5.24 among_diff_0

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Used in the automaton of nvalue.

Constraint among_diff_0(NVAR, VARIABLES)

Arguments

NVAR	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

NVAR ≥ 0
NVAR $\leq$ |VARIABLES|
required(VARIABLES, var)

Purpose NVAR is the number of variables of the collection VARIABLES that take a value different from 0.

Example

$(3, \langle 0, 5, 5, 0, 1\rangle)$
$(0, \langle 0, 0, 0, 0, 0\rangle)$
$(1, \langle 0, 0, 0, 6, 0\rangle)$

The first among_diff_0 constraint holds since exactly 3 values of the collection of values $\langle 0, 5, 5, 0, 1\rangle$ are different from 0.

Typical

NVAR > 0
NVAR $<$ |VARIABLES|
|VARIABLES| > 1
atleast(1, VARIABLES, 0)
$2 *$ among_diff_0(VARIABLES.var) $>$ |VARIABLES|

Symmetries

- Items of VARIABLES are permutable.
- An occurrence of a value of VARIABLES.var that is different from 0 can be replaced by any other value that is also different from 0.

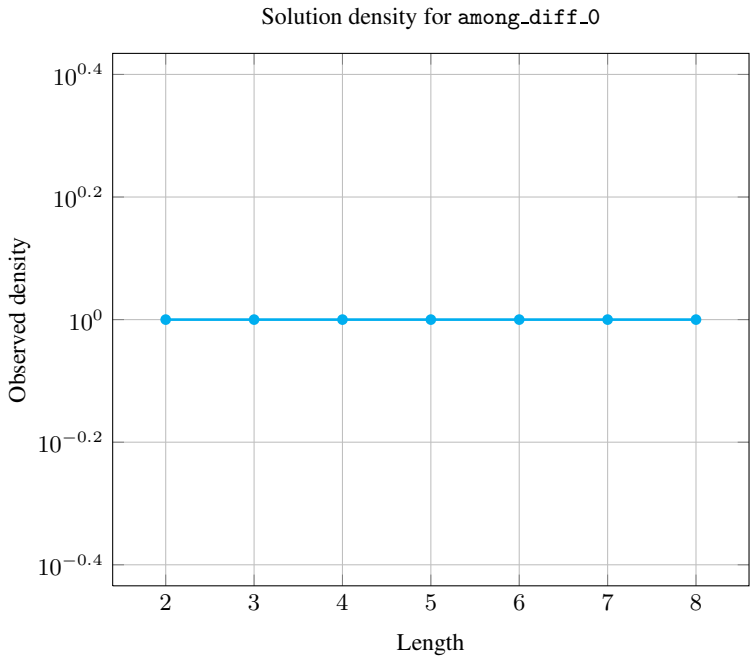
Arg. properties

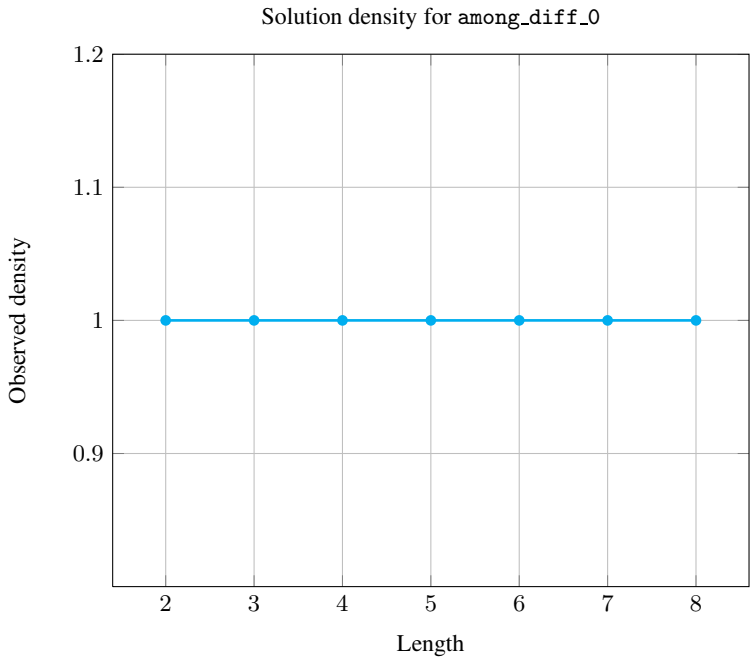
- Functional dependency: NVAR determined by VARIABLES.
- Contractible wrt. VARIABLES when NVAR $= 0$.
- Contractible wrt. VARIABLES when NVAR $=$ |VARIABLES|.
- Aggregate: NVAR(+), VARIABLES(union).

