5.340 same_modulo

DESCRIPTION **LINKS** **GRAPH**

Origin

Derived from `same`.

Constraint

`same_modulo(VARIABLES1, VARIABLES2, M)`

Arguments

`VARIABLES1`	:	`collection(var−dvar)`
`VARIABLES2`	:	`collection(var−dvar)`
`M`	:	`int`

Restrictions

$|\texttt{VARIABLES1}| = |\texttt{VARIABLES2}|$
`required(VARIABLES1, var)`
`required(VARIABLES2, var)`
$\texttt{M} > 0$

Purpose

For each integer R in $[0, \texttt{M} - 1]$, let $N1_R$ (respectively $N2_R$) denote the number of variables of `VARIABLES1` (respectively `VARIABLES2`) that have R as a rest when divided by `M`. For all R in $[0, \texttt{M} - 1]$ we have that $N1_R = N2_R$.

Example

$(\langle 1, 9, 1, 5, 2, 1\rangle, \langle 6, 4, 1, 1, 5, 5\rangle, 3)$

The values of the first collection $\langle 1, 9, 1, 5, 2, 1\rangle$ are respectively associated with the equivalence classes $1 \bmod 3 = 1$, $9 \bmod 3 = 0$, $1 \bmod 3 = 1$, $5 \bmod 3 = 2$, $2 \bmod 3 = 2$, $1 \bmod 3 = 1$. Therefore the equivalence classes 0, 1, and 2 are respectively used 1, 3, and 2 times. Similarly, the values of the second collection $\langle 6, 4, 1, 1, 5, 5\rangle$ are respectively associated with the equivalence classes $6 \bmod 3 = 0$, $4 \bmod 3 = 1$, $1 \bmod 3 = 1$, $1 \bmod 3 = 1$, $5 \bmod 3 = 2$, $5 \bmod 3 = 2$. Therefore the equivalence classes 0, 1, and 2 are respectively used 1, 3, and 2 times. Consequently the `same_modulo` constraint holds. Figure 5.692 illustrates this correspondence.

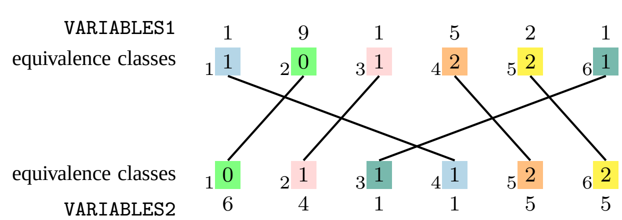


Figure 5.692: Illustration of the correspondence between the items of the `VARIABLES1` and of the `VARIABLES2` collections of the **Example** slot

Typical

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

Symmetries

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

Arg. properties

Aggregate: VARIABLES1(union), VARIABLES2(union), M(id).

Used in

k_same_modulo.

See also

implies: used_by_modulo.

soft variant: soft_same_modulo_var *(variable-based violation measure)*.

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

system of constraints: k_same_modulo.

Keywords

characteristic of a constraint: sort based reformulation, modulo.

combinatorial object: permutation.

constraint arguments: constraint between two collections of variables.

Arc input(s)	`VARIABLES1` `VARIABLES2`
Arc generator	$PRODUCT \mapsto$`collection`(`variables1`, `variables2`)
Arc arity	2
Arc constraint(s)	`variables1.var` mod `M` = `variables2.var` mod `M`
Graph property(ies)	• for all connected components: **NSOURCE**=**NSINK** • **NSOURCE**= |`VARIABLES1`| • **NSINK**= |`VARIABLES2`|

Graph model

Parts (A) and (B) of Figure 5.693 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 there is a constraint on each connected component of the final graph we also show the different connected components. Each of them corresponds to an equivalence class according to the arc constraint. The `same_modulo` constraint holds since:

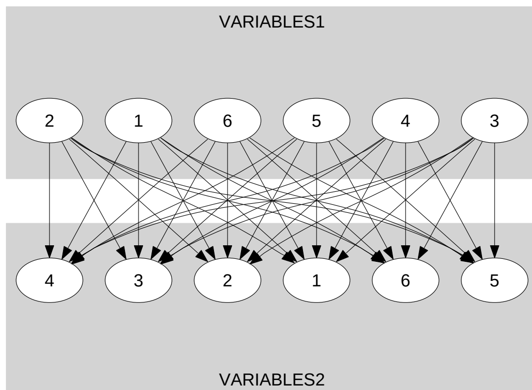
- Each connected component of the final graph has the same number of sources and of sinks.
- The number of sources of the final graph is equal to |`VARIABLES1`|.
- The number of sinks of the final graph is equal to |`VARIABLES2`|.

Signature

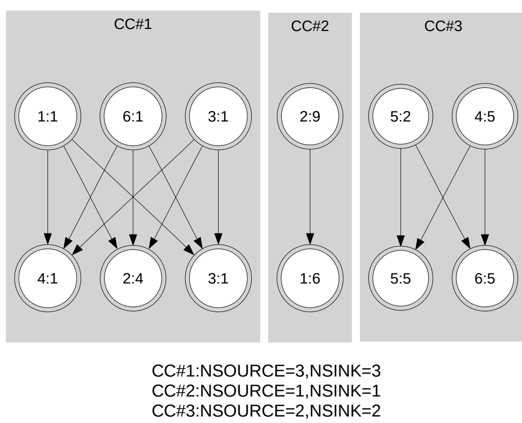
Since the initial graph contains only sources and sinks, and since isolated vertices are eliminated from the final graph, we make the following observations:

- Sources of the initial graph cannot become sinks of the final graph,
- Sinks of the initial graph cannot become sources of the final graph.

From the previous observations and since we use the $PRODUCT$ arc generator on the collections `VARIABLES1` and `VARIABLES2`, we have that the maximum number of sources and sinks of the final graph is respectively equal to |`VARIABLES1`| and |`VARIABLES2`|. Therefore we can rewrite **NSOURCE** = |`VARIABLES1`| to **NSOURCE** $\geq$ |`VARIABLES1`| and simplify $\underline{\overline{\mathbf{NSOURCE}}}$ to $\overline{\mathbf{NSOURCE}}$. In a similar way, we can rewrite **NSINK** = |`VARIABLES2`| to **NSINK** $\geq$ |`VARIABLES2`| and simplify $\underline{\overline{\mathbf{NSINK}}}$ to $\overline{\mathbf{NSINK}}$.



(A)



(B)

Figure 5.693: Initial and final graph of the `same_modulo` constraint