5.16 alldifferent_except_0

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from `alldifferent`.

Constraint `alldifferent_except_0(VARIABLES)`

Synonyms `alldiff_except_0`, `alldistinct_except_0`.

Argument `VARIABLES : collection(var−dvar)`

Restriction `required(VARIABLES, var)`

Purpose Enforce all variables of the collection `VARIABLES` to take distinct values, except those variables that are assigned value 0.

Example $(\langle 5, 0, 1, 9, 0, 3\rangle)$

The `alldifferent_except_0` constraint holds since all the values (that are different from 0) 5, 1, 9 and 3 are distinct.

All solutions Figure 5.36 gives all solutions to the following non ground instance of the `alldifferent_except_0` constraint: $\mathtt{V}_1 \in [0, 4]$, $\mathtt{V}_2 \in [1, 2]$, $\mathtt{V}_3 \in [1, 2]$, $\mathtt{V}_4 \in [0, 1]$, `alldifferent_except_0`$(\langle \mathtt{V}_1, \mathtt{V}_2, \mathtt{V}_3, \mathtt{V}_4\rangle)$.

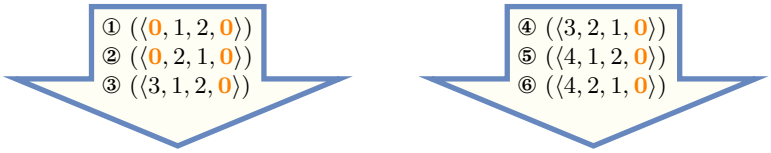


Figure 5.36: All solutions corresponding to the non ground example of the `alldifferent_except_0` constraint of the **All solutions** slot

Typical $|\mathtt{VARIABLES}| > 2$
`atleast`$(2, \mathtt{VARIABLES}, 0)$
`range`$(\mathtt{VARIABLES.var}) > 1$

Symmetries
- Items of `VARIABLES` are permutable.
- Two distinct values of `VARIABLES.var` that are both different from 0 can be swapped; a value of `VARIABLES.var` that is different from 0 can be renamed to any unused value that is also different from 0.

Arg. properties Contractible wrt. `VARIABLES`.

Usage

Quite often it appears that, for some modelling reason, you create a *joker value*. You do not want that normal constraints hold for variables that take this *joker value*. For this purpose we modify the binary arc constraint in order to discard the vertices for which the corresponding variables are assigned value 0. This will be effectively the case since all the corresponding arcs constraints will not hold.

Algorithm

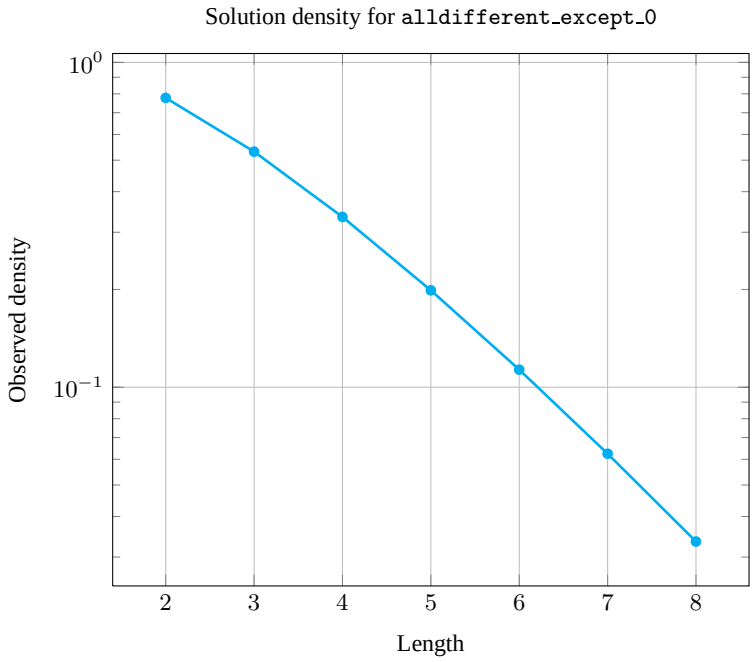
An arc-consistency filtering algorithm for the `alldifferent_except_0` constraint is described in [129]. The algorithm is based on the following ideas:

- First, one can map solutions of the `alldifferent_except_0` constraint to var-perfect matchings[4] in a bipartite graph derived from the domain of the variables of the constraint in the following way: to each variable of the `alldifferent_except_0` constraint corresponds a *variable* and a *joker* vertices, while to each potential value corresponds a *value* vertex; there is an edge between a variable vertex and a value vertex if and only if that value belongs to the domain of the corresponding variable; there is an edge between a variable vertex and its corresponding value vertex.
- Second, Dulmage-Mendelsohn decomposition [148] is used to characterise all edges that do not belong to any var-perfect matching, and therefore prune the corresponding variables.

