

Combinatory Library Documentation

Author: João Amorim

Module

[Types](#)

[Combinations](#)

Types

Type Definitions

```
type Class = String
data Partition = Node String [Partition] | Leaf Class deriving Show
data Category = Cat String [Partition] deriving Show
type Input = [Category]
data Result = R String Class deriving Show
type Combination = [Result]
type Suite = [Combination]
data Operator = Plus | Times | ST | STN | Union | Join | Intersect
data LogicOp = And | Or | Xor | Nor | Equi | Impl
data Rule = Rop Operator Rule Rule | RLog LogicOp [String] [String] | P [String] | NotP
[String] | Ctg String
```

Functions

```
showSuite :: Suite -> String -- prints suite
```

Combinations

Functions

nCombs :: Input -> Int -- returns the possible number of combinations from an input

nClasses :: Input -> Int -- returns the number of classes in an input

nCats :: Input -> Int -- returns the number of categories in an input

minimalComb :: Input -> Suite -- calculates the minimal combinations from the input

fullComb :: Input -> Suite -- calculates the full combinations from the input

semantic :: Rule -> Input -> Suite -- calculates the combinations according to the rule

minRuledComb :: Rule -> Input -> Suite -- calculates a minimal number of combinations in the input according to the rule

fullRuledComb :: Rule -> Input -> Suite -- calculates all the combinations in the input that obey the rule

fC :: Suite -> Suite -> Suite -- concats every combinations in the first suite with every combination in the second suite

mC :: Suite -> Suite -> Suite -- returns the minimal combination of the two suites

getClasses :: Category -> [Class] -- returns the list of classes of the category

classes :: Partition -> [Class] -- returns the list of classes of the partition

getCat :: String -> Input -> String -- finds the name of the category of a given partition

fClasses :: [String] -> Input -> Suite -- returns a suite with the classes of the given partitions

fNotClasses :: [String] -> Input -> Suite -- returns a suite with the classes of the sibling partitions of the given ones

fGetClasses :: String -> Input -> Suite -- returns a suite with the classes of the given category

covers :: Combination -> Combination -> Bool -- returns True if every element of the first combination is present in the second (the second covers the first)

coversComb :: Combination -> Suite -> Bool -- returns True if one combination in the suite covers the given combination