

5.57 clause_or

	DESCRIPTION	LINKS	AUTOMATON
Origin	Logic		
Constraint	<code>clause_or(POSVARS, NEGVARs, VAR)</code>		
Synonym	<code>clause</code> .		
Arguments	POSVARS : <code>collection(var-dvar)</code> NEGVARs : <code>collection(var-dvar)</code> VAR : <code>dvar</code>		
Restrictions	$ POSVARS + NEGVARs > 0$ <code>required(POSVARS, var)</code> $POSVARS.var \geq 0$ $POSVARS.var \leq 1$ <code>required(NEGVARs, var)</code> $NEGVARs.var \geq 0$ $NEGVARs.var \leq 1$ $VAR \geq 0$ $VAR \leq 1$		
Purpose	Given a first collection of 0-1 variables $POSVARS = U_1, U_2, \dots, U_p$, a second collection of 0-1 variables $NEGVARs = V_1, V_2, \dots, V_n$, and a variable VAR, enforce $VAR = (U_1 \vee U_2 \vee \dots \vee U_p) \vee (\neg V_1 \vee \neg V_2 \vee \dots \vee \neg V_n)$.		
Example	<code>((0, 0), (0), 1)</code>		
Typical	$ POSVARS + NEGVARs > 1$		
Symmetries	<ul style="list-style-type: none"> Items of POSVARS are permutable. Items of NEGVARs are permutable. 		
Remark	The <code>clause_or</code> constraint is called <code>clause</code> in Gecode (http://www.gecode.org/).		
Systems	<code>reifiedOr</code> in Choco , <code>clause</code> in Choco , <code>clause</code> in Gecode .		
See also	common keyword: clause_and, or (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 . modelling: disjunction .		

Automaton

Figure 5.126 depicts the automaton associated with the `clause_or` constraint:

- To the argument `VAR` of the `clause_or` constraint corresponds the first signature variable.
- To each variable of the argument `POSVARS` corresponds a signature variable.
- Finally, to each variable VAR_i of the argument `NEGVARS` corresponds a signature variable that is the negation of VAR_i .

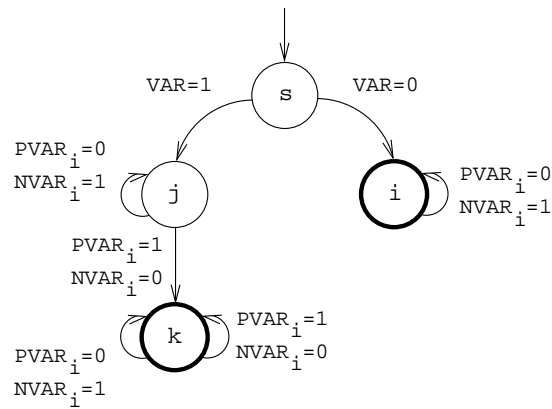


Figure 5.126: Automaton of the `clause_or` constraint ($PVAR_i$ and $NVAR_i$ respectively denote variables of `POSVARS` and `NEGVARS`)

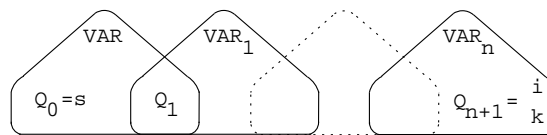


Figure 5.127: Hypergraph of the reformulation corresponding to the automaton of the `clause_or` constraint