

5.203 inverse_within_range

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>inverse</code> .		
Constraint	<code>inverse_within_range(X, Y)</code>		
Synonyms	<code>inverse_in_range</code> , <code>inverse_range</code> .		
Arguments	<div>X : <code>collection</code>(var=dvar) Y : <code>collection</code>(var=dvar)</div>		
Restrictions	<div><code>required</code>(X, var) <code>required</code>(Y, var)</div>		
Purpose	<div>If the i^{th} variable of the collection X is assigned to j and if j is greater than or equal to 1 and less than or equal to the number of items of the collection Y then the j^{th} variable of the collection Y is assigned to i. Conversely, if the j^{th} variable of the collection Y is assigned to i and if i is greater than or equal to 1 and less than or equal to the number of items of the collection X then the i^{th} variable of the collection X is assigned to j.</div>		
Example	<div><code>(⟨9, 4, 2⟩, ⟨9, 3, 9, 2⟩)</code></div>		

Since the second item of X is assigned to 4, the fourth item of Y is assigned to 2. Similarly, since the third item of X is assigned to 2, the second item of Y is assigned to 3. Figure 5.471 illustrates the correspondence between X and Y.

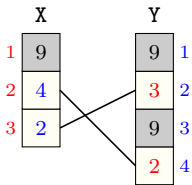


Figure 5.471: Correspondence between the items of $X = \langle 9, 4, 2 \rangle$ and the items of $Y = \langle 9, 3, 9, 2 \rangle$: on the X side values between 1 and $|Y| = 4$ are shown in blue, on the Y side values between 1 and $|X| = 3$ are shown in red.

Typical	<div><code> X > 1</code> <code>range</code>(X.var) > 1 <code> Y > 1</code> <code>range</code>(Y.var) > 1</div>
---------	--

Symmetry

Arguments are [permutable](#) w.r.t. permutation (X, Y).

Usage

Consider an integer value m and a sequence of n variables S from which you have to select a subsequence S' such that:

- All variables of S' have to be assigned to distinct values from $[1, m]$,
- All variables not in S' have to be assigned a value, not necessarily distinct, outside $[1, m]$.

As for the [inverse](#) constraint we may want to create explicitly a *value variable* for each value in $[1, m]$ in order to state some specific constraints on the *value variables* or to use a heuristics involving the original variables of S as well as the *value variables*. The purpose of the `inverse_within_range` constraint is to link the variables of S with the *value variables*.

See also

common keyword: `inverse_set` (*channelling constraint*).

specialisation: `inverse` (*the 2 collections have not necessarily the same number of items*).

Keywords

constraint type: graph constraint.

final graph structure: bipartite, no loop, symmetric.

heuristics: heuristics.

modelling: channelling constraint, dual model.

Arc input(s)	X Y
Arc generator	<i>SYMMETRIC_PRODUCT</i> \mapsto <i>collection</i> (s1, s2)
Arc arity	2
Arc constraint(s)	s1.var = s2.key
Graph class	<ul style="list-style-type: none">• BIPARTITE• NO_LOOP• SYMMETRIC

20060517

1461