5.18 alldifferent_modulo

DESCRIPTION **LINKS** **GRAPH** **AUTOMATON**

Origin Derived from `alldifferent`.

Constraint `alldifferent_modulo(VARIABLES, M)`

Synonyms `alldiff_modulo`, `alldistinct_modulo`.

Arguments

`VARIABLES` : `collection(var−dvar)`
`M` : `int`

Restrictions

`required(VARIABLES, var)`
$\texttt{M} > 0$
$\texttt{M} \geq |\texttt{VARIABLES}|$

Purpose Enforce all variables of the collection `VARIABLES` to have a distinct rest when divided by `M`.

Example $(\langle 25, 1, 14, 3\rangle, 5)$

The equivalence classes associated with values 25, 1, 14 and 3 are respectively equal to $25 \bmod 5 = 0$, $1 \bmod 5 = 1$, $14 \bmod 5 = 4$ and $3 \bmod 5 = 3$. Since they are distinct the `alldifferent_modulo` constraint holds.

All solutions Figure 5.42 gives all solutions to the following non ground instance of the `alldifferent_modulo` constraint: $\texttt{V}_1 \in \{0, 5\}$, $\texttt{V}_2 \in [2, 3]$, $\texttt{V}_3 \in [3, 4]$, $\texttt{V}_4 \in [1, 2]$, $\texttt{V}_5 \in [6, 10]$, $\texttt{alldifferent\_modulo}(\langle \texttt{V}_1, \texttt{V}_2, \texttt{V}_3, \texttt{V}_4, \texttt{V}_5\rangle, \mathbf{5})$.

① $(\langle 0_0, 2_2, 3_3, 1_1, 9_4\rangle, \mathbf{5})$
② $(\langle 0_0, 2_2, 4_4, 1_1, 8_3\rangle, \mathbf{5})$
③ $(\langle 0_0, 3_3, 4_4, 1_1, 7_2\rangle, \mathbf{5})$
④ $(\langle 0_0, 3_3, 4_4, 2_2, 6_1\rangle, \mathbf{5})$

⑤ $(\langle 5_0, 2_2, 3_3, 1_1, 9_4\rangle, \mathbf{5})$
⑥ $(\langle 5_0, 2_2, 4_4, 1_1, 8_3\rangle, \mathbf{5})$
⑦ $(\langle 5_0, 3_3, 4_4, 1_1, 7_2\rangle, \mathbf{5})$
⑧ $(\langle 5_0, 3_3, 4_4, 2_2, 6_1\rangle, \mathbf{5})$

Figure 5.42: All solutions corresponding to the non ground example of the `alldifferent_modulo` constraint of the **All solutions** slot, where the indices (in orange) correspond to the values modulo $\texttt{M} = \mathbf{5}$: all indices attached to a solution are distinct.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{M} > 1$

Symmetries

- Items of VARIABLES are permutable.
- A value u of VARIABLES.var can be renamed to any value v such that v is congruent to u modulo M.
- Two distinct values u and v of VARIABLES.var such that $u \bmod \mathtt{M} \neq v \bmod \mathtt{M}$ can be swapped.

