

5.131 equivalent

	DESCRIPTION	LINKS	AUTOMATON
Origin	Logic		
Constraint	equivalent(VAR, VARIABLES)		
Synonym	eq.		
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 VAR_1, VAR_2 . Enforce $VAR = (VAR_1 \Leftrightarrow VAR_2)$.		
Example	$(1, \langle 0, 0 \rangle)$ $(0, \langle 0, 1 \rangle)$ $(0, \langle 1, 0 \rangle)$ $(1, \langle 1, 1 \rangle)$		
Symmetries	<ul style="list-style-type: none"> Items of VARIABLES are permutable. All occurrences of 0 in VAR and in VARIABLES.var can be set to 1. 		
Systems	ifOnlyIf in Choco , rel in Gecode , eqbool in JaCoP , #<=> in SICStus .		
See also	common keyword : and, imply, nand, nor, or, xor (<i>Boolean constraint</i>).		
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 5.261 depicts the automaton associated with the equivalent constraint. To the first argument VAR of the equivalent constraint corresponds the first signature variable. To each variable VAR_i of the second argument VARIABLES of the equivalent constraint corresponds the next signature variable. There is no signature constraint.

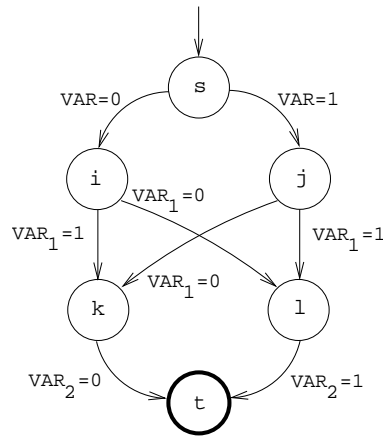


Figure 5.261: Automaton of the equivalent constraint

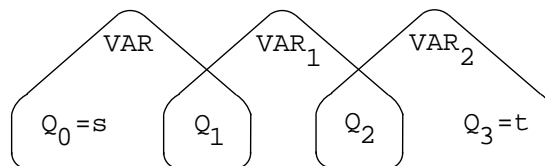


Figure 5.262: Hypergraph of the reformulation corresponding to the automaton of the equivalent constraint