SNOMEDconcept('900000000000003001')
```

attrConcept

Retrieve all attributes of a set of SNOMED CT concepts

Description

Returns the portion of the SNOMED CT relationship tables containing relationships where the given concepts are either the source or the destination.

Usage

```
attrConcept(
  conceptIds,
  SNOMED = getSNOMED(),
  tables = c("RELATIONSHIP", "STATEDRELATIONSHIP")
)
```

Arguments

conceptIds character or integer64 vector of SNOMED concept IDs
 SNOMED environment containing a SNOMED dictionary
 tables character vector of relationship tables to use

Value

a data.table with the following columns: sourceId (concept ID of source for relationship), destinationId (concept ID of source for relationship), typeId (concept ID of relationship type), typeName (description of relationship type)

Examples

```
SNOMED <- sampleSNOMED()

attrConcept(as.SNOMEDconcept('Heart failure'))
```

c.SNOMEDconcept *Concatenate vectors of SNOMED CT concepts*

Description

SNOMEDconcept is an S3 class for vectors of SNOMED concept IDs as 64-bit integers. This function concatenates two or more SNOMEDconcept vectors.

Usage

```
## S3 method for class 'SNOMEDconcept'
c(...)
```

Arguments

... SNOMEDconcept vectors

Value

concatenation of vectors

See Also

Other SNOMEDconcept functions: [as.data.frame.SNOMEDconcept\(\)](#), [is.SNOMEDconcept\(\)](#), [print.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#), [unique.SNOMEDconcept\(\)](#)

Examples

```
hf <- SNOMEDconcept('Heart failure', SNOMED = sampleSNOMED())
hf2 <- c(hf, hf)
```

createSNOMEDindices *Create indices for tables in a SNOMED environment*

Description

Creates relevant indices for fast searching of SNOMED CT tables

Usage

```
createSNOMEDindices(SNOMED)
```

Arguments

SNOMED environment containing data.table objects: CONCEPT, DESCRIPTION, RELATIONSHIP, STATEDRELATIONSHIP

Value

The environment with indices added to each table for fast searching

description *Obtain descriptions for a set of SNOMED CT terms*

Description

Returns the descriptions matching a set of concept IDs from a SNOMED dictionary

Usage

```
description(
  conceptIds,
  include_synonyms = FALSE,
  active_only = TRUE,
  SNOMED = getSNOMED()
)
```

Arguments

conceptIds character or integer64 vector

include_synonyms whether to return only the Fully Specified Name (default) or all synonyms

active_only whether to include only active descriptions

SNOMED environment containing SNOMED dictionary. Defaults to an object named 'SNOMED' in the global environment

Value

a data.table with the following columns: id, conceptId, type (only if include_synonyms = TRUE), term, active (only if active_only = FALSE)

Examples

```
hf <- SNOMEDconcept('Heart failure', SNOMED = sampleSNOMED())
description(hf, include_synonyms = FALSE, SNOMED = sampleSNOMED())
```

expandSNOMED	<i>Expand or contract a SNOMEDcodelist</i>
--------------	--

Description

SNOMEDcodelist is an S3 class for sets of SNOMED concepts. In the 'contracted' form, it may contain only parents and not child terms (to create a more succinct list). The 'Expanded' form contains all concepts.

Usage

```
expandSNOMED(x, SNOMED = getSNOMED())

contractSNOMED(x, SNOMED = getSNOMED())
```

Arguments

x	SNOMEDcodelist to expand or contract
SNOMED	environment containing a SNOMED dictionary

Value

An object of class 'SNOMEDcodelist' with attribute Expanded = TRUE

See Also

Other SNOMEDcodelist functions: [SNOMEDcodelist\(\)](#), [is.SNOMEDcodelist\(\)](#)
 Other SNOMEDcodelist functions: [SNOMEDcodelist\(\)](#), [is.SNOMEDcodelist\(\)](#)

Examples

