

5.78 common_partition

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>common</code> .		
Constraint	$\text{common_partition} \left(\begin{array}{l} \text{NCOMMON1}, \\ \text{NCOMMON2}, \\ \text{VARIABLES1}, \\ \text{VARIABLES2}, \\ \text{PARTITIONS} \end{array} \right)$		
Type	VALUES : <code>collection(val-int)</code>		
Arguments	NCOMMON1 : <code>dvar</code> NCOMMON2 : <code>dvar</code> VARIABLES1 : <code>collection(var-dvar)</code> VARIABLES2 : <code>collection(var-dvar)</code> PARTITIONS : <code>collection(p - VALUES)</code>		
Restrictions	$ \text{VALUES} \geq 1$ <code>required(VALUES, val)</code> <code>distinct(VALUES, val)</code> $\text{NCOMMON1} \geq 0$ $\text{NCOMMON1} \leq \text{VARIABLES1} $ $\text{NCOMMON2} \geq 0$ $\text{NCOMMON2} \leq \text{VARIABLES2} $ <code>required(VARIABLES1, var)</code> <code>required(VARIABLES2, var)</code> <code>required(PARTITIONS, p)</code> $ \text{PARTITIONS} \geq 2$		
Purpose	<p>NCOMMON1 is the number of variables of the VARIABLES1 collection taking a value in a partition derived from the values assigned to the variables of VARIABLES2 and from PARTITIONS.</p> <p>NCOMMON2 is the number of variables of the VARIABLES2 collection taking a value in a partition derived from the values assigned to the variables of VARIABLES1 and from PARTITIONS.</p>		
Example	$\left(\begin{array}{l} 3, 4, \langle 2, 3, 6, 0 \rangle, \\ \langle 0, 6, 3, 3, 7, 1 \rangle, \\ \langle p - \langle 1, 3 \rangle, p - \langle 4 \rangle, p - \langle 2, 6 \rangle \rangle \end{array} \right)$ <p>In the example, the last argument PARTITIONS defines the partitions $p - \langle 1, 3 \rangle$, $p - \langle 4 \rangle$ and $p - \langle 2, 6 \rangle$. As a consequence the first three items of collection $\langle 2, 3, 6, 0 \rangle$ respectively correspond to the partitions $p - \langle 2, 6 \rangle$, $p - \langle 1, 3 \rangle$, and $p - \langle 2, 6 \rangle$. Similarly the items of collection $\langle 0, 6, 3, 3, 7, 1 \rangle$ (from which we remove items 0 and 7 since they do</p>		

not belong to any partition) respectively correspond to the partitions $p = \langle 2, 6 \rangle$, $p = \langle 1, 3 \rangle$, $p = \langle 1, 3 \rangle$, and $p = \langle 1, 3 \rangle$. The `common_partition` constraint holds since:

- Its first argument $\text{NCOMMON1} = 3$ is the number of partitions associated with the items of collection $\langle 2, 3, 6, 0 \rangle$ that also correspond to partitions associated with $\langle 0, 6, 3, 3, 7, 1 \rangle$.
- Its second argument $\text{NCOMMON2} = 4$ is the number of partitions associated with the items of collection $\langle 0, 6, 3, 3, 7, 1 \rangle$ that also correspond to partitions associated with $\langle 2, 3, 6, 0 \rangle$.

Typical

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

Symmetries

- Arguments are [permutable](#) w.r.t. permutation $(\text{NCOMMON1}, \text{NCOMMON2})$ $(\text{VARIABLES1}, \text{VARIABLES2})$ (PARTITIONS) .
- Items of `VARIABLES1` are [permutable](#).
- Items of `VARIABLES2` are [permutable](#).
- Items of `PARTITIONS` are [permutable](#).
- Items of `PARTITIONS.p` are [permutable](#).
- An occurrence of a value of `VARIABLES1.var` can be replaced by any other value that also belongs to the same partition of `PARTITIONS`.
- An occurrence of a value of `VARIABLES2.var` can be replaced by any other value that also belongs to the same partition of `PARTITIONS`.

Arg. properties

- [Functional dependency](#): `NCOMMON1` determined by `VARIABLES1`, `VARIABLES2` and `PARTITIONS`.
- [Functional dependency](#): `NCOMMON2` determined by `VARIABLES1`, `VARIABLES2` and `PARTITIONS`.

See also

[specialisation](#): `common(variable ∈ partition replaced by variable)`.

[used in graph description](#): `in_same_partition`.

Keywords

[characteristic of a constraint](#): `partition`.

[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	<i>PRODUCT</i> \mapsto collection(variables1,variables2)
Arc arity	2
Arc constraint(s)	in_same_partition(variables1.var,variables2.var,PARTITIONS)
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 5.201 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 0 or 7 were removed from the final graph since there is no arc for which the associated in_same_partition constraint holds.

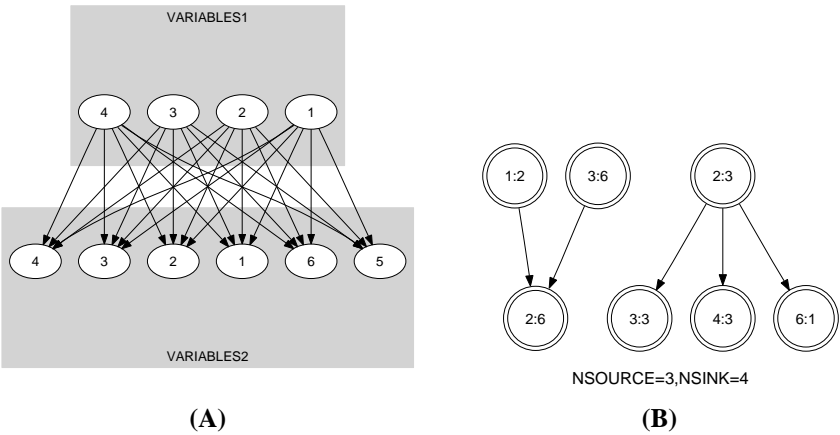


Figure 5.201: Initial and final graph of the common_partition constraint

20030820

867