

5.192 `lex_between`

	DESCRIPTION	LINKS	AUTOMATON
Origin	[86]		
Constraint	<code>lex_between(LOWER_BOUND, VECTOR, UPPER_BOUND)</code>		
Synonym	<code>between.</code>		
Arguments	<pre>LOWER_BOUND : collection(var-int) VECTOR : collection(var-dvar) UPPER_BOUND : collection(var-int)</pre>		
Restrictions	<pre>required(LOWER_BOUND, var) required(VECTOR, var) required(UPPER_BOUND, var) LOWER_BOUND = VECTOR UPPER_BOUND = VECTOR lex_lesseq(LOWER_BOUND, VECTOR) lex_lesseq(VECTOR, UPPER_BOUND)</pre>		
Purpose	The vector <code>VECTOR</code> is lexicographically greater than or equal to the fixed vector <code>LOWER_BOUND</code> and lexicographically smaller than or equal to the fixed vector <code>UPPER_BOUND</code> .		
Example	$\left(\begin{array}{l} \langle 5, 2, 3, 9 \rangle, \\ \langle 5, 2, 6, 2 \rangle, \\ \langle 5, 2, 6, 3 \rangle \end{array} \right)$		
	The <code>lex_between</code> constraint holds since:		
	<ul style="list-style-type: none"> • The vector <code>VECTOR</code> = $\langle 5, 2, 6, 2 \rangle$ is greater than or equal to the vector <code>LOWER_BOUND</code> = $\langle 5, 2, 3, 9 \rangle$. • The vector <code>VECTOR</code> = $\langle 5, 2, 6, 2 \rangle$ is less than or equal to the vector <code>UPPER_BOUND</code> = $\langle 5, 2, 6, 3 \rangle$. 		
Symmetries	<ul style="list-style-type: none"> • <code>LOWER_BOUND.var</code> can be decreased. • <code>UPPER_BOUND.var</code> can be increased. 		
Usage	This constraint does usually not occur explicitly in practice. However it shows up indirectly in the context of the <code>lex_chain_less</code> and the <code>lex_chain_lesseq</code> constraints: in order to have a complete filtering algorithm for the <code>lex_chain_less</code> and the <code>lex_chain_lesseq</code> constraints one has to come up with a complete filtering algorithm for the <code>lex_between</code> constraint. The reason is that the <code>lex_chain_less</code> as well as the <code>lex_chain_lesseq</code> constraints both compute feasible lower and upper bounds for each vector they mention. Therefore one ends up with a <code>lex_between</code> constraint for each vector of the <code>lex_chain_less</code> and <code>lex_chain_lesseq</code> constraints.		

Algorithm	[86].
Reformulation	The <code>lex_between(LOWER_BOUND, VECTORS, UPPER_BOUND)</code> constraint can be expressed as the conjunction <code>lex_lesseq(LOWER_BOUND, VECTORS) ^ lex_lesseq(VECTORS, UPPER_BOUND)</code> .
Systems	<code>lexChainEq</code> in Choco , <code>lex_chain</code> in SICStus .
See also	common keyword: <code>lex_chain_less</code> , <code>lex_chain_lesseq</code> , <code>lex_greater</code> , <code>lex_greatereq</code> , <code>lex_less</code> (<i>lexicographic order</i>). part of system of constraints: <code>lex_lesseq</code> .
Keywords	characteristic of a constraint: <code>vector</code> , <code>automaton</code> , <code>automaton without counters</code> , <code>reified automaton constraint</code> . constraint network structure: <code>Berge-acyclic constraint network</code> . constraint type: <code>order constraint</code> , <code>system of constraints</code> . filtering: <code>arc-consistency</code> . symmetry: <code>symmetry</code> , <code>lexicographic order</code> .