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	9	64	625	7776	117649	2097152	43046721

Number of solutions for among_diff_0: domains $0..n$

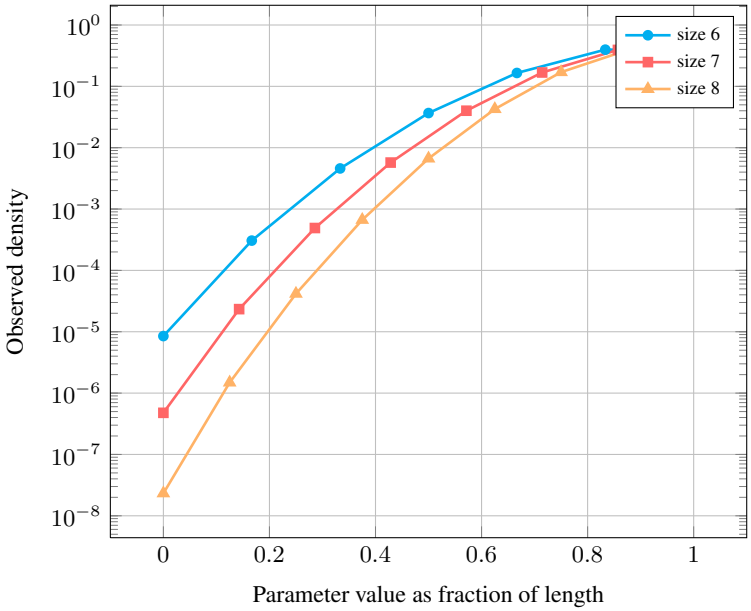

Solution density for among_diff_0
Observed density
$10^{0.4}$
$10^{0.2}$
10^{0}
$10^{-0.2}$
$10^{-0.4}$
2
3
4
5
6
7
8
Length

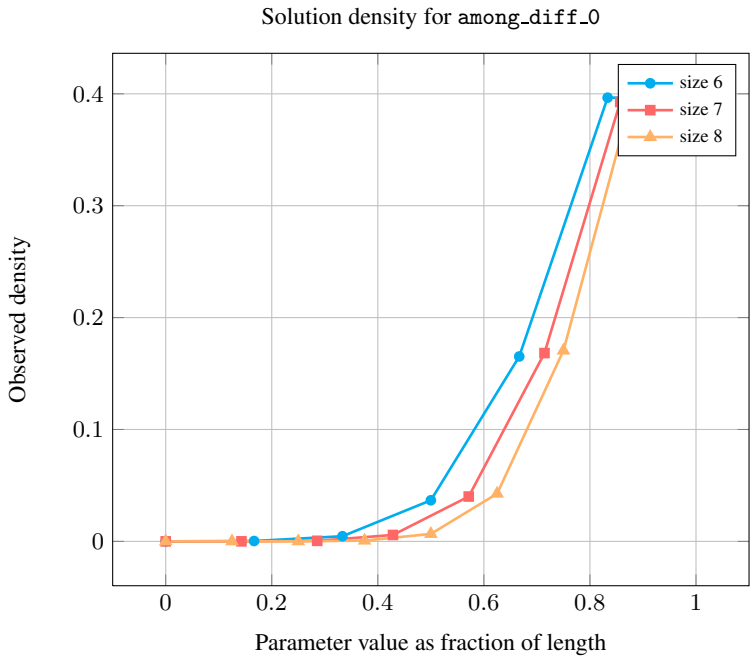

Solution density for among_diff_0
Observed density
1.2
1.1
1
0.9
2
3
4
5
6
7
8
Length

Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	0	1	1	1	1	1	1	1
	1	4	9	16	25	36	49	64
	2	4	27	96	250	540	1029	1792
	3	-	27	256	1250	4320	12005	28672
	4	-	-	256	3125	19440	84035	286720
	5	-	-	-	3125	46656	352947	1835008
	6	-	-	-	-	46656	823543	7340032
	7	-	-	-	-	-	823543	16777216
	8	-	-	-	-	-	-	16777216

Solution count for `among_diff_0`: domains $0..n$

Solution density for `among_diff_0`





See also

common keyword: nvalue *(counting constraint)*.

generalisation: among *(*variable $\neq$ 0 *replaced by* variable $\in$ values*)*.

Keywords

characteristic of a constraint: joker value, automaton, automaton with counters.

constraint arguments: pure functional dependency.

constraint network structure: alpha-acyclic constraint network(2).

constraint type: value constraint, counting constraint.

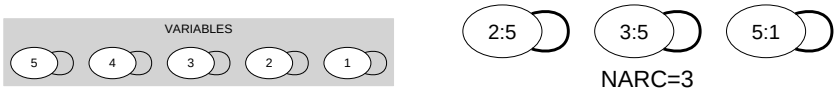
filtering: arc-consistency.

modelling: functional dependency.

Arc input(s)	VARIABLES
Arc generator	$SELF \mapsto$ collection(variables)
Arc arity	1
Arc constraint(s)	variables.var $\neq 0$
Graph property(ies)	NARC$=$ NVAR

Graph model Since this is a unary constraint we employ the *SELF* arc generator in order to produce an initial graph with a single loop on each vertex.

Parts (A) and (B) of Figure 5.55 respectively show the initial and final graph associated with first example of the **Example** slot. Since we use the NARC graph property, the loops of the final graph are stressed in bold.



(A) **(B)**

Figure 5.55: Initial and final graph of the among_diff_0 constraint

Automaton

Figure 5.56 depicts the automaton associated with the among_diff_0 constraint. To each variable VAR$_i$ of the collection VARIABLES corresponds a 0-1 signature variable S_i. The following signature constraint links VAR$_i$ and S_i: VAR$_i \neq 0 \Leftrightarrow S_i$. The automaton counts the number of variables of the VARIABLES collection that take a value different from 0 and finally assigns this number to NVAR.

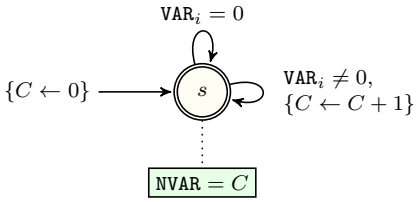


Figure 5.56: Automaton of the among_diff_0 constraint

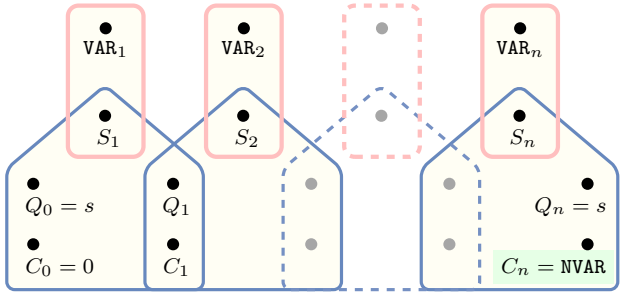


Figure 5.57: Hypergraph of the reformulation corresponding to the automaton (with one counter) of the among_diff_0 constraint: since all states variables $Q_0, Q_1, \ldots, Q_n$ are fixed to the unique state s of the automaton, the transitions constraints share only the counter variable C and the constraint network is Berge-acyclic