5.304 or

DESCRIPTION LINKS AUTOMATON

Origin Logic

Constraint or(VAR, VARIABLES)

Synonym rel.

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 $\mathtt{VAR} = \mathtt{VAR}_1 \vee \mathtt{VAR}_2 \vee \cdots \vee \mathtt{VAR}_n$.

Example

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

Symmetry Items of VARIABLES are permutable.

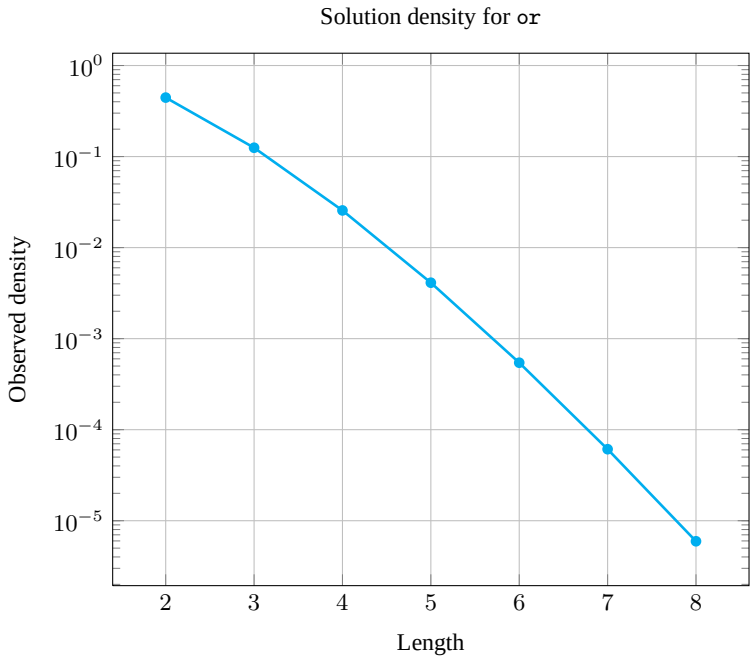
Arg. properties

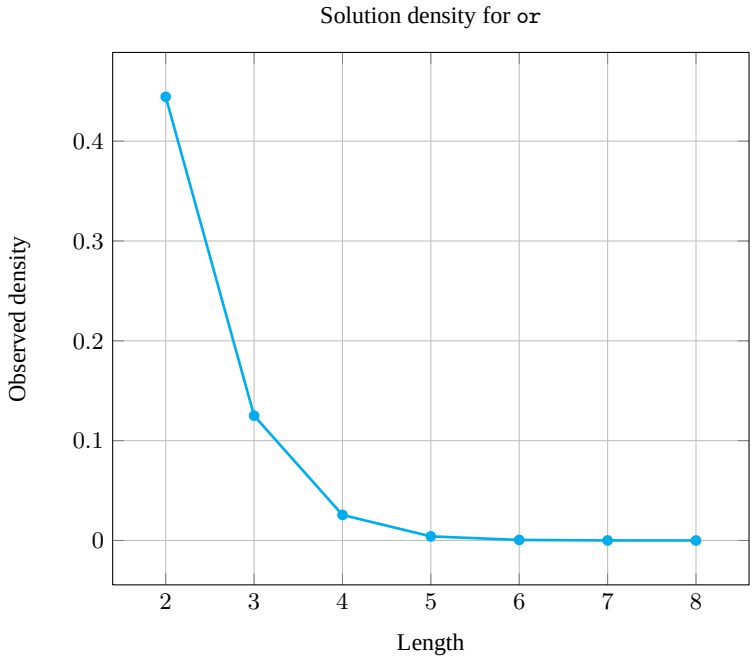
- Functional dependency: VAR determined by VARIABLES.
- Contractible wrt. VARIABLES when VAR $= 0$.
- Extensible wrt. VARIABLES when VAR $= 1$.
- Aggregate: VAR($\vee$), VARIABLES(union).

Counting

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

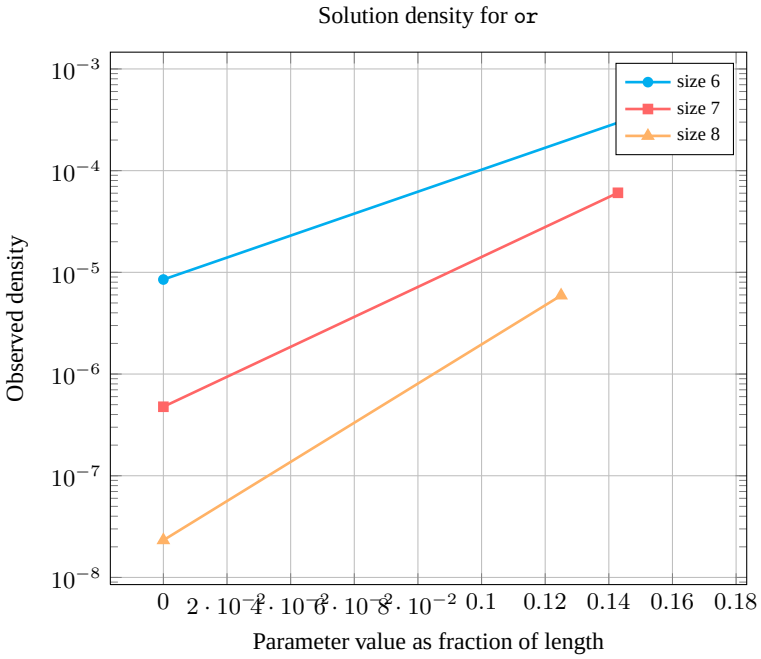
Number of solutions for or: domains $0..n$

