

5.64 common_modulo

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>common</code> .		
Constraint	<code>common_modulo(NCOMMON1, NCOMMON2, VARIABLES1, VARIABLES2, M)</code>		
Arguments	<pre> NCOMMON1 : dvar NCOMMON2 : dvar VARIABLES1 : collection(var-dvar) VARIABLES2 : collection(var-dvar) M : int </pre>		
Restrictions	<pre> NCOMMON1 ≥ 0 NCOMMON1 ≤ VARIABLES1 NCOMMON2 ≥ 0 NCOMMON2 ≤ VARIABLES2 required(VARIABLES1, var) required(VARIABLES2, var) M > 0 </pre>		
Purpose	<p>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.</p> <p>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.</p>		
Example	$\left(\begin{array}{l} 3, 4, \langle 0, 4, 0, 8 \rangle, \\ \text{var} - 7, \\ \text{var} - 5, \\ \langle \text{var} - 4, \\ \text{var} - 9, \\ \text{var} - 2, \\ \text{var} - 4 \rangle, 5 \end{array} \right)$		
	<p>In the example, the last argument $M = 5$ 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 <code>common_modulo</code> constraint holds since:</p> <ul style="list-style-type: none"> • Its first argument $NCOMMON1 = 3$ 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 $\text{NCOMMON2} = 4$ 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

```

|VARIABLES1| > 1
range(VARIABLES1.var) > 1
|VARIABLES2| > 1
range(VARIABLES2.var) > 1
M > 1
M < maxval(VARIABLES1.var)
M < maxval(VARIABLES2.var)

```

Symmetries

- Arguments are **permutable** w.r.t. permutation $(\text{NCOMMON1}, \text{NCOMMON2})$ $(\text{VARIABLES1}, \text{VARIABLES2})$ (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 .

See also

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

Keywords

characteristic of a constraint: modulo.

constraint arguments: constraint between two collections of variables.

final graph structure: acyclic, bipartite, no loop.

Arc input(s)	VARIABLES1 VARIABLES2
Arc generator	<i>PRODUCT</i> \mapsto <code>collection(variables1, variables2)</code>
Arc arity	2
Arc constraint(s)	$\text{variables1.var mod } M = \text{variables2.var mod } M$
Graph property(ies)	<ul style="list-style-type: none"> • NSOURCE = NCOMMON1 • NSINK = NCOMMON2
Graph class	<ul style="list-style-type: none"> • ACYCLIC • BIPARTITE • NO_LOOP

Graph model

Parts (A) and (B) of Figure 5.135 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.

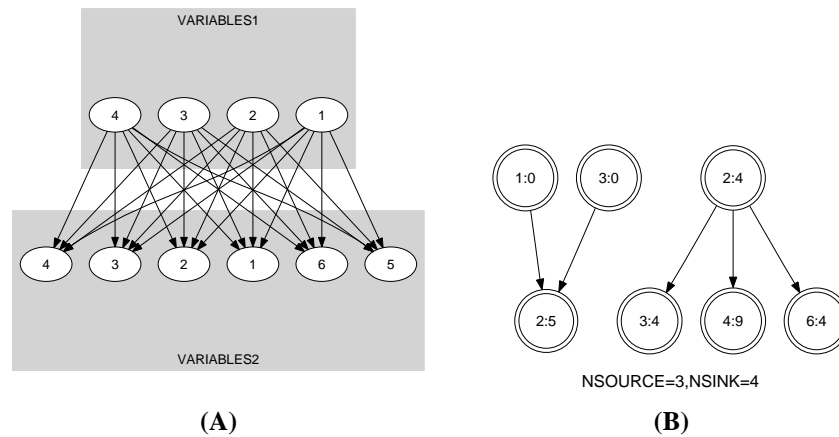


Figure 5.135: Initial and final graph of the `common_modulo` constraint

20030820

649