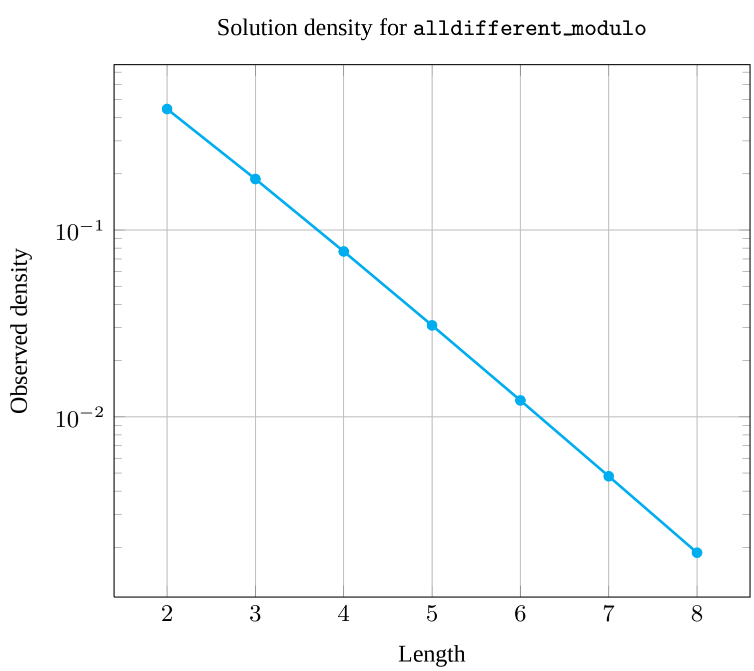
Arg. properties

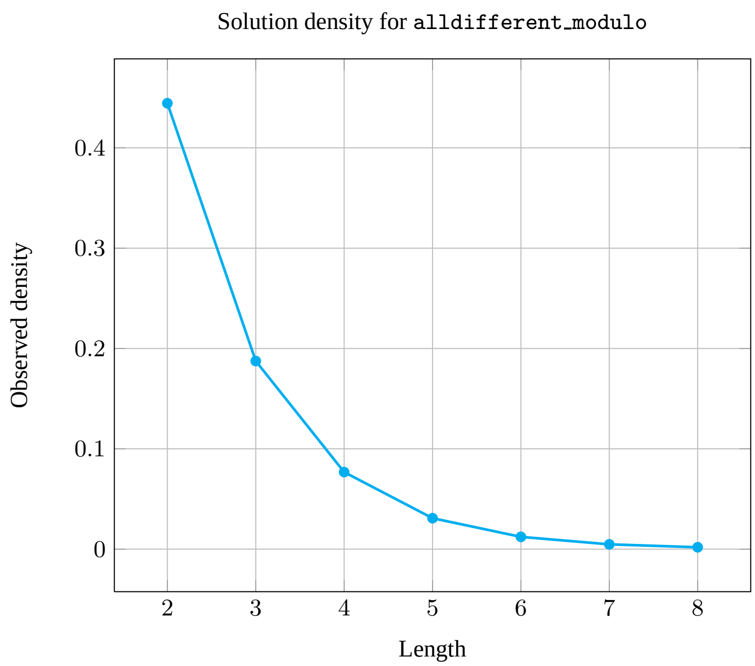
Contractible wrt. VARIABLES.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	4	12	48	240	1440	10080	80640

