5.272 nclass

DESCRIPTION LINKS GRAPH

Origin

Derived from `nvalue`.

Constraint

`nclass(NCLASS, VARIABLES, PARTITIONS)`

Type

`VALUES : collection(val−int)`

Arguments

```
NCLASS      :  dvar
VARIABLES   :  collection(var−dvar)
PARTITIONS  :  collection(p − VALUES)
```

Restrictions

```
|VALUES| ≥ 1
required(VALUES, val)
distinct(VALUES, val)
NCLASS ≥ 0
NCLASS ≤ min(|VARIABLES|, |PARTITIONS|)
NCLASS ≤range(VARIABLES.var)
required(VARIABLES, var)
required(PARTITIONS, p)
|PARTITIONS| ≥ 2
```

Purpose

Number of partitions of the collection `PARTITIONS` such that at least one value is assigned to at least one variable of the collection `VARIABLES`.

Example

$$\left(2, \langle 3, 2, 7, 2, 6\rangle, \langle \mathtt{p} - \langle 1, 3\rangle, \mathtt{p} - \langle 4\rangle, \mathtt{p} - \langle 2, 6\rangle\rangle\right)$$

Note that the values of $\langle 3, 2, 7, 2, 6\rangle$ occur within partitions $\mathtt{p} - \langle 1, 3\rangle$ and $\mathtt{p} - \langle 2, 6\rangle$ but not within $\mathtt{p} - \langle 4\rangle$. Consequently, the `nclass` constraint holds since its first argument `NCLASS` is set to value 2.

Typical

```
NCLASS > 1
NCLASS < |VARIABLES|
NCLASS <range(VARIABLES.var)
|VARIABLES| > |PARTITIONS|
```

Symmetries

- Items of `VARIABLES` are permutable.
- Items of `PARTITIONS` are permutable.
- Items of `PARTITIONS.p` are permutable.
- An occurrence of a value of `VARIABLES.var` can be replaced by any other value that also belongs to the same partition of `PARTITIONS`.
- All occurrences of two distinct tuples of values in `VARIABLES.var` or `PARTITIONS.p.val` can be swapped; all occurrences of a tuple of values in `VARIABLES.var` or `PARTITIONS.p.val` can be renamed to any unused tuple of values.

Arg. properties

- Functional dependency: NCLASS determined by VARIABLES and PARTITIONS.
- Extensible wrt. VARIABLES when NCLASS $=$ |PARTITIONS|.

Algorithm

[27, 40].

See also

related: nequivalence *(*variable $\in$ partition *replaced by* variable $\bmod$ constant*)*, ninterval *(*variable $\in$ partition *replaced by* variable$/$constant*)*, npair *(*variable $\in$ partition *replaced by* pair *of* variables*)*.

specialisation: nvalue *(*variable $\in$ partition *replaced by* variable*)*.

used in graph description: in_same_partition.

Keywords

characteristic of a constraint: partition.

constraint arguments: pure functional dependency.

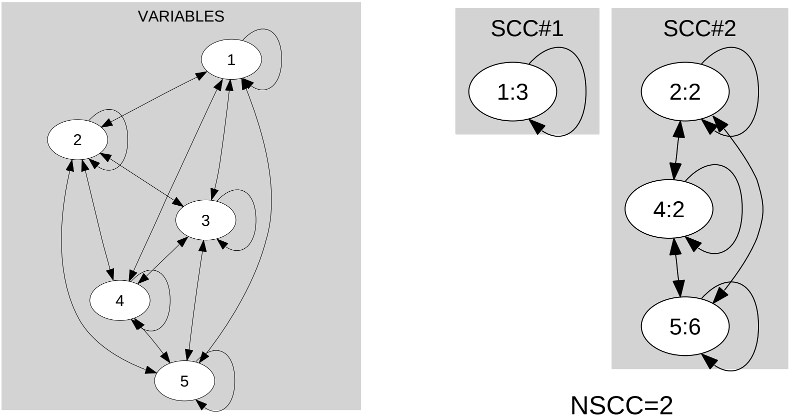
constraint type: counting constraint, value partitioning constraint.

final graph structure: strongly connected component, equivalence.

modelling: number of distinct equivalence classes, functional dependency.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	in_same_partition(variables1.var, variables2.var, PARTITIONS)
Graph property(ies)	**NSCC** = NCLASS

Graph model Parts (A) and (B) of Figure 5.592 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NSCC** graph property we show the different strongly connected components of the final graph. Each strongly connected component corresponds to a class of values that was assigned to some variables of the VARIABLES collection. We effectively use two classes of values that respectively correspond to values $\{3\}$ and $\{2, 6\}$. Note that we do not consider value 7 since it does not belong to the different classes of values we gave: all corresponding arc constraints do not hold.



(A) (B)

Figure 5.592: Initial and final graph of the nclass constraint