

5.56 clause_and

| | DESCRIPTION | LINKS | AUTOMATON |
|---------------------|--|-------|-----------|
| Origin | Logic | | |
| Constraint | <code>clause_and(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 $VAR = (U_1 \wedge U_2 \wedge \dots \wedge U_p) \wedge (\neg V_1 \wedge \neg V_2 \wedge \dots \wedge \neg V_n)$. | | |
| Example | $((\langle 1, 0 \rangle, \langle 0 \rangle, 0)$ | | |
| 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>reifiedAnd</code> in Choco , <code>clause</code> in Choco , <code>clause</code> in Gecode . | | |
| See also | common keyword: and , <code>clause_or</code> (<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 5.124 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 .

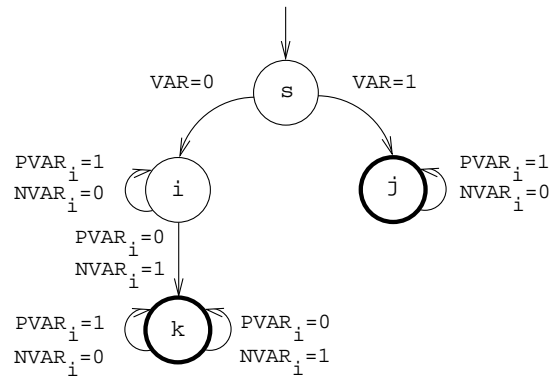


Figure 5.124: Automaton of the `clause_and` constraint ($PVAR_i$ and $NVAR_i$ respectively denote variables of `POSVARS` and `NEGVARS`)

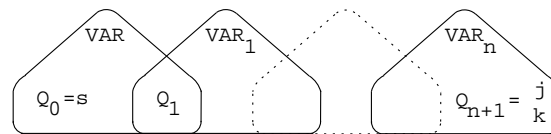


Figure 5.125: Hypergraph of the reformulation corresponding to the automaton of the `clause_and` constraint