

5.211 `k_same_partition`

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>same_partition</code> and from <code>k_same</code> .		
Constraint	<code>k_same_partition</code> (SETS, PARTITIONS)		
Types	VARIABLES : <code>collection</code> (var-dvar) VALUES : <code>collection</code> (val-int)		
Arguments	SETS : <code>collection</code> (set – VARIABLES) PARTITIONS : <code>collection</code> (p – VALUES)		
Restrictions	<code>required</code> (VARIABLES, var) $ \text{VARIABLES}  \geq 1$ $ \text{VALUES}  \geq 1$ <code>required</code> (VALUES, val) <code>distinct</code> (VALUES, val) <code>required</code> (SETS, set) $ \text{SETS}  > 1$ <code>same_size</code> (SETS, set) <code>required</code> (PARTITIONS, p) $ \text{PARTITIONS}  \geq 2$		
Purpose	Given a collection of $ \text{SETS} $ sets, each containing the same number of domain variables, the <code>k_same_partition</code> constraint forces a <code>same_partition</code> constraint between each pair of consecutive sets.		
Example	$\left( \left\langle \begin{array}{l} \text{set} - \langle 1, 2, 6, 3, 1, 2 \rangle, \\ \text{set} - \langle 6, 6, 2, 3, 1, 3 \rangle, \\ \text{set} - \langle 2, 2, 2, 1, 1, 1 \rangle \end{array} \right\rangle, \right. \\ \left. \langle p - \langle 1, 3 \rangle, p - \langle 4 \rangle, p - \langle 2, 6 \rangle \rangle \right)$ <p>The first argument SETS of the <code>k_same_partition</code> constraint corresponds to 3 collections of variables, while the second argument PARTITIONS defines the 3 sets of values {1, 3}, {4} and {2, 6}. The <code>k_same_partition</code> constraint holds since:</p> <ul style="list-style-type: none"><li>• The first and second collections of variables are assigned 3 values in the {1, 3} as well as 3 values in {2, 6}.</li><li>• The second and third collections of variables are also assigned 3 values in the {1, 3} as well as 3 values in {2, 6}.</li></ul>		
Typical	$ \text{VARIABLES}  > 1$		

**Symmetries**

- Items of SETS are [permutable](#).
- Items of SETS.set are [permutable](#).
- Items of PARTITIONS are [permutable](#).
- Items of PARTITIONS.p are [permutable](#).
- An occurrence of a value of SETS.set.var can be replaced by any other value that also belongs to the same partition of PARTITIONS.

**Arg. properties**

[Contractible](#) wrt. SETS.

**See also**

**common keyword:** [k\\_same](#) (*system of constraints*).

**implies:** [k\\_used\\_by\\_partition](#).

**part of system of constraints:** [same\\_partition](#).

**used in graph description:** [same\\_partition](#).

**Keywords**

**characteristic of a constraint:** [sort based reformulation](#), [partition](#).

**combinatorial object:** [permutation](#).

**constraint type:** [system of constraints](#), [decomposition](#).

Arc input(s)	SETS
Arc generator	$\text{PATH} \mapsto \text{collection}(\text{set1}, \text{set2})$
Arc arity	2
Arc constraint(s)	<code>same_partition</code> (set1.set, set2.set, PARTITIONS)
Graph property(ies)	$\text{NARC} =  \text{SETS}  - 1$

**Graph model** Parts (A) and (B) of Figure 5.483 respectively show the initial and final graph associated with the **Example** slot. To each vertex corresponds a collection of variables, while to each arc corresponds a `same_partition` constraint.

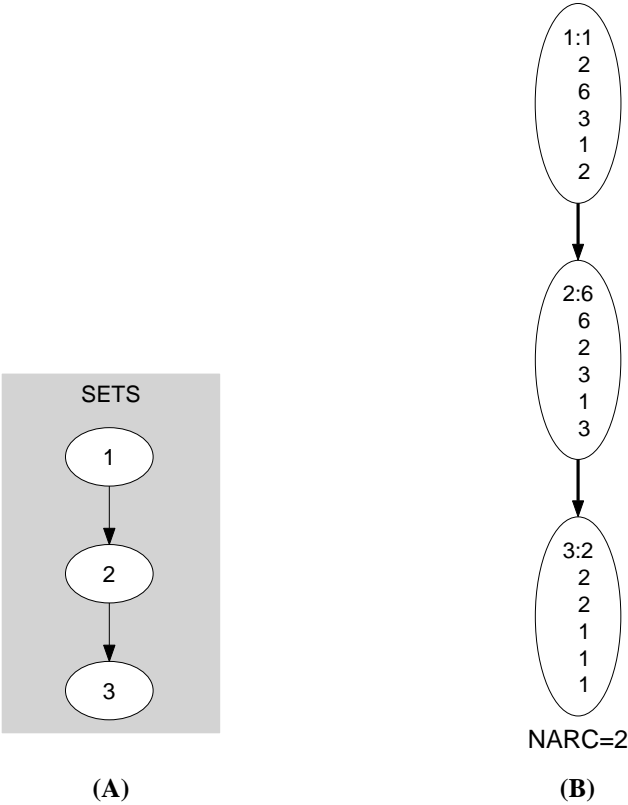


Figure 5.483: Initial and final graph of the `k_same_partition` constraint

