5.77 common_modulo

DESCRIPTION LINKS GRAPH

Origin Derived from common.

Constraint common_modulo(NCOMMON1, NCOMMON2, VARIABLES1, VARIABLES2, M)

Arguments

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

Restrictions

$\texttt{NCOMMON1} \geq 0$
$\texttt{NCOMMON1} \leq |\texttt{VARIABLES1}|$
$\texttt{NCOMMON2} \geq 0$
$\texttt{NCOMMON2} \leq |\texttt{VARIABLES2}|$
required(VARIABLES1, var)
required(VARIABLES2, var)
$\texttt{M} > 0$

Purpose

NCOMMON1 is the number of variables of the collection of variables VARIABLES1 taking a value situated in an equivalence class (congruence modulo a fixed number M) derived from the values assigned to the variables of VARIABLES2 and from M.
NCOMMON2 is the number of variables of the collection of variables VARIABLES2 taking a value situated in an equivalence class (congruence modulo a fixed number M) derived from the values assigned to the variables of VARIABLES1 and from M.

Example

$$(3, 4, \langle 0, 4, 0, 8\rangle, \langle 7, 5, 4, 9, 2, 4\rangle, 5)$$

In the example, the last argument $\texttt{M} = 5$ defines the equivalence classes $a \equiv 0 \pmod 5$, $a \equiv 1 \pmod 5$, $a \equiv 2 \pmod 5$, $a \equiv 3 \pmod 5$, and $a \equiv 4 \pmod 5$ where a is an integer. As a consequence the items of collection $\langle 0, 4, 0, 8\rangle$ respectively correspond to the equivalence classes $a \equiv 0 \pmod 5$, $a \equiv 4 \pmod 5$, $a \equiv 0 \pmod 5$, and $a \equiv 3 \pmod 5$. Similarly the items of collection $\langle 7, 5, 4, 9, 2, 4\rangle$ respectively correspond to the equivalence classes $a \equiv 2 \pmod 5$, $a \equiv 0 \pmod 5$, $a \equiv 4 \pmod 5$, $a \equiv 4 \pmod 5$, $a \equiv 2 \pmod 5$, and $a \equiv 4 \pmod 5$. The common_modulo constraint holds since:

- Its first argument $\texttt{NCOMMON1} = 3$ is the number of equivalence classes associated with the items of collection $\langle 0, 4, 0, 8\rangle$ that also correspond to equivalence classes associated with $\langle 7, 5, 4, 9, 2, 4\rangle$.
- Its second argument $\texttt{NCOMMON2} = 4$ is the number of equivalence classes associated with the items of collection $\langle 7, 5, 4, 9, 2, 4\rangle$ that also correspond to equivalence classes associated with $\langle 0, 4, 0, 8\rangle$.

Typical

$|\mathtt{VARIABLES1}| > 1$
$\mathtt{range}(\mathtt{VARIABLES1.var}) > 1$
$|\mathtt{VARIABLES2}| > 1$
$\mathtt{range}(\mathtt{VARIABLES2.var}) > 1$
$\mathtt{M} > 1$
$\mathtt{M} < \mathtt{maxval}(\mathtt{VARIABLES1.var})$
$\mathtt{M} < \mathtt{maxval}(\mathtt{VARIABLES2.var})$

Symmetries

- Arguments are permutable w.r.t. permutation (NCOMMON1, NCOMMON2) (VARIABLES1, VARIABLES2) (M).
- Items of VARIABLES1 are permutable.
- Items of VARIABLES2 are permutable.
- An occurrence of a value u of VARIABLES1.var can be replaced by any other value v such that v is congruent to u modulo M.
- An occurrence of a value u of VARIABLES2.var can be replaced by any other value v such that v is congruent to u modulo M.

Arg. properties

- Functional dependency: NCOMMON1 determined by VARIABLES1, VARIABLES2 and M.
- Functional dependency: NCOMMON2 determined by VARIABLES1, VARIABLES2 and M.

See also

specialisation: common *(*variable mod constant *replaced by* variable*)*.

Keywords

characteristic of a constraint: modulo.

constraint arguments: constraint between two collections of variables, pure functional dependency.

final graph structure: acyclic, bipartite, no loop.

modelling: functional dependency.

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

Graph model Parts (A) and (B) of Figure 5.183 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 graph has only 3 sources and 4 sinks the variables NCOMMON1 and NCOMMON2 are respectively equal to 3 and 4. Note that the vertices corresponding to the variables that take values 8, 7 or 2 were removed from the final graph since there is no arc for which the associated arc constraint holds.

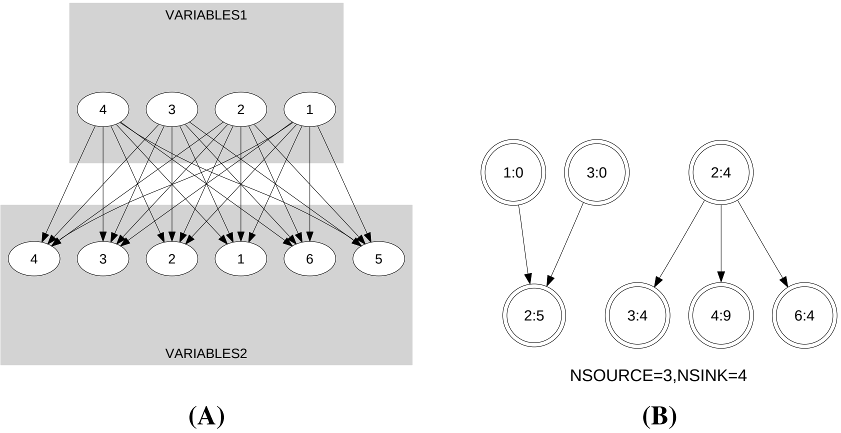

Figure 5.183: Initial and final graph of the common_modulo constraint