5.5 all_equal

DESCRIPTION LINKS GRAPH

Origin Derived from soft_all_equal_min_ctr

Constraint all_equal(VARIABLES)

Synonym rel.

Argument VARIABLES : collection(var−dvar)

Restrictions required(VARIABLES, var)
|VARIABLES| > 0

Purpose Enforce all variables of the collection VARIABLES to take the same value.

Example $(\langle 5,5,5,5\rangle)$

The all_equal constraint holds since all its variables are fixed to value 5.

All solutions Figure 5.5 gives all solutions to the following non ground instance of the all_equal constraint: $\mathtt{V}_1 \in [0,6]$, $\mathtt{V}_2 \in [0,2]$, $\mathtt{V}_3 \in [0,2]$, $\mathtt{V}_4 \in [1,4]$, all_equal$(\langle \mathtt{V}_1, \mathtt{V}_2, \mathtt{V}_3, \mathtt{V}_4\rangle)$.

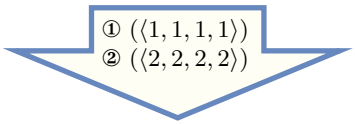


Figure 5.5: All solutions corresponding to the non ground example of the all_equal constraint of the **All solutions** slot

Typical |VARIABLES| > 2
minval(VARIABLES.var) ≠ 0

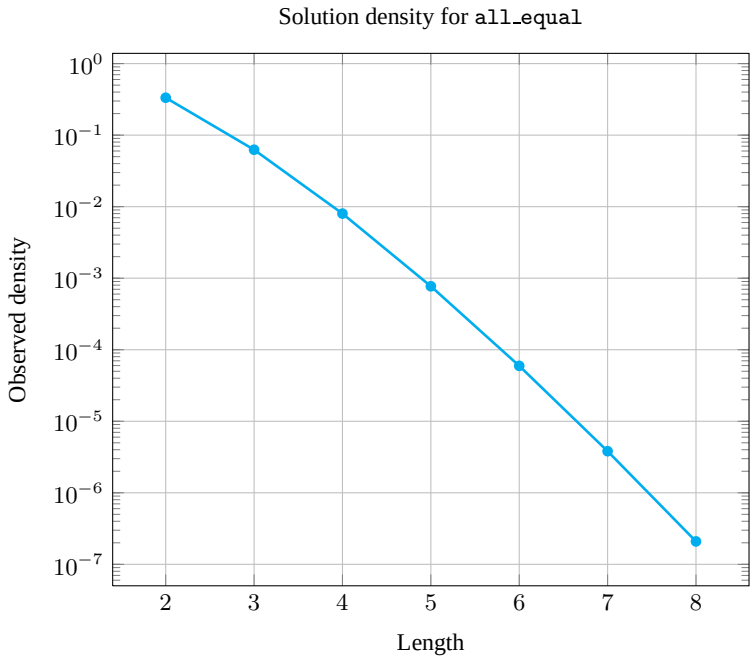
Symmetries
- Items of VARIABLES are permutable.
- All occurrences of a value of VARIABLES.var can be renamed to any unused value.

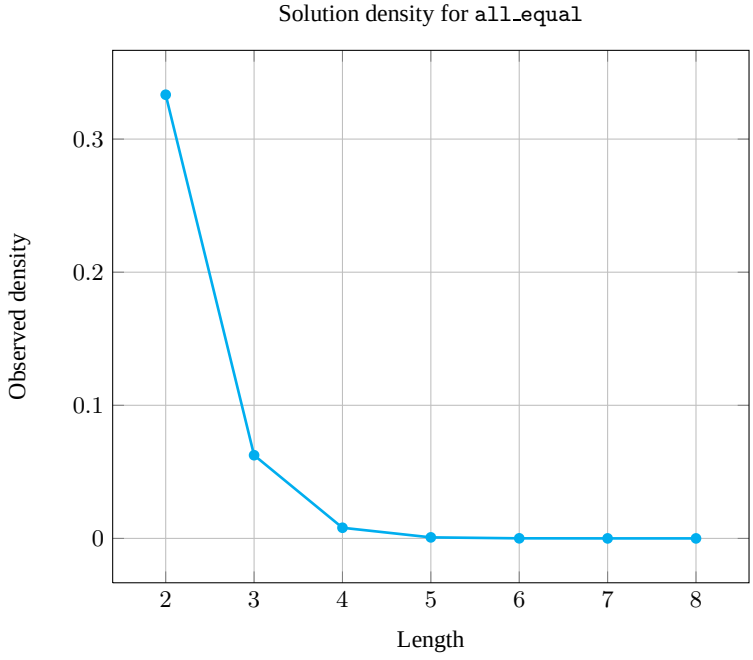
Arg. properties Contractible wrt. VARIABLES.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	3	4	5	6	7	8	9

Number of solutions for all_equal: domains $0..n$





Systems `atMostNValue` in **Choco**, `rel` in **Gecode**, `all_equal` in **MiniZinc**.

See also **generalisation:** `nvalue` *(a variable counting the number of distinct values is introduced).*

implies: consecutive_values, decreasing, increasing, multi_global_contiguity.

negation: not_all_equal.

soft variant: soft_all_equal_max_var, soft_all_equal_min_ctr *(decomposition-based violation measure)*, soft_all_equal_min_var *(variable-based violation measure)*.

specialisation: eq *(equality between just two variables)*.

Keywords **constraint type:** value constraint.

Cond. implications all_equal(VARIABLES)
with |VARIABLES| > 1
implies some_equal(VARIABLES).

Arc input(s)	VARIABLES
Arc generator	$PATH \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	**NARC**= |VARIABLES| − 1

Graph model We use the arc generator $PATH$ in order to link consecutive variables of the collection VARIABLES by a binary equality constraint.

Parts (A) and (B) of Figure 5.6 respectively show the initial and final graph of the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold.

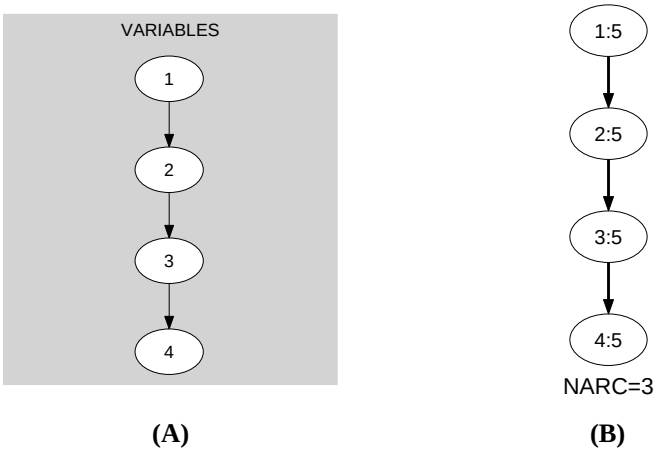


Figure 5.6: Initial and final graph of the all_equal constraint