5.76   common_interval

Origin   Derived from common.

Constraint

$$\text{common\_interval} \begin{pmatrix} \text{NCOMMON1,} \\ \text{NCOMMON2,} \\ \text{VARIABLES1,} \\ \text{VARIABLES2,} \\ \text{SIZE\_INTERVAL} \end{pmatrix}$$

Arguments

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

Restrictions

$\text{NCOMMON1} \geq 0$
$\text{NCOMMON1} \leq |\text{VARIABLES1}|$
$\text{NCOMMON2} \geq 0$
$\text{NCOMMON2} \leq |\text{VARIABLES2}|$
required(VARIABLES1, var)
required(VARIABLES2, var)
$\text{SIZE\_INTERVAL} > 0$

Purpose

NCOMMON1 is the number of variables of the collection of variables VARIABLES1 taking a value in one of the intervals derived from the values assigned to the variables of the collection VARIABLES2: To each value v assigned to a variable of the collection VARIABLES2 we associate the interval $[\text{SIZE\_INTERVAL} \cdot \lfloor v/\text{SIZE\_INTERVAL} \rfloor, \text{SIZE\_INTERVAL} \cdot \lfloor v/\text{SIZE\_INTERVAL} \rfloor + \text{SIZE\_INTERVAL} - 1]$. NCOMMON2 is the number of variables of the collection of variables VARIABLES2 taking a value in one of the intervals derived from the values assigned to the variables of the collection VARIABLES1: To each value v assigned to a variable of the collection VARIABLES1 we associate the interval $[\text{SIZE\_INTERVAL} \cdot \lfloor v/\text{SIZE\_INTERVAL} \rfloor, \text{SIZE\_INTERVAL} \cdot \lfloor v/\text{SIZE\_INTERVAL} \rfloor + \text{SIZE\_INTERVAL} - 1]$.

Example

$(3, 2, \langle 8, 6, 6, 0 \rangle, \langle 7, 3, 3, 3, 3, 7 \rangle, 3)$

In the example, the last argument SIZE_INTERVAL $= 3$ defines the following family of intervals $[3 \cdot k, 3 \cdot k + 2]$, where k is an integer. As a consequence the items of collection $\langle 8, 6, 6, 0 \rangle$ respectively correspond to intervals $[6, 8]$, $[6, 8]$, $[6, 8]$ and $[0, 2]$. Similarly the items of collection $\langle 7, 3, 3, 3, 3, 7 \rangle$ respectively correspond to intervals $[6, 8]$, $[3, 5]$, $[3, 5]$, $[3, 5]$, $[3, 5]$, $[6, 8]$. The common_interval constraint holds since:

- Its first argument NCOMMON1 $= 3$ is the number of intervals associated with the items of collection $\langle 8, 6, 6, 0 \rangle$ that also correspond to intervals associated with $\langle 7, 3, 3, 3, 3, 7 \rangle$.

- Its second argument NCOMMON2 $= 2$ is the number of intervals associated with the items of collection $\langle 7, 3, 3, 3, 3, 7 \rangle$ that also correspond to intervals associated with $\langle 8, 6, 6, 0 \rangle$.

Typical

$|\text{VARIABLES1}| > 1$
$\text{range}(\text{VARIABLES1.var}) > 1$
$|\text{VARIABLES2}| > 1$
$\text{range}(\text{VARIABLES2.var}) > 1$
$\text{SIZE\_INTERVAL} > 1$
$\text{SIZE\_INTERVAL} < \text{range}(\text{VARIABLES1.var})$
$\text{SIZE\_INTERVAL} < \text{range}(\text{VARIABLES2.var})$

Symmetries

- Arguments are permutable w.r.t. permutation (NCOMMON1, NCOMMON2) (VARIABLES1, VARIABLES2) (SIZE_INTERVAL).
- Items of VARIABLES1 are permutable.
- Items of VARIABLES2 are permutable.
- An occurrence of a value of VARIABLES1.var that belongs to the k-th interval, of size SIZE_INTERVAL, can be replaced by any other value of the same interval.
- An occurrence of a value of VARIABLES2.var that belongs to the k-th interval, of size SIZE_INTERVAL, can be replaced by any other value of the same interval.

Arg. properties

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

See also

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

Keywords

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

final graph structure: acyclic, bipartite, no loop.

modelling: interval, functional dependency.

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

Graph model   Parts (A) and (B) of Figure 5.182 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 2 sinks the variables NCOMMON1 and NCOMMON2 are respectively
equal to 3 and 2. Note that the vertices corresponding to the variables that take values 0
or 3 were removed from the final graph since there is no arc for which the associated arc
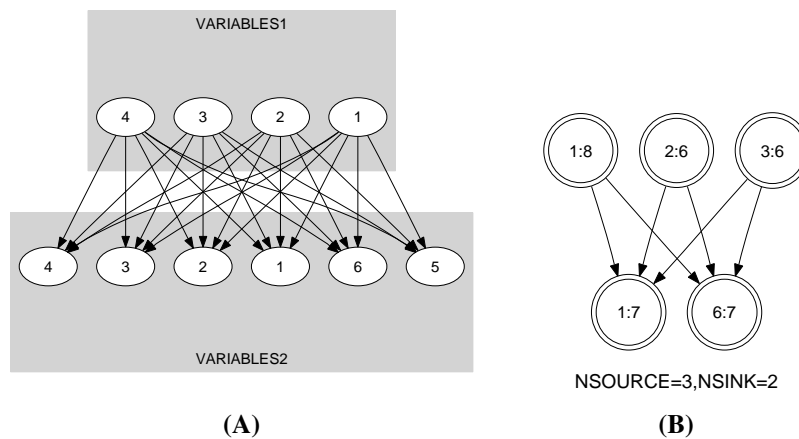constraint holds.

(A)   (B)

Figure 5.182: Initial and final graph of the common_interval constraint