5.288 nvalues

DESCRIPTION LINKS GRAPH

Origin

Inspired by `nvalue` and `count`.

Constraint

`nvalues(VARIABLES, RELOP, LIMIT)`

Arguments

`VARIABLES`	:	`collection(var−dvar)`
`RELOP`	:	`atom`
`LIMIT`	:	`dvar`

Restrictions

`required(VARIABLES, var)`
`RELOP` $\in [=, \neq, <, \geq, >, \leq]$

Purpose

Let N be the number of distinct values assigned to the variables of the `VARIABLES` collection. Enforce condition N `RELOP LIMIT` to hold.

Example

$(\langle 4, 5, 5, 4, 1, 5\rangle, =, 3)$

The `nvalues` constraint holds since the number of distinct values occurring within the collection $\langle 4, 5, 5, 4, 1, 5\rangle$ is equal (i.e., `RELOP` is set to $=$) to its third argument `LIMIT` $= 3$.

Typical

$|$`VARIABLES`$| > 1$
`LIMIT` > 1
`LIMIT` $< |$`VARIABLES`$|$
`RELOP` $\in [=, <, \geq, >, \leq]$

Symmetries

- Items of `VARIABLES` are permutable.
- All occurrences of two distinct values of `VARIABLES.var` can be swapped; all occurrences of a value of `VARIABLES.var` can be renamed to any unused value.

Arg. properties

- Contractible wrt. `VARIABLES` when `RELOP` $\in [<, \leq]$.
- Contractible wrt. `VARIABLES` when `RELOP` $\in [=]$, `LIMIT` $= 1$ and $|$`VARIABLES`$| > 0$.
- Contractible wrt. `VARIABLES` when `RELOP` $\in [=]$ and `LIMIT` $= |$`VARIABLES`$|$.
- Extensible wrt. `VARIABLES` when `RELOP` $\in [\geq, >]$.

Usage

Used in the **Constraint(s) on sets** slot for defining some constraints like `assign_and_nvalues`, `circuit_cluster` or `coloured_cumulative`.

Reformulation

The `nvalues(VARIABLES, RELOP , LIMIT)` constraint can be expressed in term of the conjunction `nvalue`$(NV,$ `VARIABLES`$) \wedge NV$ `RELOP LIMIT`.

Systems

nvalues in **Gecode**.

Used in `assign_and_nvalues`, `circuit_cluster`, `coloured_cumulative`, `coloured_cumulatives`.

See also **assignment dimension added:** `assign_and_nvalues`.

common keyword: `nvalues_except_0` *(counting constraint,number of distinct values)*.

specialisation: `nvalue` *(replace a comparison with the number of distinct values by an equality with the number of distinct values)*.

Keywords **constraint type:** counting constraint, value partitioning constraint.

final graph structure: strongly connected component, equivalence.

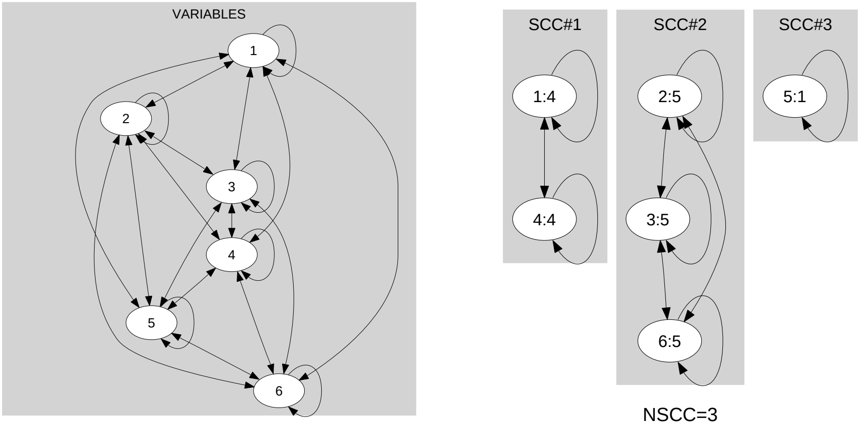
modelling: number of distinct equivalence classes, number of distinct values.

problems: domination.

Cond. implications `nvalues`(`VARIABLES`, `RELOP`, `LIMIT`)
with `minval`(`VARIABLES.var`) > 0
implies `nvalues_except_0`(`VARIABLES`, `RELOP`, `LIMIT`).

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	**NSCC** RELOP LIMIT
Graph class	EQUIVALENCE

Graph model Parts (A) and (B) of Figure 5.602 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 value that is assigned to some variables of the VARIABLES collection. The 3 following values 1, 4 and 5 are used by the variables of the VARIABLES collection.



(A) (B)

Figure 5.602: Initial and final graph of the nvalues constraint