

Derivation operators

The methods that implement the derivation operators are named after them: `intent()`, `extent()` and `closure()`. They can be applied on objects of type "Set", representing fuzzy sets of objects or attributes:

```
> S <- Set$new(fc$objects, O1 = 1, O2 = 1)
> S
{O1, O2}
> fc$intent(S)
{P2 [0.5], P4 [0.5]}
> T <- Set$new(fc$attributes, P1 = 1, P3 = 1)
> T
{P1, P3}
> fc$extent(T)
{}
> fc$closure(T)
{P1, P2, P3, P4}
```

In addition, we can perform *clarification* on the formal context, by using `fc$clarify()`, giving:

FormalContext with 3 objects and 3 attributes.

	P1	P3	[P2, P4]
O1	0	0.5	0.5
O4	0	1	0.5
[O2, O3]	0.5	0	1

The duplicated rows and columns in the formal context have been collapsed, and the corresponding attributes and objects' names are grouped together between brackets, e.g., [P2, P4].

Concept lattice

The command to compute the concept lattice for a "FormalContext" `fc` is `fc$find_concepts()`. The lattice is stored in `fc$concepts`, which is of the "ConceptLattice" class.

```
> fc$concepts
A set of 8 concepts:
1: ({O1, O2, O3, O4}, {P2 [0.5], P4 [0.5]})
2: ({O1, O4}, {P2 [0.5], P3 [0.5], P4 [0.5]})
3: ({O1 [0.5], O4}, {P2 [0.5], P3, P4 [0.5]})
4: ({O1 [0.5], O2, O3, O4 [0.5]}, {P2, P4})
5: ({O1 [0.5], O4 [0.5]}, {P2, P3, P4})
6: ({O2, O3}, {P1 [0.5], P2, P4})
7: ({O2 [0.5], O3 [0.5]}, {P1, P2, P4})
8: ({}, {P1, P2, P3, P4})
```

In order to know the *cardinality* of the set of concepts (that is, the number of concepts), we can use `fc$concepts$size()`, which gives 8 in this case. The complete list of concepts can be printed with `fc$concepts$print()`, or simply `fc$concepts`. Also, they can be translated to \LaTeX using the `to_latex()` method, as mentioned before.

The typical subsetting operation in R with brackets is implemented to select specific concepts from the lattice, giving their indexes or a boolean vector indicating which concepts to keep. The same rules for subsetting as in R base apply:

```
> fc$concepts[c(1:3, 5, 8)]
A set of 5 concepts:
1: ({O1, O2, O3, O4}, {P2 [0.5], P4 [0.5]})
2: ({O1, O4}, {P2 [0.5], P3 [0.5], P4 [0.5]})
3: ({O1 [0.5], O4}, {P2 [0.5], P3, P4 [0.5]})
4: ({O1 [0.5], O4 [0.5]}, {P2, P3, P4})
5: ({}, {P1, P2, P3, P4})
```

In addition, the user can compute concepts' *support* (the proportion of objects whose set of attributes contains the intent of a given concept) by means of `fc$concepts$support()`.

```
> fc$concepts$support()
[1] 1.00 0.50 0.25 0.50 0.00 0.50 0.00 0.00
```