

**5.351** `vec_eq_tuple`

	DESCRIPTION	LINKS	GRAPH
<b>Origin</b>	Used for defining <code>in_relation</code> .		
<b>Constraint</b>	<code>vec_eq_tuple(VARIABLES, TUPLE)</code>		
<b>Arguments</b>	VARIABLES : <code>collection(var-dvar)</code> TUPLE : <code>collection(val-int)</code>		
<b>Restrictions</b>	<code>required(VARIABLES, var)</code> <code>required(TUPLE, val)</code> <code> VARIABLES  =  TUPLE </code>		
<b>Purpose</b>	Enforce a vector of domain variables to be equal to a tuple of values.		
<b>Example</b>	$\left( \langle 5, 3, 3 \rangle, \langle 5, 3, 3 \rangle \right)$ <p>The <code>vec_eq_tuple</code> constraint holds since the first, the second and the third items of <code>VARIABLES = ⟨5, 3, 3⟩</code> are respectively equal to the first, the second and the third items of <code>TUPLE = ⟨5, 3, 3⟩</code>.</p>		
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>Arguments are <code>permutable</code> w.r.t. permutation (<code>VARIABLES, TUPLE</code>).</li> <li>Items of <code>VARIABLES</code> and <code>TUPLE</code> are <code>permutable</code> (<i>same permutation used</i>).</li> </ul>		
<b>Used in</b>	<code>in_relation</code> .		
<b>See also</b>	<b>generalisation:</b> <code>lex_equal</code> ( <i>integer replaced by variable in second argument</i> ). <b>implies:</b> <code>lex_equal</code> .		
<b>Keywords</b>	<b>characteristic of a constraint:</b> <code>tuple</code> . <b>constraint type:</b> value constraint. <b>filtering:</b> arc-consistency.		

<b>Arc input(s)</b>	VARIABLES TUPLE
<b>Arc generator</b>	$PRODUCT(=) \mapsto collection(variables, tuple)$
<b>Arc arity</b>	2
<b>Arc constraint(s)</b>	$variables.var = tuple.val$
<b>Graph property(ies)</b>	$NARC =  VARIABLES $

**Graph model**

Parts (A) and (B) of Figure 5.630 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.

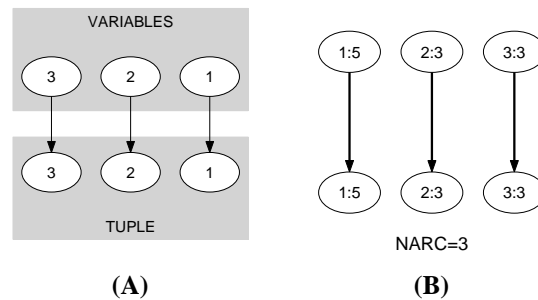


Figure 5.630: Initial and final graph of the `vec_eq_tuple` constraint

**Signature**

Since we use the arc generator  $PRODUCT(=)$  on the collections `VARIABLES` and `TUPLE`, and because of the restriction  $|VARIABLES| = |TUPLE|$ , the maximum number of arcs of the final graph is equal to  $|VARIABLES|$ . Therefore we can rewrite the graph property  $NARC = |VARIABLES|$  to  $NARC \geq |VARIABLES|$  and simplify  $\underline{NARC}$  to  $\overline{NARC}$ .