5.271 nand

DESCRIPTION LINKS AUTOMATON

Origin Logic

Constraint nand(VAR, VARIABLES)

Synonym clause.

Arguments

VAR	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

VAR ≥ 0
VAR ≤ 1
|VARIABLES| ≥ 2
required(VARIABLES, var)
VARIABLES.var ≥ 0
VARIABLES.var ≤ 1

Purpose Let VARIABLES be a collection of 0-1 variables $\mathtt{VAR}_1, \mathtt{VAR}_2, \ldots, \mathtt{VAR}_n$ $(n \geq 2)$. Enforce $\mathtt{VAR} = \neg(\mathtt{VAR}_1 \wedge \mathtt{VAR}_2 \wedge \cdots \wedge \mathtt{VAR}_n)$.

Example

$(1, \langle 0, 0\rangle)$
$(1, \langle 0, 1\rangle)$
$(1, \langle 1, 0\rangle)$
$(0, \langle 1, 1\rangle)$
$(1, \langle 1, 0, 1\rangle)$

Symmetry Items of VARIABLES are permutable.

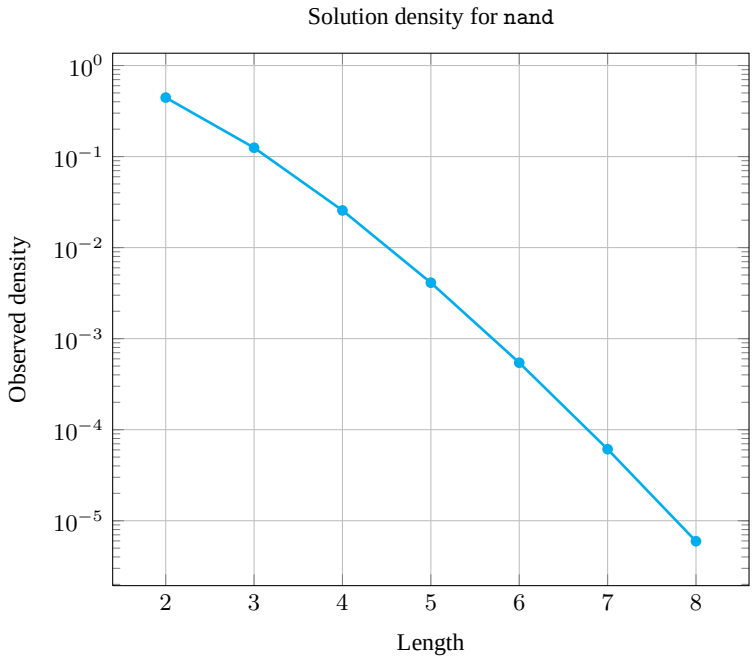
Arg. properties

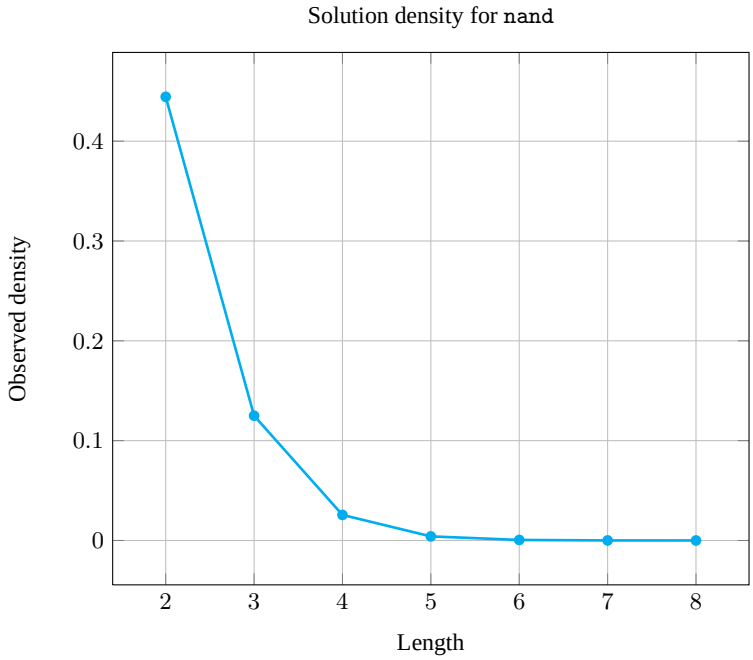
- Functional dependency: VAR determined by VARIABLES.
- Contractible wrt. VARIABLES when VAR $= 0$.
- Extensible wrt. VARIABLES when VAR $= 1$.
- Aggregate: VAR($\vee$), VARIABLES(union).

