## 5.421 xor

DESCRIPTION LINKS AUTOMATON

**Origin** Logic

**Constraint** xor(VAR, VARIABLES)

**Synonyms** odd, rel.

**Arguments**

| | | |
|---|---|---|
| VAR | : | dvar |
| VARIABLES | : | collection(var−dvar) |

**Restrictions**

VAR $\geq 0$
VAR $\leq 1$
|VARIABLES| $= 2$
required(VARIABLES, var)
VARIABLES.var $\geq 0$
VARIABLES.var $\leq 1$

**Purpose** Let VARIABLES be a collection of 0-1 variables $\text{VAR}_1, \text{VAR}_2$. Enforce VAR $=$ ($\text{VAR}_1 \neq \text{VAR}_2$).

**Example**

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

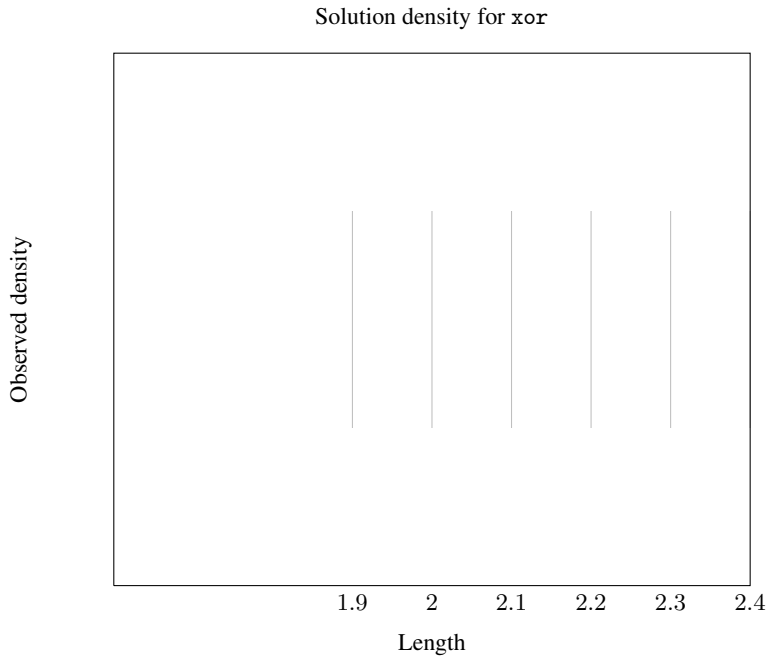
**Symmetry** Items of VARIABLES are permutable.

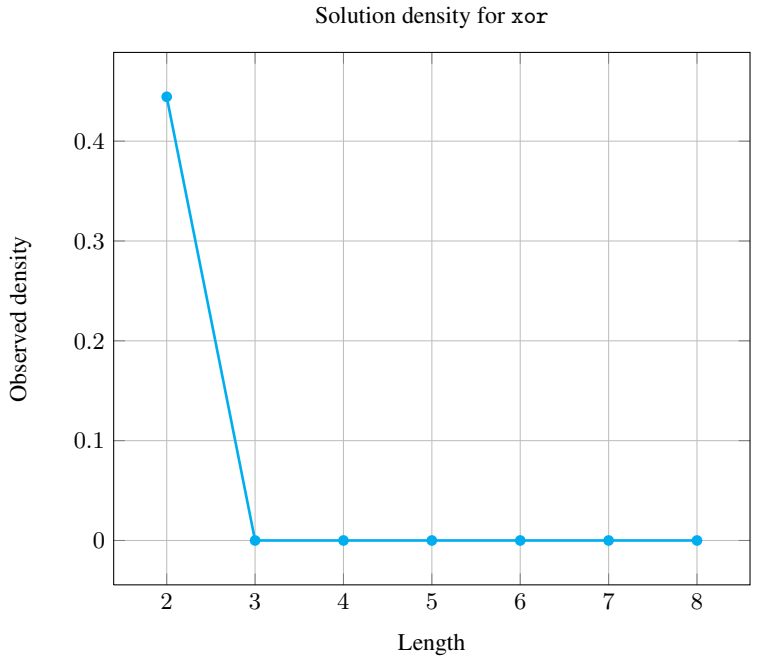
**Arg. properties** Functional dependency: VAR determined by VARIABLES.