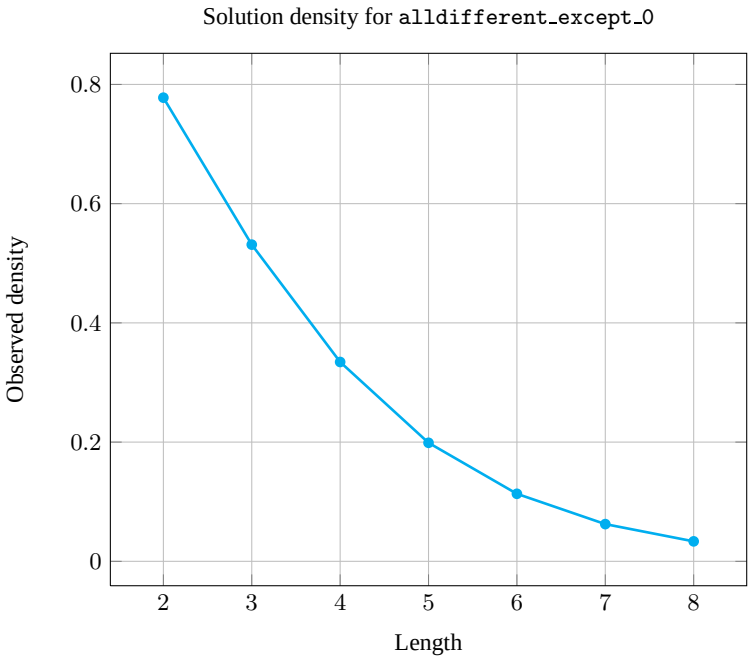
Counting

Length (n)	2	3	4	5	6	7	8
Solutions	7	34	209	1546	13327	130922	1441729

Number of solutions for `alldifferent_except_0`: domains $0..n$



[4] A *var-perfect matching* is a maximum matching covering all vertices representing variables.



See also

cost variant: `weighted_partial_alldiff`.

hard version: `alldifferent`.

implied by: `alldifferent`.

implies: `multi_global_contiguity`.

Keywords

characteristic of a constraint: joker value, all different, sort based reformulation, automaton, automaton with array of counters.

constraint type: value constraint, relaxation.

filtering: bipartite matching, arc-consistency.

final graph structure: one_succ.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	• variables1.var $\neq 0$ • variables1.var = variables2.var
Graph property(ies)	MAX_NSCC ≤ 1

Graph model The graph model is the same as the one used for the alldifferent constraint, except that we discard all variables that are assigned value 0.

Parts (A) and (B) of Figure 5.37 respectively show the initial and final graph associated with the **Example** slot. Since we use the MAX_NSCC graph property we show one of the largest strongly connected components of the final graph. The alldifferent_except_0 holds since all the strongly connected components have at most one vertex: a value different from 0 is used at most once.

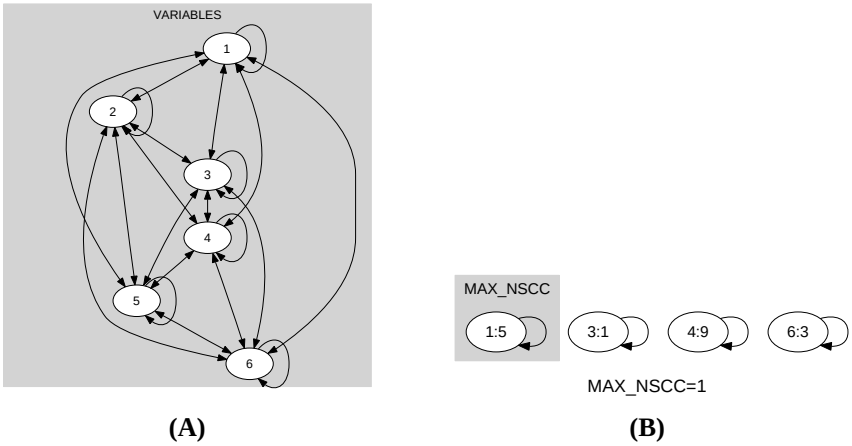


Figure 5.37: Initial and final graph of the alldifferent_except_0 constraint

Automaton

Figure 5.38 depicts the automaton associated with the `alldifferent_except_0` constraint. To each variable $\texttt{VAR}_i$ of the collection `VARIABLES` corresponds a 0-1 signature variable $\texttt{S}_i$. The following signature constraint links $\texttt{VAR}_i$ and $\texttt{S}_i$: $\texttt{VAR}_i \neq 0 \Leftrightarrow \texttt{S}_i$. The automaton counts the number of occurrences of each value different from 0 and finally imposes that each non-zero value is taken at most one time.

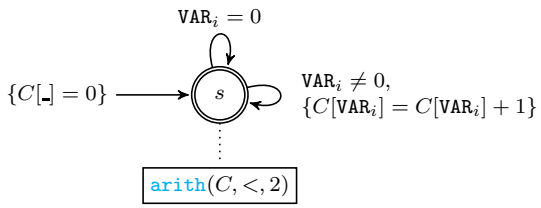


Figure 5.38: Automaton of the `alldifferent_except_0` constraint