

## 5.277 ninterval

	DESCRIPTION	LINKS	GRAPH
<b>Origin</b>	Derived from <a href="#">nvalue</a> .		
<b>Constraint</b>	<code>ninterval(NVAL, VARIABLES, SIZE_INTERVAL)</code>		
<b>Arguments</b>	NVAL : <a href="#">dvar</a> VARIABLES : <a href="#">collection</a> (var-dvar) SIZE_INTERVAL : <a href="#">int</a>		
<b>Restrictions</b>	$NVAL \geq \min(1,  VARIABLES )$ $NVAL \leq  VARIABLES $ <a href="#">required</a> (VARIABLES, var) $SIZE\_INTERVAL > 0$		
<b>Purpose</b>	Consider the intervals of the form $[SIZE\_INTERVAL \cdot k, SIZE\_INTERVAL \cdot k + SIZE\_INTERVAL - 1]$ where $k$ is an integer. NVAL is the number of intervals for which at least one value is assigned to at least one variable of the collection VARIABLES.		
<b>Example</b>	$(2, \langle 3, 1, 9, 1, 9 \rangle, 4)$		
	In the example, the third argument <code>SIZE_INTERVAL = 4</code> defines the following family of intervals $[4 \cdot k, 4 \cdot k + 3]$ , where $k$ is an integer. Values 3, 1, 9, 1 and 9 are respectively located within intervals $[0, 3]$ , $[0, 3]$ , $[8, 11]$ , $[0, 3]$ and $[8, 11]$ . Since we only use the two intervals $[0, 3]$ and $[8, 11]$ the first argument of the <code>ninterval</code> constraint is set to value 2.		
<b>Typical</b>	$NVAL > 1$ $NVAL <  VARIABLES $ $SIZE\_INTERVAL > 1$ $SIZE\_INTERVAL < \text{range}(VARIABLES.var)$ $(\text{nval}(VARIABLES.var) + SIZE\_INTERVAL - 1) / SIZE\_INTERVAL < NVAL$		
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>Items of VARIABLES are <a href="#">permutable</a>.</li> <li>An occurrence of a value of <code>VARIABLES.var</code> that belongs to the <math>k</math>-th interval, of size <code>SIZE_INTERVAL</code>, can be <a href="#">replaced</a> by any other value of the same interval.</li> </ul>		
<b>Arg. properties</b>	<ul style="list-style-type: none"> <li><a href="#">Functional dependency</a>: NVAL determined by VARIABLES and SIZE_INTERVAL.</li> <li><a href="#">Contractible</a> wrt. VARIABLES when <math>NVAL = 1</math> and <math> VARIABLES  &gt; 0</math>.</li> <li><a href="#">Contractible</a> wrt. VARIABLES when <math>NVAL =  VARIABLES </math>.</li> </ul>		
<b>Usage</b>	The <code>ninterval</code> constraint is useful for counting the number of actually used periods, no matter how many time each period is used. A period can for example stand for a hour or for a day.		

**Algorithm** [27, 40].

**See also** **related:** `nclass` (variable/constant replaced by variable  $\in$  partition),  
`nequivalence` (variable/constant replaced by variable mod constant),  
`npair` (variable/constant replaced by pair of variables).  
**specialisation:** `nvalue` (variable/constant replaced by variable).

**Keywords** **constraint arguments:** pure functional dependency.  
**constraint type:** counting constraint, value partitioning constraint.  
**final graph structure:** strongly connected component, equivalence.  
**modelling:** number of distinct equivalence classes, interval, functional dependency.

<b>Arc input(s)</b>	VARIABLES
<b>Arc generator</b>	<i>CLIQUE</i> $\mapsto$ <code>collection(variables1, variables2)</code>
<b>Arc arity</b>	2
<b>Arc constraint(s)</b>	$\text{variables1.var}/\text{SIZE\_INTERVAL} = \text{variables2.var}/\text{SIZE\_INTERVAL}$
<b>Graph property(ies)</b>	<u>NSCC</u> = NVAL

**Graph model**

Parts (A) and (B) of Figure 5.579 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 those values of an interval that are assigned to some variables of the **VARIABLES** collection. The values 1, 3 and the value 9, which respectively correspond to intervals  $[0, 3]$  and  $[8, 11]$ , are assigned to the variables of the **VARIABLES** collection.

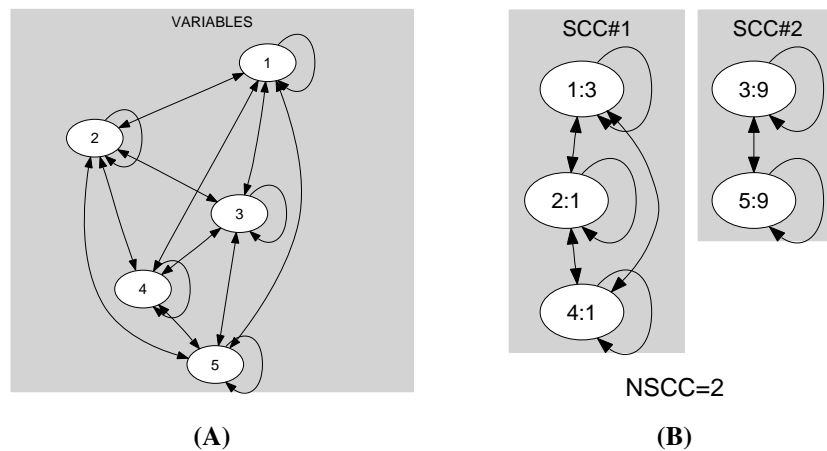


Figure 5.579: Initial and final graph of the ninterval constraint

20030820

1787