**Counting**

| Length ($n$) | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Solutions | 4 | 0 | 0 | 0 | 0 | 0 | 0 |

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

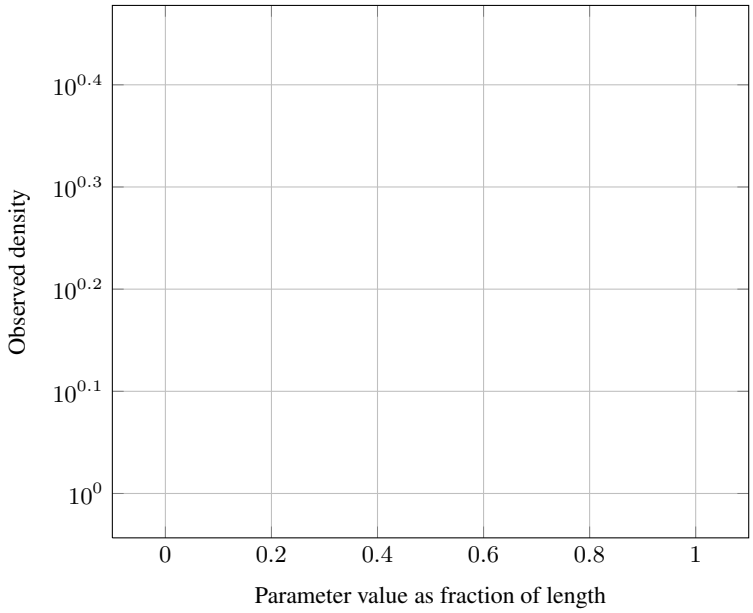

Solution density for xor
Observed density
1.9
2
2.1
2.2
2.3
2.4
Length

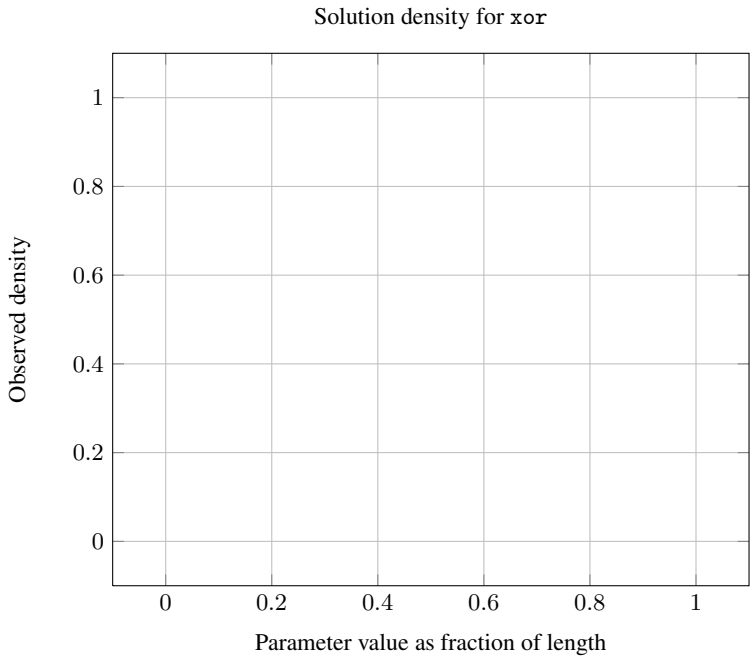

Solution density for xor
Observed density
0.4
0.3
0.2
0.1
0
2
3
4
5
6
7
8
Length

| Length ($n$) | | 2 |
|---|---|---|
| Total | | 4 |
| Parameter value | 0 | 2 |
| | 1 | 2 |

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

Solution density for `xor`






**Systems** reifiedXor in **Choco**, rel in **Gecode**, xorbool in **JaCoP**, #“in **SICStus**.

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

**implies:** `atleast_nvalue`, `soft_all_equal_max_var`, `soft_all_equal_min_var`.

**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.

**Automaton**

Figure 5.805 depicts the automaton associated with the `xor` constraint. To the first argument `VAR` of the `xor` constraint corresponds the first signature variable. To each variable $\texttt{VAR}_i$ of the second argument `VARIABLES` of the `xor` constraint corresponds the next signature variable. There is no signature constraint.

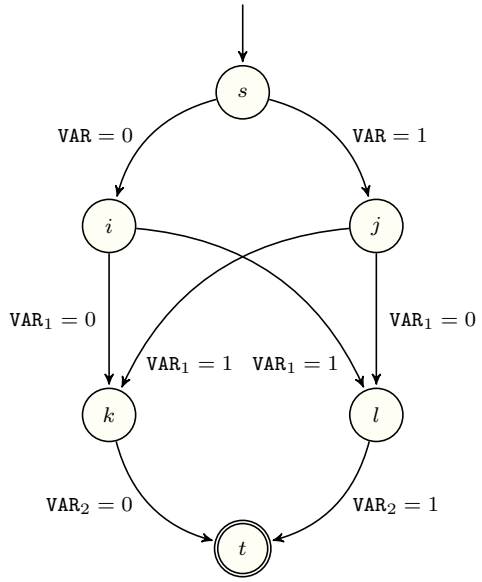



Figure 5.805: Automaton of the `xor` constraint

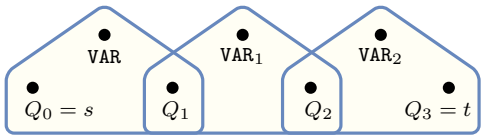



Figure 5.806: Hypergraph of the reformulation corresponding to the automaton of the `xor` constraint