```
SNOMED <- sampleSNOMED()

my_concepts <- SNOMEDconcept('Heart failure')
my_codelist <- SNOMEDcodelist(data.frame(conceptId = my_concepts,
  include_desc = TRUE))
expanded_codelist <- expandSNOMED(my_codelist)
contractSNOMED(expanded_codelist)
```

getSNOMED	<i>Retrieves SNOMED CT dictionary from the global environment</i>
-----------	---

Description

Returns an object named 'SNOMED' from the global environment. Returns an error if no such object exists, or if it is not an environment containing tables named CONCEPT, RELATIONSHIP, STATEDRELATIONSHIP and DESCRIPTION. There is no attempt to check that these tables are actually, if available. environment containing a selection of SNOMED CT terms, their relationships and descriptions which are provided with the package

Usage

```
getSNOMED()
```

Value

SNOMED environment from the global environment

Examples

```
SNOMED <- sampleSNOMED()
SNOMED2 <- getSNOMED()

# To display metadata for this SNOMED CT dictionary
SNOMED2$metadata
```

hasAttributes	<i>Whether SNOMED CT concepts have particular attributes</i>
---------------	--

Description

For each concept in the first list, whether it has the attribute in the second list. Returns a vector of Booleans.

Usage

```
hasAttributes(
  sourceIds,
  destinationIds,
  typeId = bit64::as.integer64("116680003"),
  SNOMED = getSNOMED(),
  tables = c("RELATIONSHIP", "STATEDRELATIONSHIP")
)
```

Arguments

sourceIds	character or integer64 vector of SNOMED concept IDs for children, recycled if necessary
destinationIds	character or integer64 vector of SNOMED concept IDs for parents, recycled if necessary
typeIds	character or integer64 vector of SNOMED concept IDs for relationship types, recycled if necessary. Defaults to 116680003 = 'Is a' (child/parent)
SNOMED	environment containing a SNOMED dictionary
tables	character vector of relationship tables to use

Value

a vector of Booleans stating whether the attribute exists

Examples

```
SNOMED <- sampleSNOMED()

hasAttributes(c('Heart failure', 'Acute heart failure'),
             c('Heart structure', 'Heart failure'),
             c('Finding site', 'Is a'))
```

<code>inactiveIncluded</code>	<i>Check if inactive terms are included in SNOMED CT dictionary</i>
-------------------------------	---

Description

Checks the `active_only` flag in the metadata of a SNOMED environment to determine whether inactive terms are included

Usage

```
inactiveIncluded(SNOMED = getSNOMED())
```

Arguments

SNOMED	environment containing SNOMED dictionary, defaults to an object named 'SNOMED' in the global environment
--------	--

Value

TRUE or FALSE (logical vector of length one)

Examples

```
# Create a TEST environment and load the sample dictionaries
TEST <- sampleSNOMED()
inactiveIncluded(TEST)
assign('metadata', list(active_only = TRUE), envir = TEST)
inactiveIncluded(TEST)
```

is.SNOMEDcodelist *Check if an object is a SNOMEDcodelist*

Description

SNOMEDcodelist is an S3 class for lists of SNOMED codes. This function checks whether the object has the class SNOMEDcodelist. It does not check whether it contains valid data.

Usage

```
is.SNOMEDcodelist(x)
```

Arguments

x object to check

Value

a logical vector of length one: TRUE or FALSE

See Also

Other SNOMEDcodelist functions: [SNOMEDcodelist\(\)](#), [expandSNOMED\(\)](#)

is.SNOMEDconcept *Check if an object is a SNOMEDconcept*

Description

SNOMEDconcept is an S3 class for vectors of SNOMED concept IDs as 64-bit integers. This function checks whether the object has the class SNOMEDconcept and is a vector of 64-bit integers.

Usage

