5.62 common

DESCRIPTION LINKS GRAPH

Origin N. Beldiceanu

Constraint common(NCOMMON1, NCOMMON2, VARIABLES1, VARIABLES2)

Arguments

NCOMMON1	:	dvar
NCOMMON2	:	dvar
VARIABLES1	:	collection(var−dvar)
VARIABLES2	:	collection(var−dvar)

Restrictions

NCOMMON1 ≥ 0
NCOMMON1 $\leq$ |VARIABLES1|
NCOMMON2 ≥ 0
NCOMMON2 $\leq$ |VARIABLES2|
required(VARIABLES1, var)
required(VARIABLES2, var)

Purpose

NCOMMON1 is the number of variables of the collection of variables VARIABLES1 taking a value in VARIABLES2.
NCOMMON2 is the number of variables of the collection of variables VARIABLES2 taking a value in VARIABLES1.

Example

$$\left\langle \begin{array}{c} 3, 4, \langle 1, 9, 1, 5\rangle, \\ \left\langle \begin{array}{c} \mathtt{var}-2, \\ \mathtt{var}-1, \\ \mathtt{var}-9, \\ \mathtt{var}-9, \\ \mathtt{var}-6, \\ \mathtt{var}-9 \end{array} \right\rangle \end{array} \right\rangle$$

The common constraint holds since:

- Its first argument NCOMMON1 $= 3$ corresponds to the number of values of the collection $\langle 1, 9, 1, 5\rangle$ that occur within $\langle 2, 1, 9, 9, 6, 9\rangle$.
- Its second argument NCOMMON2 $= 4$ corresponds to the number of values of the collection $\langle 2, 1, 9, 9, 6, 9\rangle$ that occur within $\langle 1, 9, 1, 5\rangle$.

Typical

|VARIABLES1| > 1
range(VARIABLES1.var) > 1
|VARIABLES2| > 1
range(VARIABLES2.var) > 1

Symmetries

- Arguments are permutable w.r.t. permutation (NCOMMON1, NCOMMON2) (VARIABLES1, VARIABLES2).
- Items of VARIABLES1 are permutable.
- Items of VARIABLES2 are permutable.
- All occurrences of two distinct values in VARIABLES1.var or VARIABLES2.var can be swapped; all occurrences of a value in VARIABLES1.var or VARIABLES2.var can be renamed to any unused value.

Remark

It was shown in [64] that, finding out whether the common constraint has a solution or not is NP-hard. This was achieved by reduction from 3-SAT.

See also

common keyword: alldifferent_on_intersection, nvalue_on_intersection, same_intersection *(constraint on the intersection)*.

generalisation: common_interval *(*variable *replaced by* variable/constant*)*, common_modulo *(*variable *replaced by* variable mod constant*)*, common_partition *(*variable *replaced by* variable ∈ partition*)*.

related: among_var, roots.

root concept: among.

specialisation: uses *(*NCOMMON2=|VARIABLES2|*)*.

Keywords

complexity: 3-SAT.

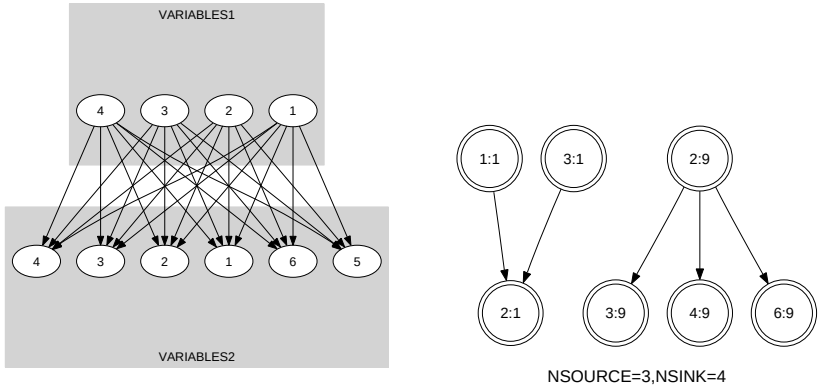
constraint arguments: constraint between two collections of variables.

constraint type: constraint on the intersection.

final graph structure: acyclic, bipartite, no loop.

Arc input(s)	VARIABLES1 VARIABLES2
Arc generator	$PRODUCT \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	• **NSOURCE**= NCOMMON1 • **NSINK**= NCOMMON2
Graph class	• ACYCLIC • BIPARTITE • NO_LOOP

Graph model Parts (A) and (B) of Figure 5.133 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NSOURCE** and **NSINK** graph properties, the source and sink vertices of the final graph are stressed with a double circle. Since the final graph has only 3 sources and 4 sinks the variables NCOMMON1 and NCOMMON2 are respectively equal to 3 and 4. Note that all the vertices corresponding to the variables that take values 5, 2 or 6 were removed from the final graph since there is no arc for which the associated equality constraint holds.



(A) **(B)**

Figure 5.133: Initial and final graph of the common constraint