

5.181 in_relation

	DESCRIPTION	LINKS	GRAPH
Origin	Constraint explicitly defined by tuples of values.		
Constraint	<code>in_relation(VARIABLES, TUPLES_OF_VALS)</code>		
Synonyms	<code>case</code> , <code>extension</code> , <code>extensional</code> , <code>extensional_support</code> , <code>extensional_supportva</code> , <code>extensional_supportmdd</code> , <code>extensional_supportstr</code> , <code>feastupleac</code> , <code>table</code> .		
Types	TUPLE_OF_VARS : <code>collection</code> (var – dvar) TUPLE_OF_VALS : <code>collection</code> (val – int)		
Arguments	VARIABLES : TUPLE_OF_VARS TUPLES_OF_VALS : <code>collection</code> (tuple – TUPLE_OF_VALS)		
Restrictions	<code>required</code> (TUPLE_OF_VARS, var) $ TUPLE_OF_VARS \geq 1$ $ TUPLE_OF_VALS \geq 1$ $ TUPLE_OF_VALS = VARIABLES $ <code>required</code> (TUPLE_OF_VALS, val) <code>required</code> (TUPLES_OF_VALS, tuple)		
Purpose	Enforce the tuple of variables VARIABLES to take its value out of a set of tuples of values TUPLES_OF_VALS. The <i>value</i> of a tuple of variables $\langle V_1, V_2, \dots, V_n \rangle$ is a tuple of values $\langle U_1, U_2, \dots, U_n \rangle$ if and only if $V_1 = U_1 \wedge V_2 = U_2 \wedge \dots \wedge V_n = U_n$.		
Example	$\left(\langle 5, 3, 3 \rangle, \langle \text{tuple} - \langle 5, 2, 3 \rangle, \text{tuple} - \langle 5, 2, 6 \rangle, \text{tuple} - \langle 5, 3, 3 \rangle \rangle \right)$ <p>The <code>in_relation</code> constraint holds since its first argument $\langle 5, 3, 3 \rangle$ corresponds to the third item of the collection of tuples TUPLES_OF_VALS.</p>		
Typical	$ TUPLE_OF_VARS > 1$		
Symmetries	<ul style="list-style-type: none"> Items of TUPLES_OF_VALS are <code>permutable</code>. Items of VARIABLES and TUPLES_OF_VALS.<code>tuple</code> are <code>permutable</code> (<i>same permutation used</i>). All occurrences of two distinct tuples of values in VARIABLES or TUPLES_OF_VALS.<code>tuple</code> can be <code>swapped</code>; all occurrences of a tuple of values in VARIABLES or TUPLES_OF_VALS.<code>tuple</code> can be <code>renamed</code> to any unused tuple of values. 		
Arg. properties	<code>Extensible</code> wrt. TUPLES_OF_VALS.		

Usage	Quite often some constraints cannot be easily expressed, neither by a formula, nor by a regular pattern. In this case one has to define the constraint by specifying in extension the combinations of allowed values.
Remark	The <code>in_relation</code> constraint is called <code>extensional_support</code> in JaCoP (http://www.jacop.eu/). Within SICStus Prolog the constraint can be applied to more than a single tuple of variables and is called <code>table</code> . Within [83] this constraint is called <code>extension</code> . The <code>in_relation</code> constraint is called <code>table</code> in MiniZinc (http://www.minizinc.org/).
Systems	<code>feasPairAC</code> in Choco , <code>infeasPairAC</code> in Choco , <code>relationPairAC</code> in Choco , <code>feasTupleAC</code> in Choco , <code>infeasTupleAC</code> in Choco , <code>relationTupleAC</code> in Choco , <code>extensional</code> in Gecode , <code>extensionalsupportVA</code> in JaCoP , <code>extensionalsupportMDD</code> in JaCoP , <code>extensionalsupportSTR</code> in JaCoP , <code>table</code> in MiniZinc , <code>case</code> in SICStus , <code>relation</code> in SICStus , <code>table</code> in SICStus .
Used in	<code>cond_lex_cost</code> , <code>cond_lex_greater</code> , <code>cond_lex_greatereq</code> , <code>cond_lex_less</code> , <code>cond_lex_lesseq</code> .
See also	common keyword: <code>element</code> (<i>data constraint</i>). cost variant: <code>cond_lex_cost</code> (<i>COST parameter added</i>). used in graph description: <code>vec_eq_tuple</code> .
Keywords	characteristic of a constraint: tuple, derived collection. combinatorial object: relation. constraint type: data constraint, extension. filtering: arc-consistency.

Derived Collection

$$\text{col} \left(\begin{array}{l} \text{TUPLES_OF_VARS} - \text{collection}(\text{vec} - \text{TUPLE_OF_VARS}), \\ [\text{item}(\text{vec} - \text{VARIABLES})] \end{array} \right)$$

Arc input(s)

TUPLES_OF_VARS TUPLES_OF_VALS

Arc generator $\text{PRODUCT} \mapsto \text{collection}(\text{tuples_of_vars}, \text{tuples_of_vals})$ **Arc arity**

2

Arc constraint(s) $\text{vec_eq_tuple}(\text{tuples_of_vars.vec}, \text{tuples_of_vals.tuple})$ **Graph property(ies)** $\text{NARC} \geq 1$ **Graph model**

Parts (A) and (B) of Figure 5.406 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the unique arc of the final graph is stressed in bold.

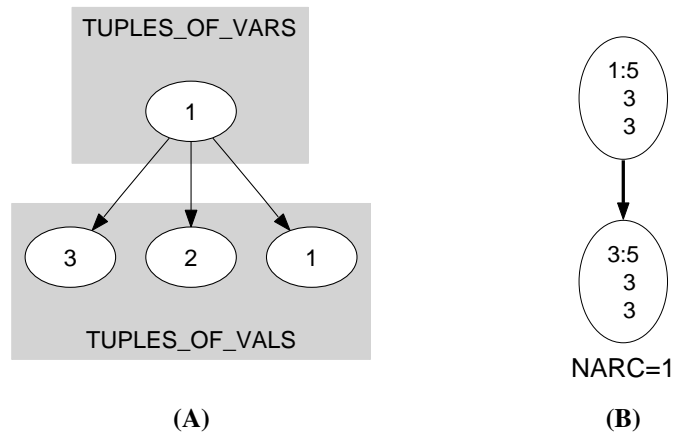


Figure 5.406: Initial and final graph of the `in_relation` constraint

20030820

1347