```
is.SNOMEDconcept(x)
```

Arguments

x object to check

Value

a logical vector of length one: TRUE or FALSE

See Also

Other SNOMEDconcept functions: [as.data.frame.SNOMEDconcept\(\)](#), [c.SNOMEDconcept\(\)](#), [print.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#), [unique.SNOMEDconcept\(\)](#)

loadSNOMED	<i>Load SNOMED files from a folder(s) into R data.table objects</i>
------------	---

Description

Identifies relevant SNOMED files from a distribution and loads them into an R environment. Files from two folders (e.g. International and UK versions) can be loaded together and appended.

Usage

```
loadSNOMED(folders, active_only = TRUE)
```

Arguments

folders Vector of folder paths containing SNOMED CT files
 active_only Whether to limit to current (active) SNOMED CT terms

Value

An environment containing data.table objects: CONCEPT, DESCRIPTION, RELATIONSHIP, STAT-EDRELATIONSHIP

Examples

```
# Create a TEST environment and load the sample dictionaries
TEST <- sampleSNOMED()

# Export to temporary directory
for (table in c('Concept', 'Description', 'Relationship',
  'StatedRelationship')){
  write.table(get(toupper(table), envir = TEST), paste0(tempdir(),
    '/sct_', table, '_text.txt'), row.names = FALSE, sep = '\t', quote = FALSE)
}

# Try to import using the loadSNOMED function
TEST2 <- loadSNOMED(tempdir(), active_only = FALSE)

# Check that reimported SNOMED dictionary is the same as the original
all.equal(TEST$CONCEPT, TEST2$CONCEPT)
all.equal(TEST$DESCRIPTION, TEST2$DESCRIPTION)
all.equal(TEST$RELATIONSHIP, TEST2$RELATIONSHIP)
all.equal(TEST$STATEDRELATIONSHIP, TEST2$STATEDRELATIONSHIP)
```

parents	<i>Ancestors and descendants of SNOMED CT concepts</i>
---------	--

Description

Returns concepts with 'Is a' or inverse 'Is a' relationship with a set of target concepts. Ancestors include parents and all higher relations. Descendants include children and all lower relations.

Usage

```
parents(conceptIds, SNOMED = getSNOMED(), ...)
ancestors(conceptIds, SNOMED = getSNOMED(), ...)
children(conceptIds, SNOMED = getSNOMED(), ...)
descendants(conceptIds, SNOMED = getSNOMED(), ...)
```

Arguments

conceptIds	character or integer64 vector of SNOMED concept IDs
SNOMED	environment containing a SNOMED dictionary
...	other arguments to pass to relatedConcepts

Value

a bit64 vector of SNOMED CT concepts

Examples

```
SNOMED <- sampleSNOMED()

parents('Heart failure')
children('Heart failure')
ancestors('Heart failure')
descendants('Heart failure')
```

print.SNOMEDconcept	<i>Display a SNOMEDconcept object with descriptions</i>
---------------------	---

Description

SNOMEDconcept is an S3 class for vectors of SNOMED concept IDs as 64-bit integers. This function checks whether the object has the class SNOMEDconcept and is a vector of 64-bit integers.

Usage

```
## S3 method for class 'SNOMEDconcept'
print(x, ...)
```

Arguments

```
x          SNOMEDconcept object, or something that can be coerced to one
...        not required
```

Value

invisibly returns a character vector of the SNOMED CT concepts with descriptions separated by pipe (|)

See Also

Other SNOMEDconcept functions: [as.data.frame.SNOMEDconcept\(\)](#), [c.SNOMEDconcept\(\)](#), [is.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#), [unique.SNOMEDconcept\(\)](#)

Rdiagnosislist	<i>Rdiagnosislist: A package for manipulating SNOMED CT diagnosis lists</i>
----------------	---

Description

The Rdiagnosislist package makes it easy to load a SNOMED dictionary into R and use the hierarchies to search for concepts and navigate relations between concepts.

relatedConcepts	<i>Obtain related concepts for a set of SNOMED CT concepts</i>
-----------------	--

Description

Returns concepts with a particular relation to a supplied set of SNOMED CT concepts

Usage

