

## 5.226 lex\_chain\_lesseq

	DESCRIPTION	LINKS	GRAPH
<b>Origin</b>	[95]		
<b>Constraint</b>	lex_chain_lesseq(VECTORS)		
<b>Usual name</b>	lex_chain		
<b>Type</b>	VECTOR : collection(var-dvar)		
<b>Argument</b>	VECTORS : collection(vec - VECTOR)		
<b>Restrictions</b>	$ \text{VECTOR}  \geq 1$ <a href="#">required</a> (VECTOR, var) <a href="#">required</a> (VECTORS, vec) <a href="#">same_size</a> (VECTORS, vec)		
<b>Purpose</b>	<p>For each pair of consecutive vectors <math>\text{VECTOR}_i</math> and <math>\text{VECTOR}_{i+1}</math> of the VECTORS collection we have that <math>\text{VECTOR}_i</math> is lexicographically less than or equal to <math>\text{VECTOR}_{i+1}</math>. Given two vectors, <math>\vec{X}</math> and <math>\vec{Y}</math> of <math>n</math> components, <math>\langle X_0, \dots, X_{n-1} \rangle</math> and <math>\langle Y_0, \dots, Y_{n-1} \rangle</math>, <math>\vec{X}</math> is <i>lexicographically less than or equal to</i> <math>\vec{Y}</math> if and only if <math>n = 0</math> or <math>X_0 &lt; Y_0</math> or <math>X_0 = Y_0</math> and <math>\langle X_1, \dots, X_{n-1} \rangle</math> is lexicographically less than or equal to <math>\langle Y_1, \dots, Y_{n-1} \rangle</math>.</p>		
<b>Example</b>	$((\text{vec} - \langle 5, 2, 3, 9 \rangle, \text{vec} - \langle 5, 2, 6, 2 \rangle, \text{vec} - \langle 5, 2, 6, 2 \rangle))$		
	<p>The <code>lex_chain_lesseq</code> constraint holds since:</p> <ul style="list-style-type: none"> <li>• The first vector <math>\langle 5, 2, 3, 9 \rangle</math> of the VECTORS collection is lexicographically less than or equal to the second vector <math>\langle 5, 2, 6, 2 \rangle</math> of the VECTORS collection.</li> <li>• The second vector <math>\langle 5, 2, 6, 2 \rangle</math> of the VECTORS collection is lexicographically less than or equal to the third vector <math>\langle 5, 2, 6, 2 \rangle</math> of the VECTORS collection.</li> </ul>		
<b>Typical</b>	$ \text{VECTOR}  > 1$ $ \text{VECTORS}  > 1$		
<b>Arg. properties</b>	<ul style="list-style-type: none"> <li>• <a href="#">Contractible</a> wrt. VECTORS.</li> <li>• <a href="#">Suffix-contractible</a> wrt. VECTORS.vec (<i>remove items from same position</i>).</li> </ul>		
<b>Usage</b>	<p>This constraint was motivated for breaking symmetry: more precisely when one wants to lexicographically order the consecutive columns of a matrix of decision variables. A further motivation is that using a set of lexicographic ordering constraints between two vectors does usually not allow to come up with a complete pruning.</p>		