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

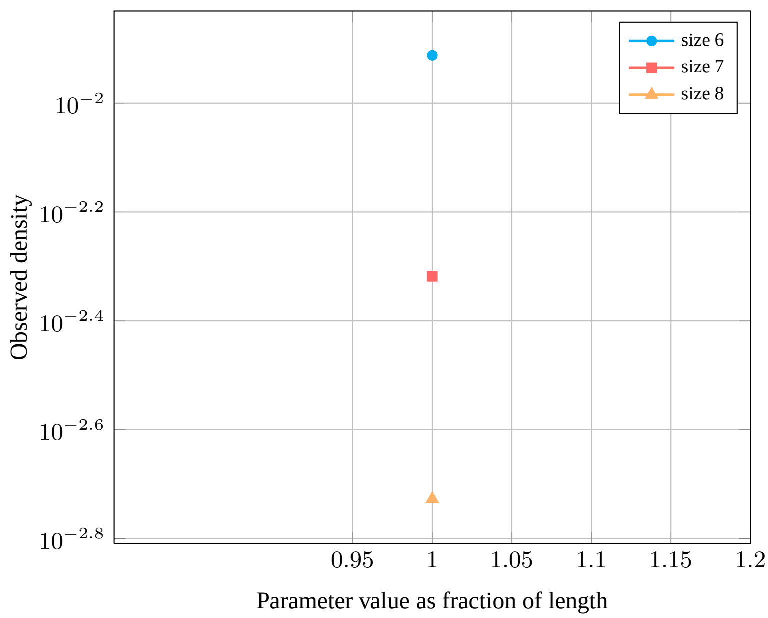




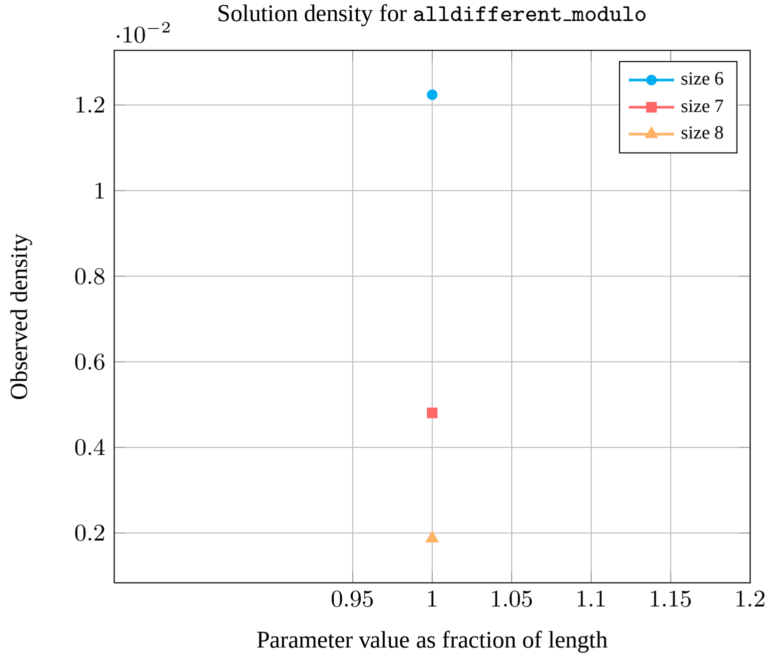
Length (n)		2	3	4	5	6	7	8
Total		4	12	48	240	1440	10080	80640
Parameter value	2	4	-	-	-	-	-	-
	3	-	12	-	-	-	-	-
	4	-	-	48	-	-	-	-
	5	-	-	-	240	-	-	-
	6	-	-	-	-	1440	-	-
	7	-	-	-	-	-	10080	-
	8	-	-	-	-	-	-	80640

Solution count for `alldifferent_modulo`: domains $0..n$

Solution density for `alldifferent_modulo`



Solution density for `alldifferent_modulo`



See also

implies: `soft_alldifferent_var`.

specialisation: `alldifferent` *(*`variable` mod `constant` *replaced by* `variable`*)*.

Keywords **characteristic of a constraint:** modulo, all different, sort based reformulation, automaton, automaton with array of counters.

constraint type: value constraint.

filtering: 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 $\bmod$ M $=$ variables2.var $\bmod$ M
Graph property(ies)	**MAX_NSCC** ≤ 1
Graph class	ONE_SUCC

Graph model Exploit the same model used for the `alldifferent` constraint. We replace the binary *equality* constraint by another equivalence relation depicted by the arc constraint. We generate a *clique* with a binary *equality modulo* M constraint between each pair of vertices (including a vertex and itself) and state that the size of the largest strongly connected component should not exceed 1.

Parts (A) and (B) of Figure 5.43 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.

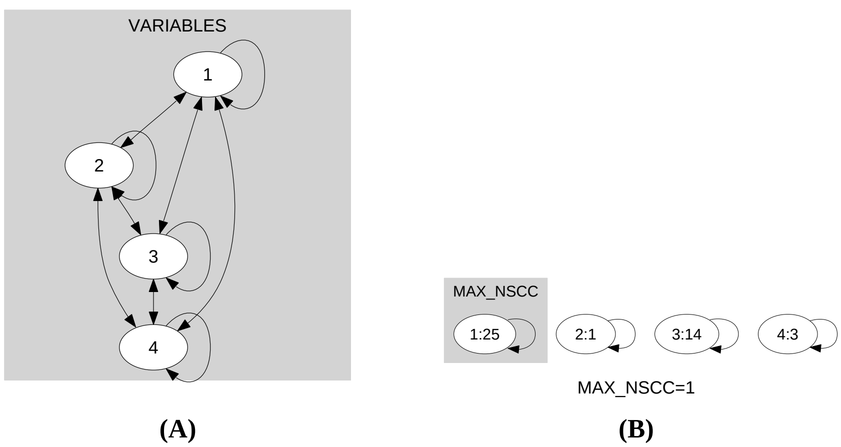


Figure 5.43: Initial and final graph of the `alldifferent_modulo` constraint

Automaton Figure 5.44 depicts the automaton associated with the `alldifferent_modulo` constraint. To each item of the collection `VARIABLES` corresponds a signature variable $\mathtt{S}_i$ that is equal to 1. The automaton counts for each equivalence class the number of used values and finally imposes that each equivalence class is used at most one time.

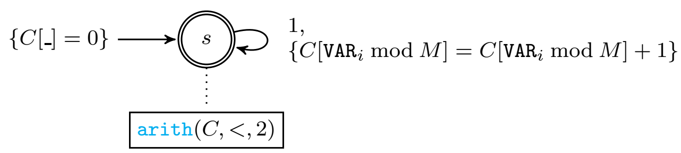


Figure 5.44: Automaton of the `alldifferent_modulo` constraint