```
relatedConcepts(
  conceptIds,
  typeId = bit64::as.integer64("116680003"),
  tables = c("RELATIONSHIP", "STATEDRELATIONSHIP"),
  reverse = FALSE,
  recursive = FALSE,
  active_only = TRUE,
  SNOMED = getSNOMED()
)
```

Arguments

conceptIds	character or integer64 vector
typeId	concept ID of relationship type. Defaults to 116680003 = Is a
tables	vector of names of relationship table(s) to use; by default use both RELATIONSHIP and STATEDRELATIONSHIP
reverse	whether to reverse the relationship
recursive	whether to re-apply the function on the outputs
active_only	whether to limit the output to active concepts only
SNOMED	environment containing a SNOMED dictionary

Value

a data.table with the following columns: id, conceptId, type (only if include_synonyms = TRUE), term, active (only if active_only = FALSE)

Examples

```
# Load sample SNOMED CT dictionary
SNOMED <- sampleSNOMED()

# Example: anatomical site of a finding
findingSite <- function(x){
  relatedConcepts(as.SNOMEDconcept(x),
    typeId = as.SNOMEDconcept('Finding site'))
}

description(findingSite('Heart failure'))
# Heart structure (body structure)
```

sampleSNOMED	<i>Sample SNOMED CT dictionary</i>
--------------	------------------------------------

Description

Returns an environment containing a selection of SNOMED CT terms, their relationships and descriptions which are provided with the package

Usage

```
sampleSNOMED()
```

Value

environment containing four data.table objects: CONCEPT, DESCRIPTION, RELATIONSHIP, STATEDRELATIONSHIP and a list named 'metadata'

Examples

```
TEST <- sampleSNOMED()
inactiveIncluded(TEST)
SNOMEDconcept('Heart failure', SNOMED = TEST)

# To display metadata for this SNOMED CT dictionary
sampleSNOMED()$metadata
```

semanticType	<i>Retrieves semantic types using the text 'tag' in the description</i>
--------------	---

Description

Retrieves semantic types using the text 'tag' in the description

Usage

```
semanticType(conceptIds, SNOMED = getSNOMED())
```

Arguments

conceptIds	character or integer64 vector of SNOMED concept IDs
SNOMED	environment containing a SNOMED dictionary

Value

a character vector of semantic tags corresponding to the conceptIDs

Examples

```
SNOMED <- sampleSNOMED()

semanticType(as.SNOMEDconcept(c('Heart failure', 'Is a')))
```

simplify	<i>Retrieves closest single ancestor within a given set of SNOMED CT concepts</i>
----------	---

Description

Returns a vector of SNOMED CT concept IDs for an ancestor of each concept that is within a second list. If multiple ancestors are included in the second list, the concept is not simplified (i.e. the original concept ID is returned). This functionality can be used to translate concepts into simpler forms for display, e.g. 'Heart failure' instead of 'Heart failure with reduced ejection fraction'.

Usage

```
simplify(
  conceptIds,
  ancestorIds,
  SNOMED = getSNOMED(),
  tables = c("RELATIONSHIP", "STATEDRELATIONSHIP")
)
```

Arguments

conceptIds	character or integer64 vector of SNOMED concept IDs for concepts for which an ancestor is sought
ancestorIds	character or integer64 vector of SNOMED concept IDs for possible ancestors
SNOMED	environment containing a SNOMED dictionary
tables	character vector of relationship tables to use

Value

a data.table with the following columns: originalId (integer64) = original conceptId, ancestorId (integer64) = closest single ancestor, or original concept ID if no ancestor is included in the

Examples

```
SNOMED <- sampleSNOMED()

original_terms <- c('Systolic heart failure', 'Is a',
  'Heart failure with reduced ejection fraction',
  'Acute kidney injury due to circulatory failure (disorder)')
# Note in this example 'Is a' has no parents in ancestors,
# and acute kidney failure has two parents in ancestors
# so neither of the parents will be chosen.
# Also test out inclusion of duplicate concepts.

