5.176　imply

Origin　　Logic

Constraint　　imply(VAR, VARIABLES)

Synonyms　　rel, ifthen.

Arguments

VAR　　　　　 : dvar
VARIABLES　　: collection(var−dvar)

Restrictions

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

Purpose

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

Example

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

Symmetry　　All occurrences of 0 in VAR and in VARIABLES.var can be set to 1.

Arg. properties
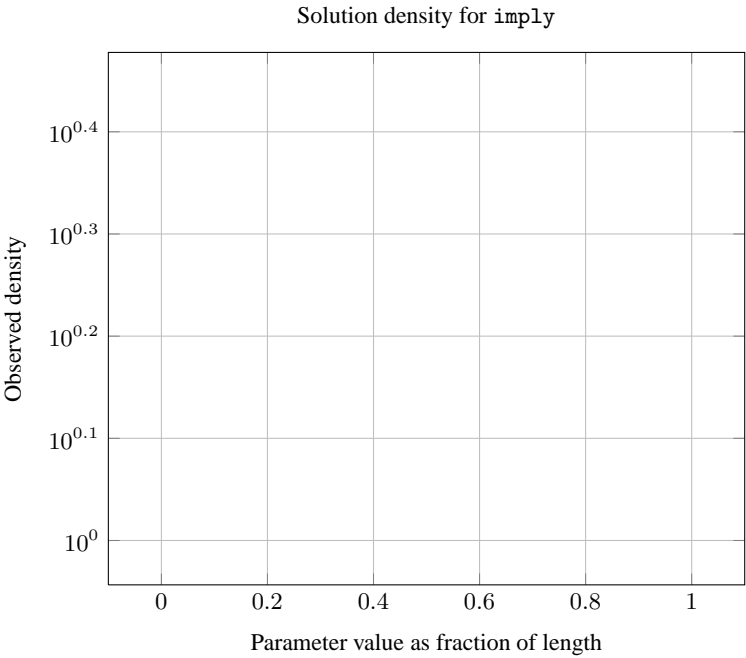
Functional dependency: VAR determined by VARIABLES.

Counting

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

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

Solution density for `imply`

Solution density for `imply`
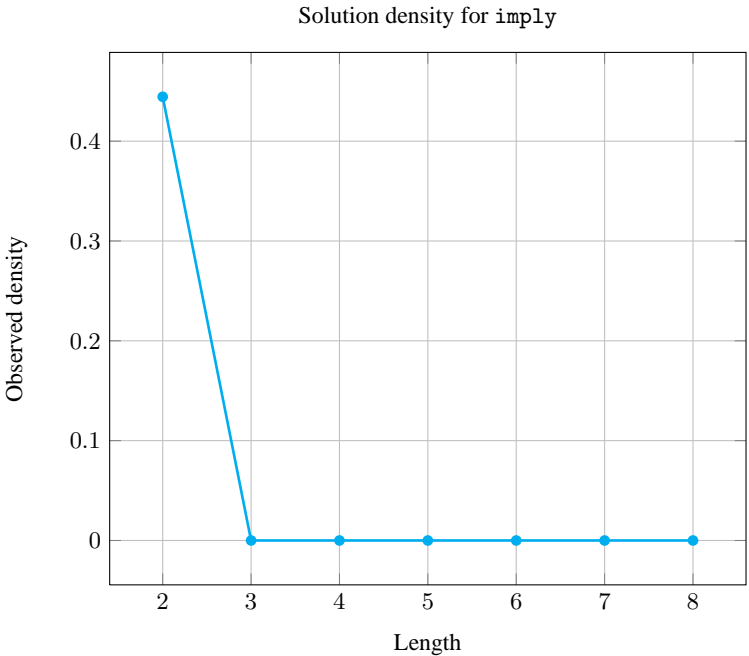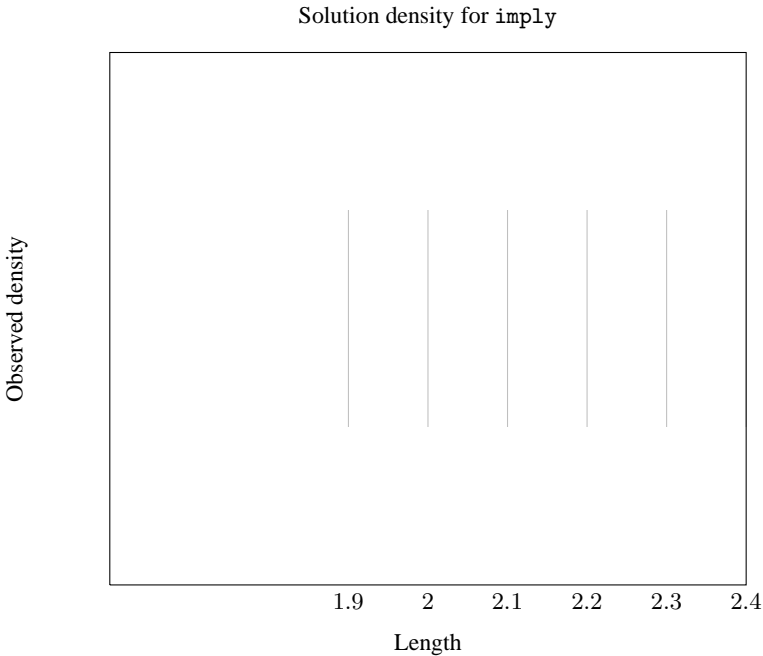
Length (n)		2
Total		4
Parameter	0	1
value	1	3

