5.14 alldifferent_same_value

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from alldifferent.

Constraint alldifferent_same_value(NSAME, VARIABLES1, VARIABLES2)

Synonyms alldiff_same_value, alldistinct_same_value.

Arguments

NSAME	:	dvar
VARIABLES1	:	collection(var−dvar)
VARIABLES2	:	collection(var−dvar)

Restrictions

NSAME ≥ 0
NSAME $\leq$ |VARIABLES1|
|VARIABLES1| = |VARIABLES2|
required(VARIABLES1, var)
required(VARIABLES2, var)

Purpose All the values assigned to the variables of the collection VARIABLES1 are pairwise distinct. NSAME is equal to number of constraints of the form VARIABLES1$[i]$.var = VARIABLES2$[i]$.var $(1 \leq i \leq$ |VARIABLES1|$)$ that hold.

Example $\left(\begin{array}{l} 2, \langle 7, 3, 1, 5\rangle, \\ \langle 1, 3, 1, 7\rangle \end{array}\right)$

The alldifferent_same_value constraint holds since:

- All the values 7, 3, 1 and 3 are distinct,
- Among the four expressions $7 = 1$, $3 = 3$, $1 = 1$ and $5 = 7$ exactly 2 conditions hold.

Typical

NSAME $<$ |VARIABLES1|
|VARIABLES1| > 2

Symmetries

- Items of VARIABLES1 and VARIABLES2 are permutable (*same permutation used*).
- All occurrences of two distinct values in VARIABLES1.var or VARIABLES2.var can be swapped; all occurrences of a value in VARIABLES1.var or VARIABLES2.var can be renamed to any unused value.

Usage When all variables of the second collection are initially bound to distinct values the alldifferent_same_value constraint can be explained in the following way:

- We interpret the variables of the second collection as the previous solution of a problem where all variables have to be distinct.

- We interpret the variables of the first collection as the current solution to find, where all variables should again be pairwise distinct.

The variable NSAME measures the distance of the current solution from the previous solution. This corresponds to the number of variables of VARIABLES2 that are not assigned to the same previous value.

See also **root concept:** alldifferent.

Keywords **characteristic of a constraint:** automaton, automaton with array of counters.

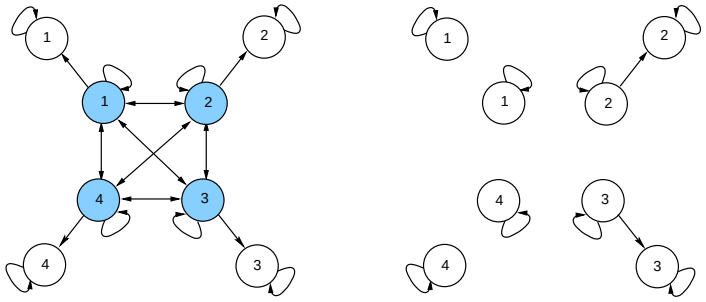
constraint type: proximity constraint.

Arc input(s)	VARIABLES1 VARIABLES2
Arc generator	$PRODUCT(CLIQUE, LOOP, =) \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	• MAX_NSCC ≤ 1 • NARC_NO_LOOP = NSAME

Graph model The arc generator $PRODUCT(CLIQUE, LOOP, =)$ is used in order to generate all the arcs of the initial graph:

- The arc generator $CLIQUE$ creates all links between the items of the first collection VARIABLES1,
- The arc generator $LOOP$ creates a loop for each item of the second collection VARIABLES2,
- Finally the arc generator $PRODUCT(=)$ creates an arc between items located at the same position in the collections VARIABLES1 and VARIABLES2.

Part (A) of Figure 5.19 gives the initial graph associated with the **Example** slot. Variables of collection VARIABLES1 are coloured, while variables of collection VARIABLES2 are kept in white. Part (B) represents the final graph associated with the **Example** slot. In this graph each vertex constitutes a strongly connected component and the number of arcs that do not correspond to a loop is equal to 2 (i.e., NSAME).



(A) **(B)**

Figure 5.19: Initial and final graph of the alldifferent_same_value constraint

Automaton

Figure 5.20 depicts the automaton associated with the `alldifferent_same_value` constraint. Let $\texttt{VAR1}_i$ and $\texttt{VAR2}_i$ respectively denote the i^{th} variables of the `VARIABLES1` and `VARIABLES2` collections. To each pair of variables $(\texttt{VAR1}_i, \texttt{VAR2}_i)$ corresponds a signature variable $\texttt{S}_i$. The following signature constraint links $\texttt{VAR1}_i$, $\texttt{VAR2}_i$ and $\texttt{S}_i$: $\texttt{VAR1}_i = \texttt{VAR2}_i \Leftrightarrow \texttt{S}_i$.

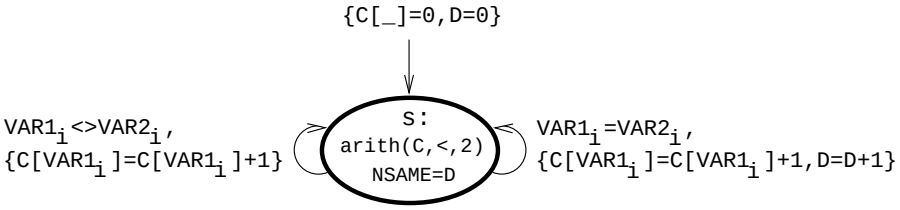


Figure 5.20: Automaton of the `alldifferent_same_value` constraint