ancestors <- simplify(c(as.SNOMEDconcept(original_terms),
  as.SNOMEDconcept(original_terms)[3:4]),
  as.SNOMEDconcept(c('Heart failure', 'Acute heart failure',
  'Cardiorenal syndrome (disorder)'))))
print(cbind(original_terms, description(ancestors$ancestorId)$term))
```

 SNOMEDcodelist

Convert a data.frame to a SNOMEDcodelist object

Description

SNOMEDcodelist is an S3 class for lists of SNOMED codes. It consists of conceptId and include_desc columns. The option to include descendants allows the creation of more succinct SNOMED codelists.

Usage

```
SNOMEDcodelist(x, include_desc = TRUE, SNOMED = getSNOMED())

as.SNOMEDcodelist(x, ...)
```

Arguments

x	vector of SNOMED CT concept IDs, something which can be coerced to a SNOMEDconcept object, or a data.frame with a column 'conceptId' containing SNOMED CT concept concept IDs in integer64 or text format and optional column 'include_desc' (Boolean) stating whether descendants of the term should be included.
include_desc	Boolean vector stating whether descendants are included, recycled if necessary. Default = TRUE. Ignored if
SNOMED	environment containing a SNOMED dictionary
...	other arguments to pass to SNOMEDcodelist

Details

Input is a data.frame or data.table with column names 'conceptId' and optionally 'include_desc', which is FALSE by default, but if TRUE then the codelist automatically includes all descendants of that concept.

as.SNOMEDcodelist converts its argument into a SNOMEDcodelist but leaves it unchanged if it is already a SNOMEDcodelist.

Value

An object of class 'SNOMEDcodelist'

See Also

Other SNOMEDcodelist functions: [expandSNOMED\(\)](#), [is.SNOMEDcodelist\(\)](#)

Other SNOMEDcodelist functions: [expandSNOMED\(\)](#), [is.SNOMEDcodelist\(\)](#)

Examples

```
SNOMED <- sampleSNOMED()

my_concepts <- SNOMEDconcept('Heart failure')
SNOMEDcodelist(my_concepts)
SNOMEDcodelist(data.frame(conceptId = my_concepts))
as.SNOMEDcodelist(data.frame(conceptId = my_concepts,
  include_desc = TRUE))
```

SNOMED_CONCEPT	<i>Sample concept table from SNOMED CT dictionary</i>
----------------	---

Description

A sample of the SNOMED CT concept table.

Usage

```
data(CONCEPT)
```

Format

An object of class "data.table"

Details

- `id integer64`: SNOMED CT conceptId (primary key)
- `moduleId integer64`: class of SNOMED CT concept (whether it is used for recording information or is a metadata concept)
- `definitionStatusId integer64`: 900000000000074008 = primitive concept, 900000000000073002 = defined by conditions
- `effectiveTime IDate`: when the concept became active
- `active logical`: whether this concept is currently active

Examples

```
# Create a TEST environment and load the sample dictionaries
TEST <- new.env()
data(CONCEPT, envir = TEST)
data(DESCRIPTION, envir = TEST)
data(RELATIONSHIP, envir = TEST)
data(STATEDRELATIONSHIP, envir = TEST)

# Show properties of the CONCEPT table
str(TEST$CONCEPT)
```

SNOMED_DESCRIPTION *Sample description table from SNOMED CT dictionary*

Description

A sample of the SNOMED CT description table. Each concept may has a fully specified name and may have any number of synonyms.

Usage

```
data(DESCRIPTION)
```

Format

An object of class "data.table"

Details

- id integer64: description ID
- moduleId integer64: class of SNOMED CT concept (whether it is used for recording information or is a metadata concept)
- conceptId integer64: SNOMED CT concept ID
- languageCode character: 'en' = English
- typeId integer64: 900000000000013009 = Synonym, 90000000000003001 = Fully Specified Name
- term character: term description
- caseSignificanceId integer64: 900000000000020002 = Initial character case sensitive, 90000000000017005 = Whole term case sensitive, 900000000000448009 = Whole term case insensitive
- effectiveTime IDate: when the concept became active
- active logical: whether this concept is currently active

Examples

