5.151 imply

DESCRIPTION LINKS AUTOMATON

Origin Logic

Constraint imply(VAR, VARIABLES)

Synonyms rel, ifthen.

Arguments

VAR	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

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

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

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.

Systems reifiedLeftImp in **Choco**, rel in **Gecode**, ifthenbool in **JaCoP**, #=> in **SICStus**.

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

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

constraint network structure: Berge-acyclic constraint network.

constraint type: Boolean constraint.

filtering: arc-consistency.

Automaton

Figure 5.305 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 $\texttt{VAR}_i$ of the second argument `VARIABLES` of the `imply` constraint corresponds the next signature variable. There is no signature constraint.

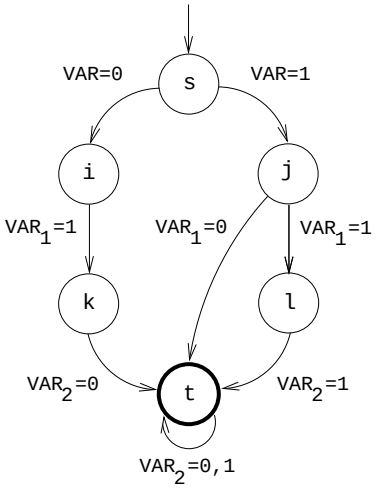

Figure 5.305: Automaton of the `imply` constraint

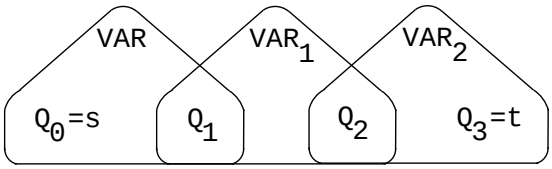

Figure 5.306: Hypergraph of the reformulation corresponding to the automaton of the `imply` constraint