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

Solution density for `imply`

Observed density (vertical axis, 10^0 to $10^{0.4}$)

Parameter value as fraction of length (horizontal axis, 0 to 1)

Solution density for `imply`
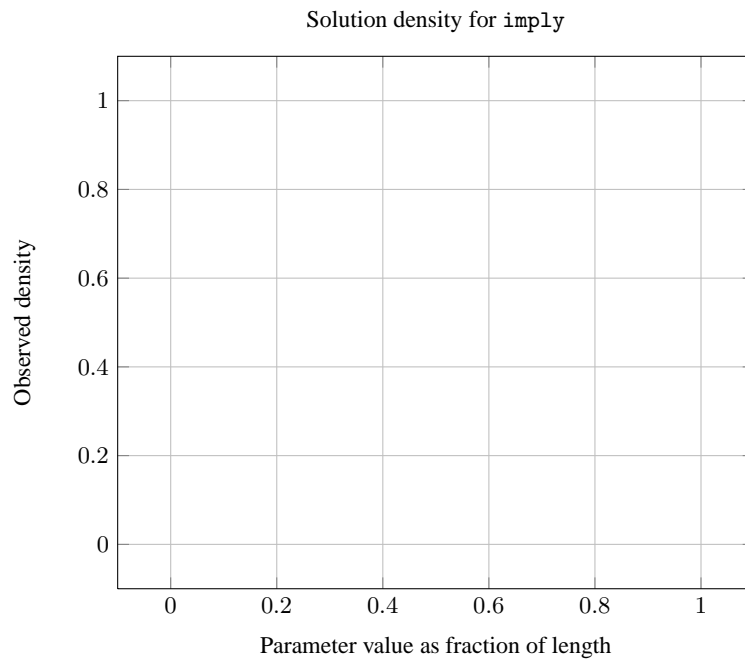
Systems `reifiedLeftImp` in **Choco**, `rel` in **Gecode**, `ifthenbool` in **JaCoP**, `#=>` in **SICStus**.

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

implies: `atleast_nvalue`, `soft_alldifferent_ctr`.

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 5.413 depicts the automaton associated with the imply constraint. To the first argument VAR of the imply constraint corresponds the first signature variable. To each variable VAR_i of the second argument VARIABLES of the imply constraint corresponds the next signature variable. There is no signature constraint.
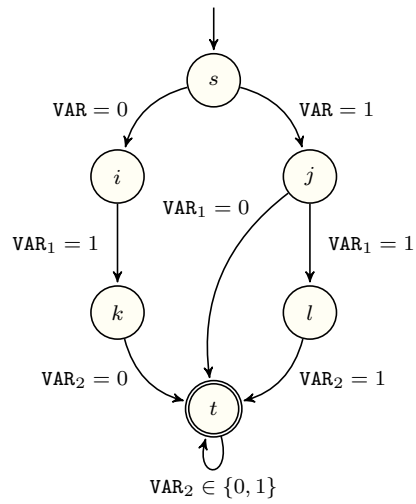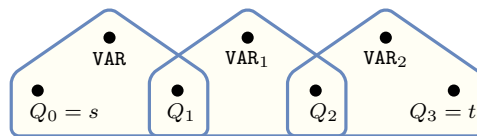

Figure 5.413: Automaton of the imply constraint

Figure 5.414: Hypergraph of the reformulation corresponding to the automaton of the imply constraint