```
# Create a TEST environment and load the sample dictionaries
TEST <- new.env()
data(CONCEPT, envir = TEST)
data(DESCRIPTION, envir = TEST)
data(RELATIONSHIP, envir = TEST)
data(STATEDRELATIONSHIP, envir = TEST)

# Show properties of the DESCRIPTION table
str(TEST$DESCRIPTION)
```

SNOMED_RELATIONSHIP *Sample relationship tables from SNOMED CT dictionary*

Description

Samples of the SNOMED CT tables of stated relationships (RELATIONSHIP) and inferred relationships (RELATIONSHIP).

Usage

```
data(RELATIONSHIP); data(STATEDRELATIONSHIP)
```

```
STATEDRELATIONSHIP
```

Format

An object of class "data.table"

An object of class data.table (inherits from data.frame) with 61 rows and 10 columns.

Details

- id integer64: ID of the relationship record (primary key)
- active logical: whether this concept is currently active
- moduleId integer64: class of SNOMED CT concept (whether it is used for recording information or is a metadata concept)
- sourceId integer64: source SNOMED CT concept for the relationship
- destinationId integer64: destination SNOMED CT concept for the relationship
- relationshipGroup integer: group ID for relationships that are grouped
- characteristicTypeId integer64: 900000000000011006 = Inferred relationship
- modifierId integer64: 900000000000451002 = Existential restriction modifier
- effectiveTime IDate: when the concept became active
- typeId integer64: type of relationship, e.g. 116680003 = Is a, 42752001 = Due to, 246090004 = Associated finding, 363698007 = Finding site, 363702006 = Has focus

Examples

```
# Create a TEST environment and load the sample dictionaries
TEST <- new.env()
data(CONCEPT, envir = TEST)
data(DESCRIPTION, envir = TEST)
data(RELATIONSHIP, envir = TEST)
data(STATEDRELATIONSHIP, envir = TEST)

# Show properties of the relationship tables
str(TEST$RELATIONSHIP)
str(TEST$STATEDRELATIONSHIP)
```

union.SNOMEDconcept *Set operations for SNOMEDconcept vectors*

Description

The default set functions in the base package do not handle integer64 vectors correctly, so this package also provides new generic functions for union, intersect and setdiff, which enable the appropriate object-specific function to be called according to the class of the vector. This means that SNOMEDconcept vectors will remain as SNOMEDconcept vectors when these functions are used.

Usage

```
## S3 method for class 'SNOMEDconcept'  
union(x, y)  
  
union(x, y)  
  
## Default S3 method:  
union(x, y)  
  
## S3 method for class 'SNOMEDconcept'  
intersect(x, y)  
  
intersect(x, y)  
  
## Default S3 method:  
intersect(x, y)  
  
## S3 method for class 'SNOMEDconcept'  
setdiff(x, y)  
  
setdiff(x, y)  
  
## Default S3 method:  
setdiff(x, y)
```

Arguments

x	SNOMEDconcept vector
y	SNOMEDconcept vector, or an object that can be coerced to SNOMEDconcept by as.SNOMEDconcept

Value

an integer64 vector of SNOMEDconcept class

See Also

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Other SNOMEDconcept functions: `as.data.frame.SNOMEDconcept()`, `c.SNOMEDconcept()`, `is.SNOMEDconcept()`, `print.SNOMEDconcept()`, `unique.SNOMEDconcept()`

Examples

```
sys_acute <- SNOMEDconcept(c('Systolic heart failure',
  'Acute heart failure'), SNOMED = sampleSNOMED())
acute_left_right <- SNOMEDconcept(c('Acute heart failure',
  'Left heart failure', 'Right heart failure'),
  SNOMED = sampleSNOMED())
union(sys_acute, acute_left_right)
intersect(sys_acute, acute_left_right)
setdiff(sys_acute, acute_left_right)
```

`unique.SNOMEDconcept` *Unique vector of SNOMED CT concepts*

Description

SNOMEDconcept is an S3 class for vectors of SNOMED concept IDs as 64-bit integers. This function returns a vector containing only unique SNOMEDconcept values.

Usage