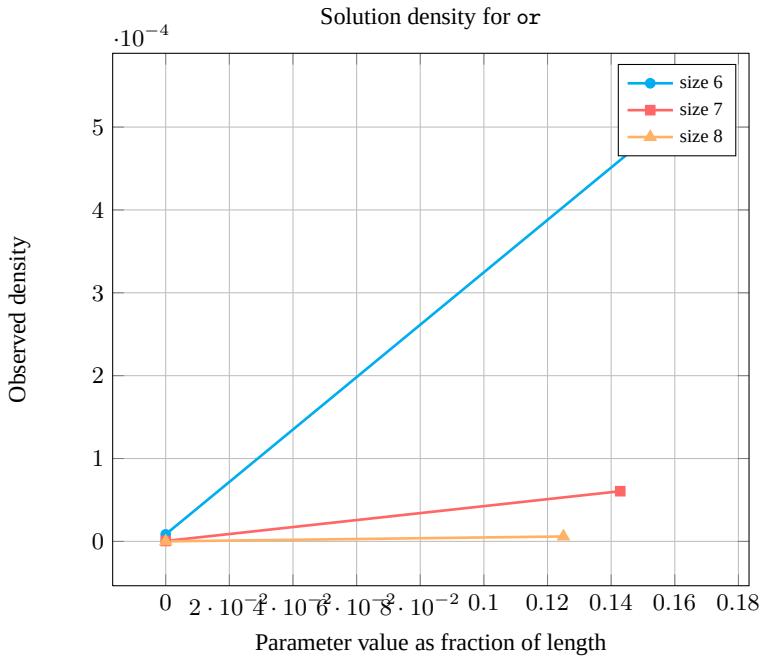

Solution density for or
Observed density
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
2
3
4
5
6
7
8
Length


Solution density for or
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 `or`: domains $0..n$





Systems `reifiedOr` in **Choco**, `rel` in **Gecode**, `orbool` in **JaCoP**, `#\/` in **SICStus**.

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

implies: `atleast_nvalue`, `maximum`.

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: disjunction, functional dependency.

Cond. implications

- `or`(`VAR`, `VARIABLES`)
  with |`VARIABLES`| > 2
  implies `some_equal`(`VARIABLES`).
- `or`(`VAR`, `VARIABLES`)
  with `VAR` = 0
  implies `nor`(`VAR`, `VARIABLES`)
  when `VAR` = 1.
- `or`(`VAR`, `VARIABLES`)
  with `VAR` = 1
  implies `nor`(`VAR`, `VARIABLES`)
  when `VAR` = 0.

Automaton

Figure 5.640 depicts a first deterministic automaton without counter associated with the or constraint. To the first argument VAR of the or constraint corresponds the first signature variable. To each variable $\texttt{VAR}_i$ of the second argument VARIABLES of the or constraint corresponds the next signature variable. There is no signature constraint.

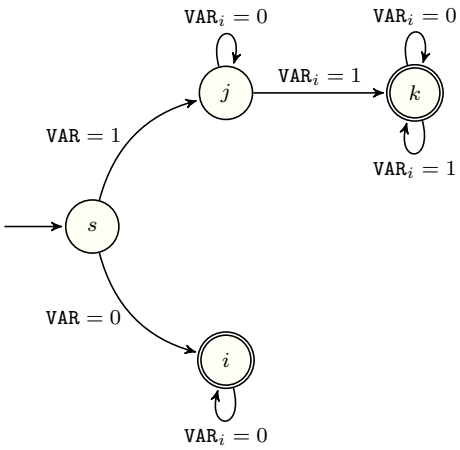


Figure 5.640: Counter free automaton of the or constraint

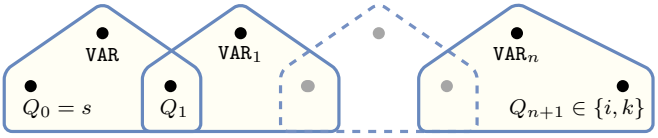


Figure 5.641: Hypergraph of the reformulation corresponding to the automaton of the or constraint

Figure 5.642 depicts a second deterministic automaton with one counter associated with the or constraint, where the argument VAR is unified to the final value of the counter.

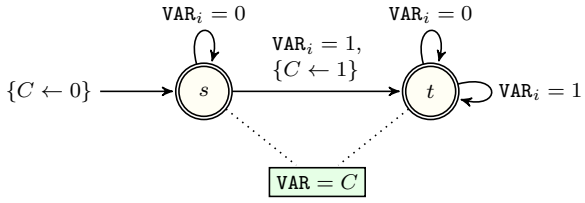


Figure 5.642: Automaton (with one counter) of the or constraint

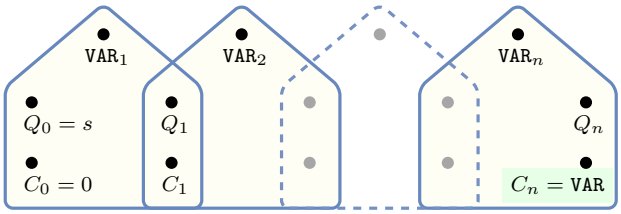


Figure 5.643: Hypergraph of the reformulation corresponding to the automaton (with one counter) of the `or` constraint (since all states of the automaton are accepting there is no restriction on the last variable Q_n)