<b>Algorithm</b>	<p>A filtering algorithm achieving <a href="#">arc-consistency</a> for a chain of lexicographical ordering constraints is presented in [95].</p> <p>Six different ways of integrating a chain of lexicographical ordering constraints within non-overlapping constraints like <a href="#">diffn</a> or <a href="#">geost</a> and within their corresponding necessary condition like the <a href="#">cumulative</a> constraint are shown in [3].</p>
<b>Systems</b>	<a href="#">lexChainEq</a> in <b>Choco</b> , <a href="#">lex_chain</a> in <b>SICStus</b> .
<b>See also</b>	<p><b>common keyword:</b> <a href="#">allperm</a> (<i>lexicographic order</i>), <a href="#">geost</a> (<i>symmetry, lexicographic ordering on the origins of tasks, rectangles, ...</i>), <a href="#">lex_between</a>, <a href="#">lex_greater</a>, <a href="#">lex_greatereq</a>, <a href="#">lex_less</a> (<i>lexicographic order</i>).</p> <p><b>implied by:</b> <a href="#">lex2</a> (<i>columns lex ordering imposed by constraint <a href="#">lex2</a> removed</i>), <a href="#">lex_chain_less</a> (<i>non-strict order implied by strict order</i>), <a href="#">ordered_atleast_nvector</a> (NVEC of constraint <a href="#">ordered_atleast_nvector</a> removed), <a href="#">ordered_atmost_nvector</a> (NVEC of constraint <a href="#">ordered_atmost_nvector</a> removed), <a href="#">ordered_nvector</a> (NVEC of constraint <a href="#">ordered_nvector</a> removed).</p> <p><b>part of system of constraints:</b> <a href="#">lex_lesseq</a>.</p> <p><b>related:</b> <a href="#">cumulative</a>, <a href="#">diffn</a> (<i>lexicographic ordering on the origins of tasks, rectangles, ...</i>).</p> <p><b>system of constraints:</b> <a href="#">lex2</a>.</p> <p><b>used in graph description:</b> <a href="#">lex_lesseq</a>.</p>
<b>Keywords</b>	<p><b>characteristic of a constraint:</b> <a href="#">vector</a>.</p> <p><b>constraint type:</b> <a href="#">system of constraints</a>, <a href="#">decomposition</a>, <a href="#">order constraint</a>.</p> <p><b>filtering:</b> <a href="#">arc-consistency</a>.</p> <p><b>heuristics:</b> <a href="#">heuristics and lexicographical ordering</a>.</p> <p><b>symmetry:</b> <a href="#">symmetry</a>, <a href="#">matrix symmetry</a>, <a href="#">lexicographic order</a>.</p>

<b>Arc input(s)</b>	VECTORS
<b>Arc generator</b>	$\text{PATH} \mapsto \text{collection}(\text{vectors1}, \text{vectors2})$
<b>Arc arity</b>	2
<b>Arc constraint(s)</b>	$\text{lex\_lesseq}(\text{vectors1.vec}, \text{vectors2.vec})$
<b>Graph property(ies)</b>	$\text{NARC} =  \text{VECTORS}  - 1$

**Graph model**

Parts (A) and (B) of Figure 5.481 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold. The  $\text{lex\_chain\_lesseq}$  constraint holds since all the arc constraints of the initial graph are satisfied.

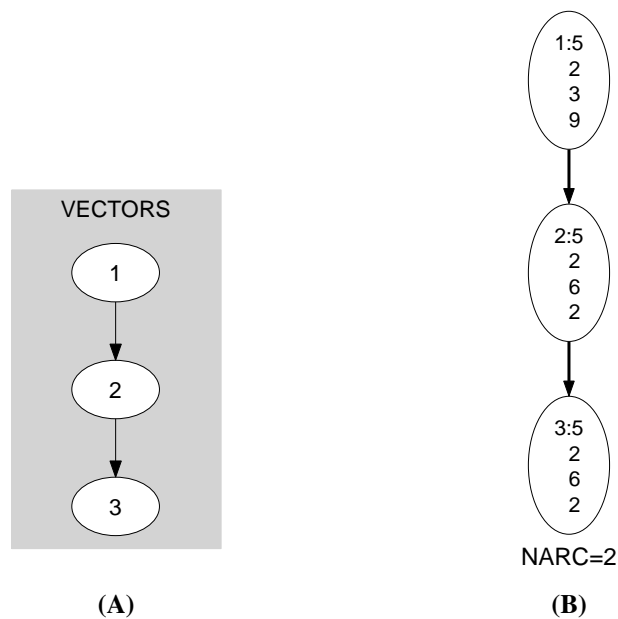


Figure 5.481: Initial and final graph of the  $\text{lex\_chain\_lesseq}$  constraint

**Signature**

Since we use the  $\text{PATH}$  arc generator on the **VECTORS** collection the number of arcs of the initial graph is equal to  $|\text{VECTORS}| - 1$ . For this reason we can rewrite  $\text{NARC} = |\text{VECTORS}| - 1$  to  $\text{NARC} \geq |\text{VECTORS}| - 1$  and simplify  $\overline{\text{NARC}}$  to  $\text{NARC}$ .

20030820

1543