5.155 exactly

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from atleast and atmost.

Constraint exactly(N, VARIABLES, VALUE)

Synonym count.

Arguments

N	:	int
VARIABLES	:	collection(var−dvar)
VALUE	:	int

Restrictions

$N \geq 0$
$N \leq |\text{VARIABLES}|$
required(VARIABLES, var)

Purpose Exactly N variables of the VARIABLES collection are assigned value VALUE.

Example $(2, \langle 4, 2, 4, 5\rangle, 4)$

The exactly constraint holds since exactly $N = 2$ variables of the $\text{VARIABLES} = \langle 4, 2, 4, 5\rangle$ collection are assigned value $\text{VALUE} = 4$.

Typical

$N > 0$
$N < |\text{VARIABLES}|$
$|\text{VARIABLES}| > 1$

Symmetries

- Items of VARIABLES are permutable.
- An occurrence of a value of VARIABLES.var that is different from VALUE can be replaced by any other value that is also different from VALUE.

Arg. properties

- Functional dependency: N determined by VARIABLES and VALUE.
- Aggregate: N(+), VARIABLES(union), VALUE(id).

Systems occurence in **Choco**, count in **Gecode**, exactly in **Gecode**, count in **JaCoP**, exactly in **MiniZinc**, count in **SICStus**.

See also **generalisation:** among (constant *replaced by* variable *and* value *replaced by* list *of* values).

implies: atleast (= N *replaced by* $\geq$ N), atmost (= N *replaced by* $\leq$ N).

Keywords

characteristic of a constraint: automaton, automaton with counters.
constraint arguments: reverse of a constraint, pure functional dependency.
constraint network structure: alpha-acyclic constraint network(2).
constraint type: value constraint, counting constraint.
filtering: glue matrix, arc-consistency.
modelling: functional dependency.

Arc input(s)	VARIABLES
Arc generator	$SELF \mapsto$ collection(variables)
Arc arity	1
Arc constraint(s)	variables.var = VALUE
Graph property(ies)	NARC= N

Graph model Since each arc constraint involves only one vertex (VALUE is fixed), we employ the $SELF$ arc generator in order to produce a graph with a single loop on each vertex.

Parts (A) and (B) of Figure 5.349 respectively show the initial and final graph associated with the **Example** slot. Since we use the NARC graph property, the loops of the final graph are stressed in bold. The exactly constraint holds since exactly two variables are assigned value 4.

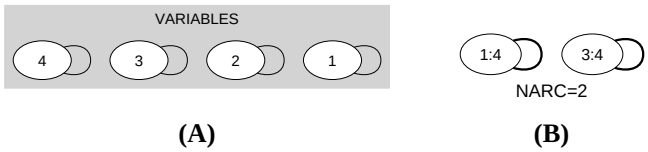
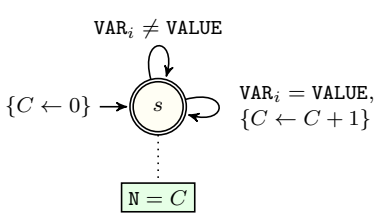
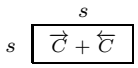


Figure 5.349: Initial and final graph of the exactly constraint

Automaton

Figure 5.350 depicts the automaton associated with the `exactly` constraint. To each variable $\texttt{VAR}_i$ of the collection `VARIABLES` corresponds a 0-1 signature variable S_i. The following signature constraint links $\texttt{VAR}_i$ and S_i: $\texttt{VAR}_i = \texttt{VALUE} \Leftrightarrow S_i$.





Glue matrix where $\overrightarrow{C}$ and $\overleftarrow{C}$ resp. represent the counter value C at the end of a prefix and at the end of the corresponding reverse suffix that partitions the sequence `VARIABLES`.

Figure 5.350: Automaton (with one counter) of the `exactly` constraint and its glue matrix

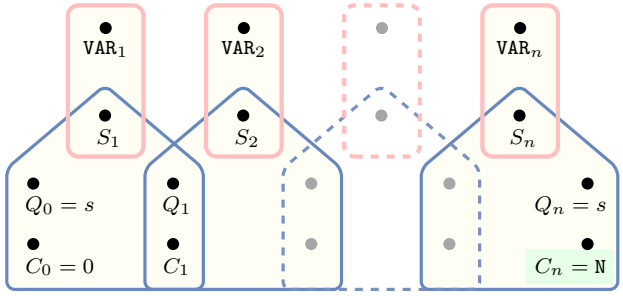


Figure 5.351: Hypergraph of the reformulation corresponding to the automaton (with one counter) of the `exactly` constraint: since all states variables $Q_0, Q_1, \ldots, Q_n$ are fixed to the unique state s of the automaton, the transitions constraints share only the counter variable C and the constraint network is Berge-acyclic