5.202 inverse_set

DESCRIPTION LINKS GRAPH

Origin Derived from inverse.

Constraint inverse_set(X, Y)

Arguments

X	:	collection(index−int, set−svar)
Y	:	collection(index−int, set−svar)

Restrictions

required(X, [index, set])
required(Y, [index, set])
increasing_seq(X, index)
increasing_seq(Y, index)
X.index ≥ 1
X.index ≤ |X|
Y.index ≥ 1
Y.index ≤ |Y|
X.set ≥ 1
X.set ≤ |Y|
Y.set ≥ 1
Y.set ≤ |X|

Purpose

The following two statements are equivalent:

1. Value j belongs to the set variable of the i^{th} item of the X collection.
2. Value i belongs to the set variable of the j^{th} item of the Y collection.

I.e., $j \in \mathtt{X}[i] \Leftrightarrow i \in \mathtt{Y}[j]$.

Example

$$\left\langle \left\langle \begin{array}{ll} \mathtt{index}-1 & \mathtt{set}-\{2,4\}, \\ \mathtt{index}-2 & \mathtt{set}-\{4\}, \\ \mathtt{index}-3 & \mathtt{set}-\{1\}, \\ \mathtt{index}-4 & \mathtt{set}-\{4\} \end{array} \right\rangle, \left\langle \begin{array}{ll} \mathtt{index}-1 & \mathtt{set}-\{3\}, \\ \mathtt{index}-2 & \mathtt{set}-\{1\}, \\ \mathtt{index}-3 & \mathtt{set}-\emptyset, \\ \mathtt{index}-4 & \mathtt{set}-\{1,2,4\}, \\ \mathtt{index}-5 & \mathtt{set}-\emptyset \end{array} \right\rangle \right\rangle$$

The inverse_set constraint holds since:

$$\left\{ \begin{array}{l} 2 \in \mathtt{X}[1].\mathtt{set} \Leftrightarrow 1 \in \mathtt{Y}[2].\mathtt{set}, \quad 4 \in \mathtt{X}[1].\mathtt{set} \Leftrightarrow 1 \in \mathtt{Y}[4].\mathtt{set}, \\ 4 \in \mathtt{X}[2].\mathtt{set} \Leftrightarrow 2 \in \mathtt{Y}[4].\mathtt{set}, \\ 1 \in \mathtt{X}[3].\mathtt{set} \Leftrightarrow 3 \in \mathtt{Y}[1].\mathtt{set}, \\ 4 \in \mathtt{X}[4].\mathtt{set} \Leftrightarrow 4 \in \mathtt{Y}[4].\mathtt{set}. \end{array} \right.$$

Typical

|X| > 1
|Y| > 1

Symmetries

- Arguments are permutable w.r.t. permutation (`X`, `Y`).
- Items of `X` are permutable.
- Items of `Y` are permutable.

Usage

The `inverse_set` constraint can for instance be used in order to model problems where one has to place items on a rectangular board in such a way that a column or a row can have more than one item. We have one set variable for each row of the board; Its values are the column indexes corresponding to the positions where an item is placed. Similarly we have also one set variable for each column of the board; Its values are the row indexes corresponding to the positions where an item is placed. The `inverse_set` constraint maintains the link between the rows and the columns variables. Figure 5.469 shows the board that can be associated with the example of the **Example** slot.

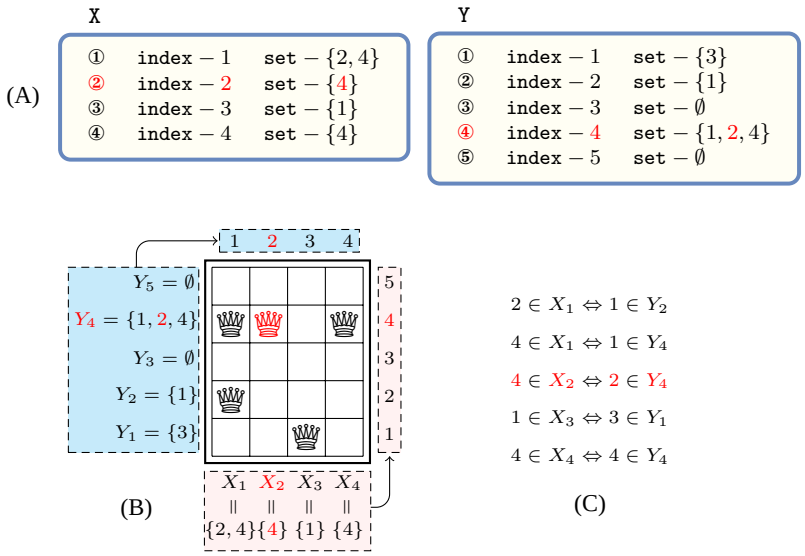


Figure 5.469: Illustration of the **Example** slot where we highlight in red the second item of the `X` collection and the fourth item of the `Y` collection showing the relation between X_2 and Y_4, where X_i (with $1 \leq i \leq 4$) and Y_j (with $1 \leq j \leq 5$) respectively stands for the set attribute of the $i^{\texttt{th}}$ item of the `X` collection and of the $j^{\texttt{th}}$ item of the `Y` collection (A) Collections `X` and `Y` passed to the `inverse_set` constraint, (B) Corresponding board, (C) Conditions linking the items of `X` and the items of `Y`.

Systems

`inverseSet` in **Choco**, `inverse_set` in **MiniZinc**.

See also

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

specialisation: `inverse` *(*`set variable` *replaced by* `domain variable`*)*.

used in graph description: `in_set`.

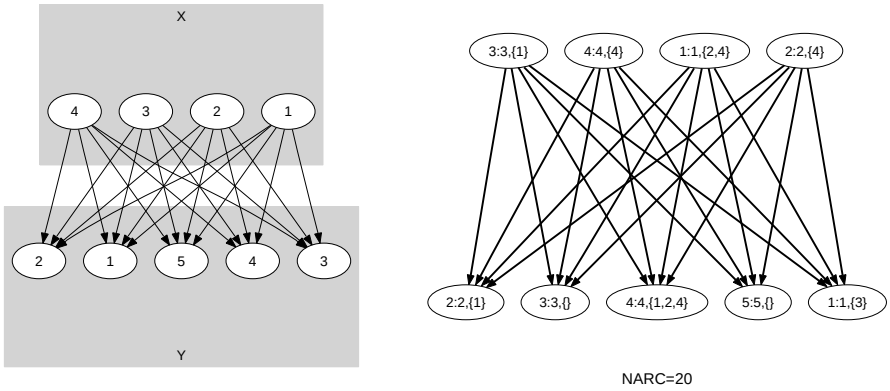
Keywords

constraint arguments: constraint involving set variables.

modelling: channelling constraint, set channel, dual model.

Arc input(s)	X Y
Arc generator	$PRODUCT \mapsto$ collection(x, y)
Arc arity	2
Arc constraint(s)	in_set(y.index, x.set) $\Leftrightarrow$ in_set(x.index, y.set)
Graph property(ies)	**NARC**$= \|X\| * \|Y\|$

Graph model Parts (A) and (B) of Figure 5.470 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.



(A) **(B)**

Figure 5.470: Initial and final graph of the inverse_set constraint