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	4	8	16	32	64	128	256

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

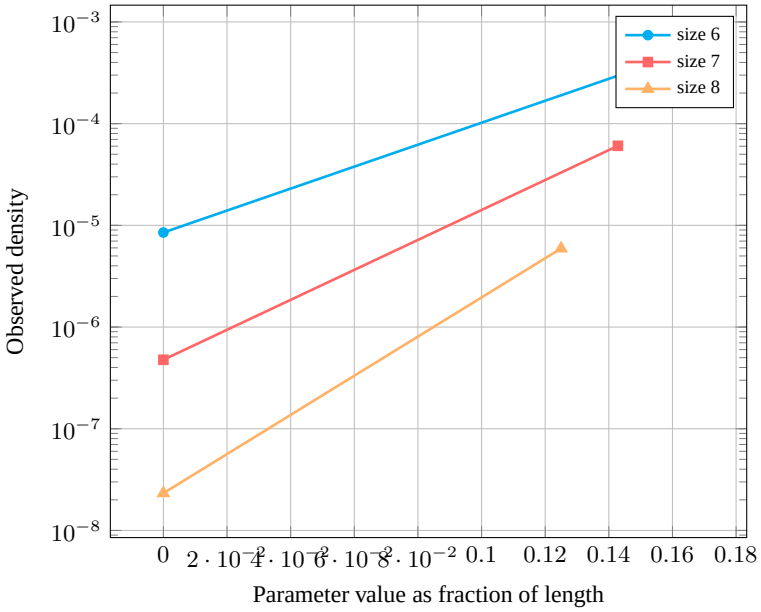

Solution density for nand
Observed density
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
2
3
4
5
6
7
8
Length

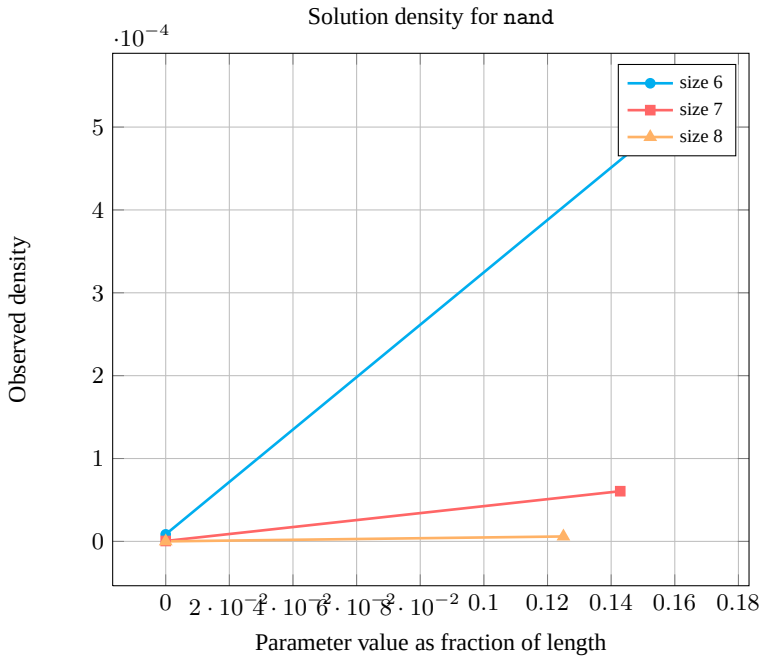

Solution density for nand
Observed density
0.4
0.3
0.2
0.1
0
2
3
4
5
6
7
8
Length

Length (n)		2	3	4	5	6	7	8
Total		4	8	16	32	64	128	256
Parameter value	0	1	1	1	1	1	1	1
	1	3	7	15	31	63	127	255

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

Solution density for `nand`





Systems `clause` in **Choco**, `clause` in **Gecode**, `#/\` in **SICStus**.

See also **common keyword:** `and`, `equivalent`, `imply`, `nor`, `or`, `xor` *(Boolean constraint)*.

implies: `atleast_nvalue`.

Keywords **characteristic of a constraint:** automaton, automaton without counters, reified automaton constraint.

constraint arguments: pure functional dependency.

constraint network structure: Berge-acyclic constraint network.

constraint type: Boolean constraint.

filtering: arc-consistency.

modelling: functional dependency.

Cond. implications `nand(VAR, VARIABLES)`
with $|$`VARIABLES`$| > 2$
implies `some_equal(VARIABLES)`.

Automaton

Figure 5.590 depicts the automaton associated with the nand constraint. To the first argument VAR of the nand constraint corresponds the first signature variable. To each variable $\texttt{VAR}_i$ of the second argument VARIABLES of the nand constraint corresponds the next signature variable. There is no signature constraint.

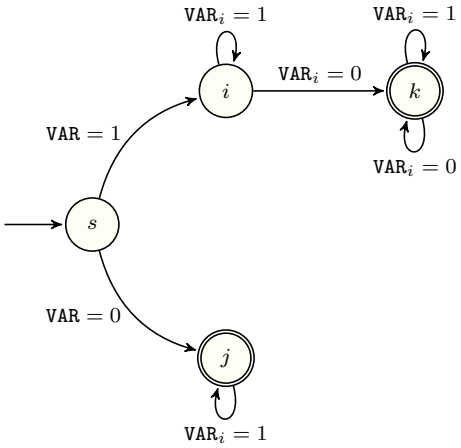


Figure 5.590: Automaton of the nand constraint

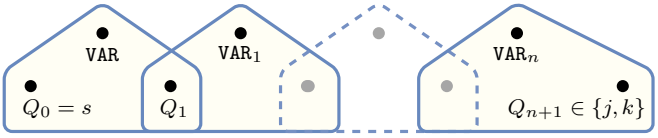


Figure 5.591: Hypergraph of the reformulation corresponding to the automaton of the nand constraint