

## 5.72 connected

	DESCRIPTION	LINKS	GRAPH
Origin	[126]		
Constraint	<code>connected(NODES)</code>		
Argument	<code>NODES : collection(index—int, succ—svar)</code>		
Restrictions	<pre>required(NODES, [index, succ]) NODES.index ≥ 1 NODES.index ≤  NODES  distinct(NODES, index) NODES.succ ≥ 1 NODES.succ ≤  NODES </pre>		
Purpose	Consider a digraph $G$ described by the <code>NODES</code> collection. Select a subset of arcs of $G$ so that the corresponding graph is symmetric (i.e., if there is an arc from $i$ to $j$ , there is also an arc from $j$ to $i$ ) and connected (i.e., there is a path between any pair of vertices of $G$ ).		
Example	$\left( \begin{array}{ll} \text{index} - 1 & \text{succ} - \{1, 2, 3\}, \\ \text{index} - 2 & \text{succ} - \{1, 3\}, \\ \text{index} - 3 & \text{succ} - \{1, 2, 4\}, \\ \text{index} - 4 & \text{succ} - \{3, 5, 6\}, \\ \text{index} - 5 & \text{succ} - \{4\}, \\ \text{index} - 6 & \text{succ} - \{4\} \end{array} \right)$		
	The <code>connected</code> constraint holds since the <code>NODES</code> collection depicts a symmetric graph involving one single connected component.		
Typical	<code> NODES  &gt; 1</code>		
Symmetry	Items of <code>NODES</code> are <a href="#">permutable</a> .		
Algorithm	A filtering algorithm for the <code>connected</code> constraint is sketched in [126, page 88]. Beside the pruning associated with the fact that the final graph is symmetric, it is based on the fact that all bridges and cutvertices on a path between two vertices that should for sure belong to the final graph should also belong to the final graph.		
See also	<a href="#">common keyword: symmetric (symmetric)</a> . <a href="#">implies: strongly_connected</a> . <a href="#">used in graph description: in_set</a> .		
Keywords	<a href="#">constraint arguments: constraint involving set variables</a> . <a href="#">constraint type: graph constraint</a> . <a href="#">final graph structure: connected component, symmetric</a> .		

<b>Arc input(s)</b>	NODES
<b>Arc generator</b>	$CLIQUE \mapsto collection(nodes1, nodes2)$
<b>Arc arity</b>	2
<b>Arc constraint(s)</b>	$in\_set(nodes2.index, nodes1.succ)$
<b>Graph property(ies)</b>	$NCC = 1$
<b>Graph class</b>	<b>SYMMETRIC</b>

**Graph model**

Part (A) of Figure 5.149 shows the initial graph from which we start. It is derived from the set associated with each vertex. Each set describes the potential values of the succ attribute of a given vertex. Part (B) of Figure 5.149 gives the final graph associated with the **Example** slot.

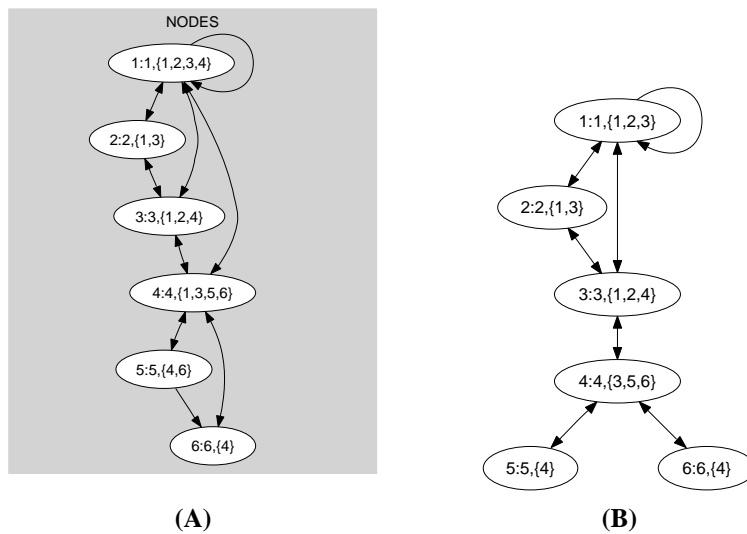


Figure 5.149: Initial and final graph of the connected set constraint