

5.225 `lex_chain_less`

	DESCRIPTION	LINKS	GRAPH
Origin	[95]		
Constraint	<code>lex_chain_less(VECTORS)</code>		
Usual name	<code>lex_chain</code>		
Type	<code>VECTOR : collection(var-dvar)</code>		
Argument	<code>VECTORS : collection(vec - VECTOR)</code>		
Restrictions	$ \text{VECTOR} \geq 1$ <code>required(VECTOR, var)</code> <code>required(VECTORS, vec)</code> <code>same_size(VECTORS, vec)</code>		
Purpose	<p>For each pair of consecutive vectors VECTOR_i and VECTOR_{i+1} of the <code>VECTORS</code> collection we have that VECTOR_i is lexicographically strictly less than VECTOR_{i+1}. Given two vectors, \vec{X} and \vec{Y} of n components, $\langle X_0, \dots, X_{n-1} \rangle$ and $\langle Y_0, \dots, Y_{n-1} \rangle$, \vec{X} is <i>lexicographically strictly less than</i> \vec{Y} if and only if $X_0 < Y_0$ or $X_0 = Y_0$ and $\langle X_1, \dots, X_{n-1} \rangle$ is lexicographically strictly less than $\langle Y_1, \dots, Y_{n-1} \rangle$.</p>		
Example	<code>((vec - <5, 2, 3, 9>, vec - <5, 2, 6, 2>, vec - <5, 2, 6, 3>))</code>		
	<p>The <code>lex_chain_less</code> constraint holds since:</p> <ul style="list-style-type: none"> • The first vector $\langle 5, 2, 3, 9 \rangle$ of the <code>VECTORS</code> collection is lexicographically strictly less than the second vector $\langle 5, 2, 6, 2 \rangle$ of the <code>VECTORS</code> collection. • The second vector $\langle 5, 2, 6, 2 \rangle$ of the <code>VECTORS</code> collection is lexicographically strictly less than the third vector $\langle 5, 2, 6, 3 \rangle$ of the <code>VECTORS</code> collection. 		
Typical	$ \text{VECTOR} > 1$ $ \text{VECTORS} > 1$		
Arg. properties	<ul style="list-style-type: none"> • Contractible wrt. <code>VECTORS</code>. • Suffix-extensible wrt. <code>VECTORS.vec</code> (<i>add items at same position</i>). 		
Usage	<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>		

Algorithm	<p>A filtering algorithm achieving arc-consistency 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 diffn or geost and within their corresponding necessary condition like the cumulative constraint are shown in [3].</p>
Systems	<p>lexChain in Choco, lex_chain in SICStus.</p>
See also	<p>common keyword: geost (<i>symmetry, lexicographic ordering on the origins of tasks, rectangles, ...</i>), lex_between, lex_greater, lex_greatereq, lex_lesseq (<i>lexicographic order</i>).</p> <p>implied by: strict_lex2.</p> <p>implies: lex_alldifferent, lex_chain_lesseq.</p> <p>part of system of constraints: lex_less.</p> <p>related: cumulative, diffn (<i>lexicographic ordering on the origins of tasks, rectangles, ...</i>).</p> <p>system of constraints: strict_lex2.</p> <p>used in graph description: lex_less.</p>
Keywords	<p>application area: floor planning problem.</p> <p>characteristic of a constraint: vector.</p> <p>constraint type: decomposition, order constraint, system of constraints.</p> <p>filtering: arc-consistency.</p> <p>heuristics: heuristics and lexicographical ordering.</p> <p>modelling: degree of diversity of a set of solutions.</p> <p>modelling exercises: degree of diversity of a set of solutions.</p> <p>symmetry: symmetry, matrix symmetry, lexicographic order.</p>

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

Graph model

Parts (A) and (B) of Figure 5.480 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 lex_chain_less constraint holds since all the arc constraints of the initial graph are satisfied.

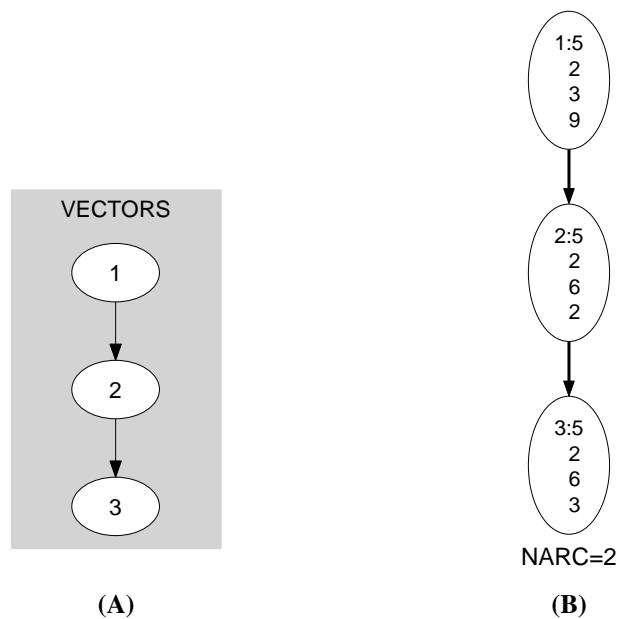


Figure 5.480: Initial and final graph of the lex_chain_less constraint

Signature

Since we use the 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 NARC .

20030820

1539