The Base Functions of the RelView System, Version 7.0*

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität Kiel
Olshausenstraße 40
D-24098 Kiel, Germany

The base functions of RelView can be divided into several parts:

1. Base functions for calculating constant relations and domains:

Syntax	Meaning
`L(R)`	Universal relation with the same dimension as `R`
`O(R)`	Empty relation with the same dimension as `R`
`I(R)`	Identity relation with the same dimension as `R`
`Ln1(R)`	Universal column vector with the same row number as `R`
`On1(R)`	Empty column vector with the same row number as `R`
`L1n(R)`	Universal row vector with the same column number as `R`
`O1n(R)`	Empty row vector with the same column number as `R`
`dom(R)`	Domain `R*Ln1(R^)` of relation `R` as column vector

2. Base functions for calculating Boolean operations:

Syntax	Meaning
`-R`	Negation (complement) of relation `R`
`R | S`	Union (join) of `R` and `S`
`R & S`	Intersection (meet) of `R` and `S`

3. Base functions for calculating relationalgebraic operations:

Syntax	Meaning
`R^`	Transposition of relation `R`
`R*S`	Composition (product) of `R` and `S`

4. Base functions for calculating residuals and symmetric quotients:

Syntax	Meaning
`S/R`	Left residual of `R` and `S`
`R\S`	Right residual of `R` and `S`
`syq(R,S)`	Symmetric quotient of `R` and `S`

*WWW: http://www.informatik.uni-kiel.de/~progsys/relview.shtml

5. Base functions for calculating closures:

Syntax	Meaning
`trans(R)`	Transitive closure of `R`
`refl(R)`	Reflexive closure of `R`
`symm(R)`	Symmetric closure of `R`

6. Various base functions concerning vectors and points without choice operations:

Syntax	Meaning
`inj(v)`	Injection induced by the non-empty vector `v`
`init(v)`	Initial point with the same dimension as the vector `v`
`succ(v)`	Homogeneous successor relation with a dimension given by the number of rows of the vector `v`
`next(p)`	Successor of the point `p` with the same dimension as `p`

7. Base operations for choices:

Syntax	Meaning
`point(v)`	A point included in the non-empty column vector `v`
`atom(R)`	An atom (a relation consisting of one pair) included in the non-empty relation `R`

8. Base operations which generate random relations. In the following, `XY` stands for two digits between `00` and `99` and denotes the probability that a pair is contained in the result:

Syntax	Meaning
`randomXY(R)`	Generation of a random relation with the same dimension as `R`
`randomcfXY(R)`	Generation of a cyclefree random relation with the same dimension as the homogeneous relation `R`
`randomperm(v)`	Generation of a random permutation, where the dimension is given by the number of rows of the vector `v`

9. Base functions for certain tests on relations. The result is *true* (represented by the universal relation on a singleton set) or *false* (represented by the empty relation on a singleton set):

Syntax	Meaning
`empty(R)`	Test, whether `R` is empty
`unival(R)`	Test, whether `R` is univalent
`eq(R,S)`	Test, whether `R` and `S` are equal
`incl(R,S)`	Test, whether `R` is included in `S`
`cardeq(R,S)`	Test, whether the cardinalities of `R` and `S` are equal
`cardlt(R,S)`	Test, whether the cardinality of `R` is less than that of `S`
`cardgt(R,S)`	Test, whether the cardinality of `R` is greater than that of `S`

10. Base functions concerning operations on powersets:

Syntax	Meaning
`epsi(v)`	Membership relation, where the cardinality of the base set is given by the row number of the vector `v`
`cardrel(v)`	Size comparison relation on a powerset, where the cardinality of the base set is given by the row number of the vector `v`
`cardfilter(v,w)`	If `v` is a vector with a powerset 2^X as argument set and `w` is a vector with $n \leq \|X\| + 1$ columns, then the operation selects from `v` all sets s fulfilling $\|s\| < n$

11. Base functions concerning relational product and sum domains. Most of these base functions take a domain definition as argument, the result however is always a relation:

Syntax	Meaning
`1-st(DD)`	1st component (`DD` domain)
`2-nd(DD)`	2nd component (`DD` domain)
`p-1(PP)`	Projection onto the 1st component (`PP` product domain)
`p-2(PP)`	Projection onto the 2nd component (`PP` product domain)
`p-ord(PP)`	Product order (`PP` product domain)
`[R,S]`	Tupling of relations
`i-1(SS)`	Injection into 1st component (`SS` sum domain)
`i-2(SS)`	Injection into 2nd component (`SS` sum domain)
`s-ord(SS)`	Sum order (`SS` sum domain)
`R+S`	Sum of relations

12. Base functions concerning function domains:

Syntax	Meaning
`part-f(R,S)`	Columnwise representation of partial functions
`tot-f(R,S)`	Columnwise representation of total functions