

5.69 clause_and

	DESCRIPTION	LINKS	AUTOMATON
Origin	Logic		
Constraint	clause_and(POSVARS, NEGVARs, VAR)		
Synonym	clause.		
Arguments	POSVARS : collection(var-dvar) NEGVARs : collection(var-dvar) VAR : dvar		
Restrictions	$ POSVARS + NEGVARs > 0$ required(POSVARS, var) $POSVARS.var \geq 0$ $POSVARS.var \leq 1$ required(NEGVARs, var) $NEGVARs.var \geq 0$ $NEGVARs.var \leq 1$ $VAR \geq 0$ $VAR \leq 1$		
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 $VAR = (U_1 \wedge U_2 \wedge \dots \wedge U_p) \wedge (\neg V_1 \wedge \neg V_2 \wedge \dots \wedge \neg V_n)$.		
Example	$((1, 0), (0), 0)$		
Typical	$ POSVARS + NEGVARs > 1$		
Symmetries	<ul style="list-style-type: none"> Items of POSVARS are permutable. Items of NEGVARs are permutable. 		
Arg. properties	<ul style="list-style-type: none"> Extensible wrt. POSVARS when VAR = 0. Extensible wrt. NEGVARs when VAR = 0. 		
Remark	The clause_or constraint is called clause in Gecode (http://www.gecode.org/).		
Systems	reifiedAnd in Choco , clause in Choco , clause in Gecode .		
See also	common keyword : and, clause_or (<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 5.171 depicts the automaton associated with the `clause_and` constraint:

- To the argument `VAR` of the `clause_and` 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 .

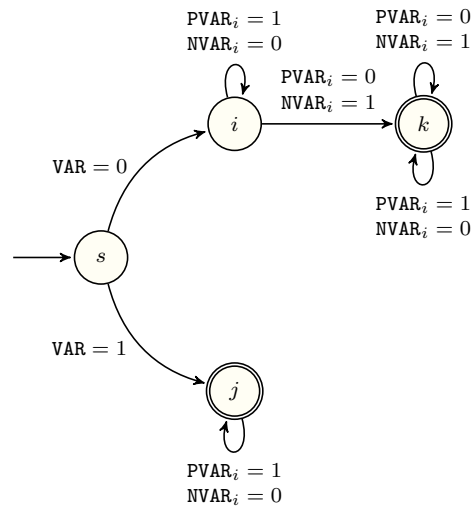


Figure 5.171: Automaton of the `clause_and` constraint (PVAR_i and NVAR_i respectively denote variables of `POSVARS` and `NEGVARS`)

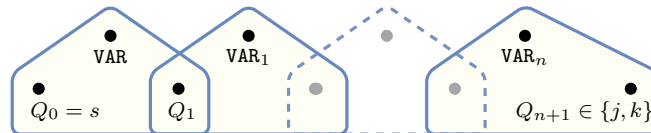


Figure 5.172: Hypergraph of the reformulation corresponding to the automaton of the `clause_and` constraint ($\text{VAR}_1, \dots, \text{VAR}_n$ denotes $\text{PVAR}_1, \dots, \text{PVAR}_{|\text{POSVARS}|}, 1 - \text{NVAR}_1, \dots, 1 - \text{NVAR}_{|\text{NEGVARS}|}$)