```
## S3 method for class 'SNOMEDconcept'
unique(x, ...)
```

Arguments

`x` SNOMEDconcept vector
`...` other variables to pass on to the underlying 'unique' function

Value

SNOMEDconcept vector with duplicates removed

See Also

Other SNOMEDconcept functions: [as.data.frame.SNOMEDconcept\(\)](#), [c.SNOMEDconcept\(\)](#), [is.SNOMEDconcept\(\)](#), [print.SNOMEDconcept\(\)](#), [union.SNOMEDconcept\(\)](#)

Examples

```
hf <- SNOMEDconcept('Heart failure', SNOMED = sampleSNOMED())
hf2 <- c(hf, hf)
unique(hf2)
```

Index

- * **SNOMEDcodelist functions**
 - expandSNOMED, 6
 - is.SNOMEDcodelist, 9
 - SNOMEDcodelist, 15
- * **SNOMEDconcept functions**
 - as.data.frame.SNOMEDconcept, 2
 - c.SNOMEDconcept, 4
 - is.SNOMEDconcept, 9
 - print.SNOMEDconcept, 11
 - union.SNOMEDconcept, 20
 - unique.SNOMEDconcept, 21
- * **SNOMEDsample**
 - SNOMED_RELATIONSHIP, 19
- * **datasets**
 - SNOMED_CONCEPT, 17
 - SNOMED_DESCRIPTION, 18
 - SNOMED_RELATIONSHIP, 19
- ancestors (parents), 11
- as.data.frame.SNOMEDconcept, 2, 4, 10, 12, 21, 22
- as.integer64.SNOMEDconcept
 - (as.data.frame.SNOMEDconcept), 2
- as.SNOMEDcodelist (SNOMEDcodelist), 15
- as.SNOMEDconcept
 - (as.data.frame.SNOMEDconcept), 2
- attrConcept, 3
- c.SNOMEDconcept, 3, 4, 10, 12, 21, 22
- children (parents), 11
- CONCEPT (SNOMED_CONCEPT), 17
- contractSNOMED (expandSNOMED), 6
- createSNOMEDindices, 5
- descendants (parents), 11
- DESCRIPTION (SNOMED_DESCRIPTION), 18
- description, 5
- expandSNOMED, 6, 9, 16
- getSNOMED, 7
- hasAttributes, 7
- inactiveIncluded, 8
- intersect (union.SNOMEDconcept), 20
- is.SNOMEDcodelist, 6, 9, 16
- is.SNOMEDconcept, 3, 4, 9, 12, 21, 22
- loadSNOMED, 10
- parents, 11
- print.SNOMEDconcept, 3, 4, 10, 11, 21, 22
- Rdiagnosislist, 12
- relatedConcepts, 12
- RELATIONSHIP (SNOMED_RELATIONSHIP), 19
- sampleSNOMED, 13
- semanticType, 14
- setdiff (union.SNOMEDconcept), 20
- simplify, 14
- SNOMED_CONCEPT, 17
- SNOMED_DESCRIPTION, 18
- SNOMED_RELATIONSHIP, 19
- SNOMED_STATEDRELATIONSHIP
 - (SNOMED_RELATIONSHIP), 19
- SNOMEDcodelist, 6, 9, 15
- SNOMEDconcept
 - (as.data.frame.SNOMEDconcept), 2
- STATEDRELATIONSHIP
 - (SNOMED_RELATIONSHIP), 19
- union (union.SNOMEDconcept), 20
- union.SNOMEDconcept, 3, 4, 10, 12, 20, 22
- unique.SNOMEDconcept, 3, 4, 10, 12, 21, 21