Automaton

Figure 5.373 depicts the automaton associated with the `lex_between` constraint. Let L_i , V_i and U_i respectively be the `var` attributes of the i^{th} items of the `LOWER_BOUND`, the `VECTOR` and the `UPPER_BOUND` collections. To each triple (L_i, V_i, U_i) corresponds a signature variable S_i as well as the following signature constraint:

- $(L_i < V_i) \wedge (V_i < U_i) \Leftrightarrow S_i = 0 \wedge$
- $(L_i < V_i) \wedge (V_i = U_i) \Leftrightarrow S_i = 1 \wedge$
- $(L_i < V_i) \wedge (V_i > U_i) \Leftrightarrow S_i = 2 \wedge$
- $(L_i = V_i) \wedge (V_i < U_i) \Leftrightarrow S_i = 3 \wedge$
- $(L_i = V_i) \wedge (V_i = U_i) \Leftrightarrow S_i = 4 \wedge$
- $(L_i = V_i) \wedge (V_i > U_i) \Leftrightarrow S_i = 5 \wedge$
- $(L_i > V_i) \wedge (V_i < U_i) \Leftrightarrow S_i = 6 \wedge$
- $(L_i > V_i) \wedge (V_i = U_i) \Leftrightarrow S_i = 7 \wedge$
- $(L_i > V_i) \wedge (V_i > U_i) \Leftrightarrow S_i = 8.$

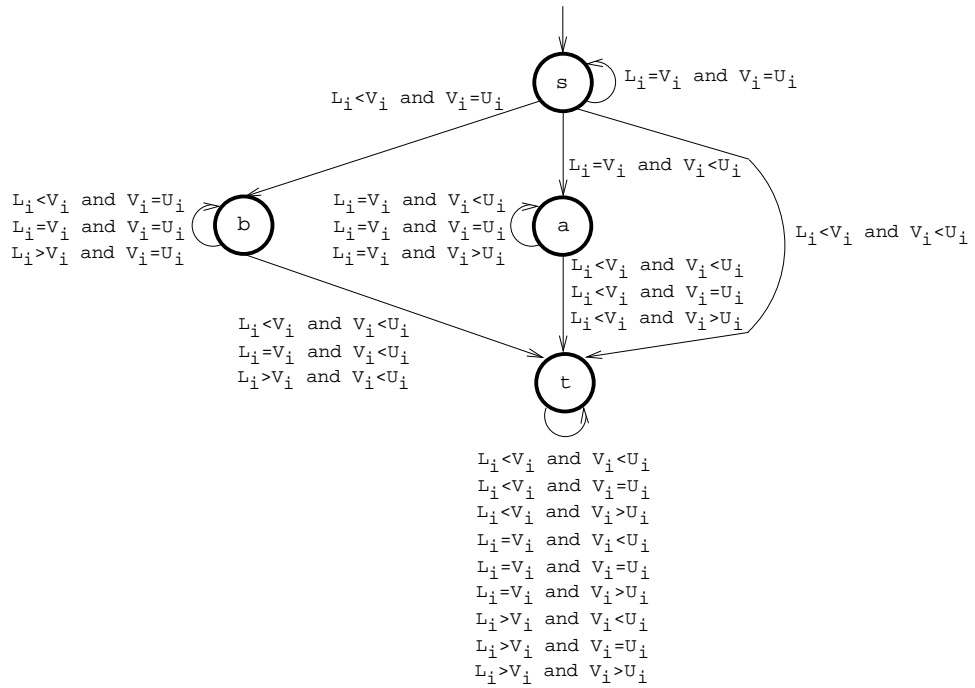


Figure 5.373: Automaton of the `lex_between` constraint

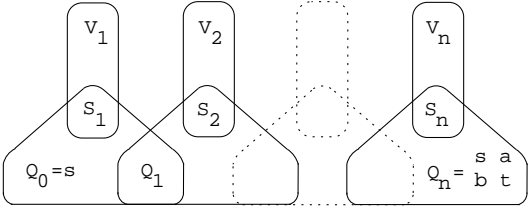


Figure 5.374: Hypergraph of the reformulation corresponding to the automaton of the `lex_between` constraint