

5.289 `nvalues_except_0`

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>nvalues</code> .		
Constraint	<code>nvalues_except_0(VARIABLES, RELOP, LIMIT)</code>		
Arguments	VARIABLES : <code>collection(var-dvar)</code> RELOP : <code>atom</code> LIMIT : <code>dvar</code>		
Restrictions	<code>required(VARIABLES, var)</code> RELOP \in [=, \neq , <, \geq , >, \leq]		
Purpose	<div style="border: 1px solid pink; padding: 5px;"> Let N be the number of distinct values, different from 0, assigned to the variables of the VARIABLES collection. Enforce condition N RELOP LIMIT to hold. </div>		
Example	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> $((4, 5, 5, 4, 0, 1), =, 3)$ </div> <p>The <code>nvalues_except_0</code> constraint holds since the number of distinct values, different from 0, occurring within the collection $\langle 4, 5, 5, 4, 0, 1 \rangle$ is equal (i.e., RELOP is set to $=$) to its third argument $LIMIT = 3$.</p>		
Typical	$ VARIABLES > 1$ $LIMIT > 1$ $LIMIT < VARIABLES $ <code>atleast(1, VARIABLES, 0)</code> RELOP \in [=, <, \geq , >, \leq]		
Symmetries	<ul style="list-style-type: none"> • Items of VARIABLES are permutable. • All occurrences of two distinct values of VARIABLES.var that are both different from 0 can be swapped; all occurrences of a value of VARIABLES.var that is different from 0 can be renamed to any unused value that is also different from 0. 		
Arg. properties	<ul style="list-style-type: none"> • Contractible wrt. VARIABLES when RELOP \in [<, \leq]. • Extensible wrt. VARIABLES when RELOP \in [\geq, >]. 		
Reformulation	The <code>nvalues_except_0($\langle V_1, V_2, \dots, V_{ VARIABLES } \rangle$, RELOP, LIMIT)</code> constraint can be expressed in term of the conjunction <code>nvalue(NV1, $\langle 0, V_1, V_2, \dots, V_{ VARIABLES } \rangle$) \wedge NV1 - 1 RELOP LIMIT</code> .		
Used in	<code>cycle_or_accessibility</code> .		
See also	common keyword: <code>assign_and_nvalues</code> (<i>number of distinct values</i>), <code>nvalue</code> , <code>nvalues</code> (<i>counting constraint, number of distinct values</i>).		

Keywords

characteristic of a constraint: joker value.

constraint type: counting constraint, value partitioning constraint.

final graph structure: strongly connected component.

modelling: number of distinct values.

Arc input(s)	VARIABLES
Arc generator	<i>CLIQUE</i> \mapsto <code>collection(variables1, variables2)</code>
Arc arity	2
Arc constraint(s)	<ul style="list-style-type: none"> • <code>variables1.var</code> \neq 0 • <code>variables1.var</code> = <code>variables2.var</code>
Graph property(ies)	<u>NSCC</u> RELOP LIMIT

Graph model

Parts (A) and (B) of Figure 5.603 respectively show the initial and final graph associated with the **Example** slot. Since we use the NSCC graph property we show the different strongly connected components of the final graph. Each strongly connected component corresponds to a value distinct from 0 that is assigned to some variables of the VARIABLES collection. Beside value 0, the 3 following values 1, 4 and 5 are assigned to the variables of the VARIABLES collection.

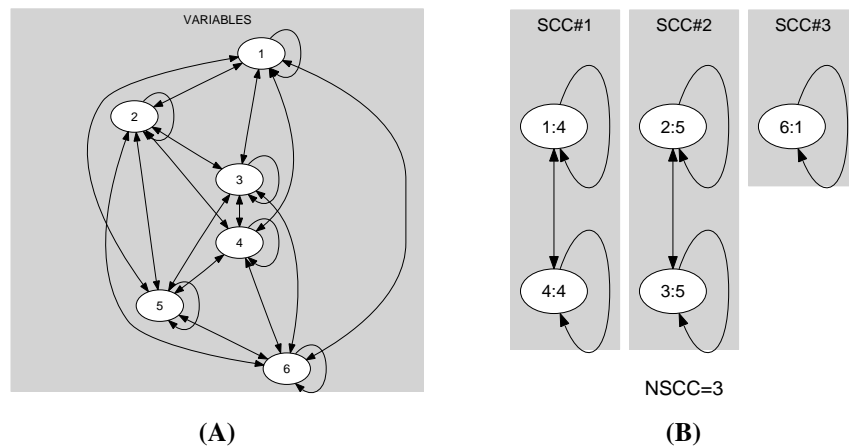


Figure 5.603: Initial and final graph of the `nvalues_except_0` constraint

20030820

1853