

5.12 alldifferent_on_intersection

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Derived from common and alldifferent .			
Constraint	<code>alldifferent_on_intersection(VARIABLES1, VARIABLES2)</code>			
Synonyms	<code>alldiff_on_intersection</code> , <code>alldistinct_on_intersection</code> .			
Arguments	VARIABLES1 : collection (<code>var-dvar</code>) VARIABLES2 : collection (<code>var-dvar</code>)			
Restrictions	required (VARIABLES1, <code>var</code>) required (VARIABLES2, <code>var</code>)			
Purpose	<div style="border: 1px solid pink; padding: 5px;"> The values that both occur in the VARIABLES1 and VARIABLES2 collections have only one occurrence. </div>			
Example	<div style="border: 1px solid blue; padding: 10px; display: inline-block;"> $\left(\begin{array}{c} \langle 5, 9, 1, 5 \rangle, \\ \text{var} - 2, \\ \text{var} - 1, \\ \langle \text{var} - 6, \rangle \\ \text{var} - 9, \\ \text{var} - 6, \\ \text{var} - 2 \end{array} \right)$ </div> <p>The <code>alldifferent_on_intersection</code> constraint holds since the values 9 and 1 that both occur in $\langle 5, 9, 1, 5 \rangle$ as well as in $\langle 2, 1, 6, 9, 6, 2 \rangle$ have exactly one occurrence in each collection.</p>			
Typical	$ VARIABLES1 > 1$ $ VARIABLES2 > 1$			
Symmetries	<ul style="list-style-type: none"> Arguments are permutable w.r.t. permutation (VARIABLES1, VARIABLES2). Items of VARIABLES1 are permutable. Items of VARIABLES2 are permutable. All occurrences of two distinct values in <code>VARIABLES1.var</code> or <code>VARIABLES2.var</code> can be swapped; all occurrences of a value in <code>VARIABLES1.var</code> or <code>VARIABLES2.var</code> can be renamed to any unused value. 			
See also	common keyword: common , nvalue_on_intersection (<i>constraint on the intersection</i>). implied by: disjoint . implies: same_intersection . root concept: alldifferent .			

Keywords

characteristic of a constraint: all different, automaton, automaton with array of counters.

constraint arguments: constraint between two collections of variables.

constraint type: constraint on the intersection, value constraint.

final graph structure: connected component, acyclic, bipartite, no loop.

Arc input(s)	VARIABLES1 VARIABLE2
Arc generator	<i>PRODUCT</i> \mapsto <code>collection(variables1, variables2)</code>
Arc arity	2
Arc constraint(s)	<code>variables1.var = variables2.var</code>
Graph property(ies)	<u>MAX_NCC</u> \leq 2
Graph class	<ul style="list-style-type: none"> • <u>ACYCLIC</u> • <u>BIPARTITE</u> • <u>NO_LOOP</u>

Graph model

Parts (A) and (B) of Figure 5.16 respectively show the initial and final graph associated with the **Example** slot. Since we use the MAX_NCC graph property we show one of the largest connected component of the final graph. The `alldifferent_on_intersection` constraint holds since each connected component has at most two vertices. Note that all the vertices corresponding to the variables that take values 5, 2 or 6 were removed from the final graph since there is no arc for which the associated equality constraint holds.

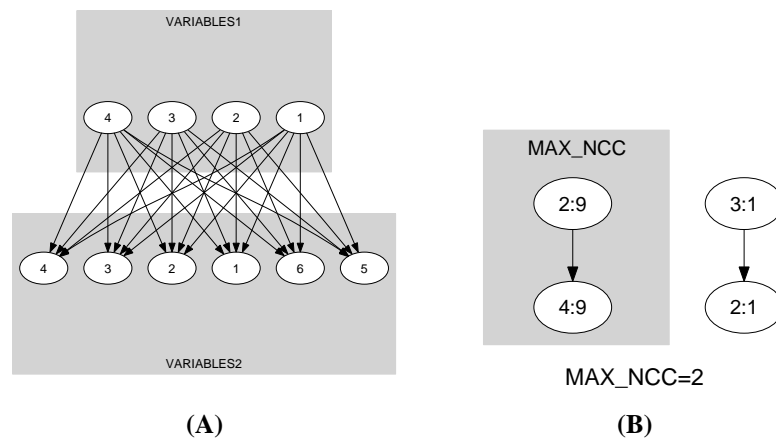


Figure 5.16: Initial and final graph of the `alldifferent_on_intersection` constraint

Automaton

Figure 5.17 depicts the automaton associated with the `alldifferent_on_intersection` constraint. To each variable $VAR1_i$ of the collection `VARIABLES1` corresponds a signature variable S_i that is equal to 0. To each variable $VAR2_i$ of the collection `VARIABLES2` corresponds a signature variable $S_{i+|VARIABLES1|}$ that is equal to 1. The automaton first counts the number of occurrences of each value assigned to the variables of the `VARIABLES1` collection. It then counts the number of occurrences of each value assigned to the variables of the `VARIABLES2` collection. Finally, the automaton imposes that each value is not taken by two variables of both collections.

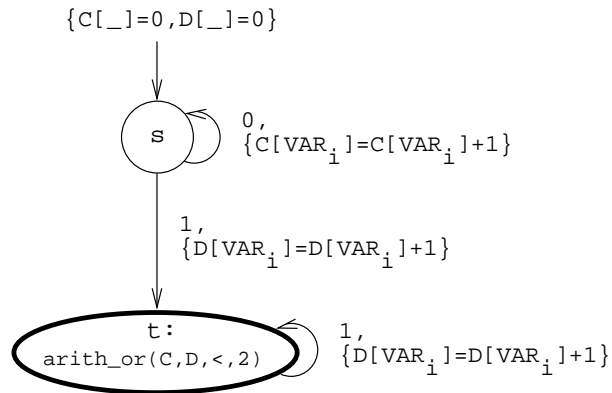


Figure 5.17: Automaton of the `alldifferent_on_intersection` constraint