5.31 arith

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Used in the definition of several automata

Constraint arith(VARIABLES, RELOP, VALUE)

Synonym rel.

Arguments

VARIABLES	:	collection(var−dvar)
RELOP	:	atom
VALUE	:	int

Restrictions

required(VARIABLES, var)
RELOP $\in [=, \neq, <, \geq, >, \leq]$

Purpose Enforce for all variables var of the VARIABLES collection to have var RELOP VALUE.

Example $(\langle 4, 5, 7, 4, 5\rangle, <, 9)$

The arith constraint holds since all values of the collection $\langle 4, 5, 7, 4, 5\rangle$ are strictly less than 9.

All solutions Figure 5.84 gives all solutions to the following non ground instance of the arith constraint: $V_1 \in [0, 5]$, $V_2 \in [2, 3]$, $V_3 \in [2, 4]$, $V_4 \in [1, 6]$, arith$(\langle V_1, V_2, V_3, V_4\rangle, \leq, 2)$.

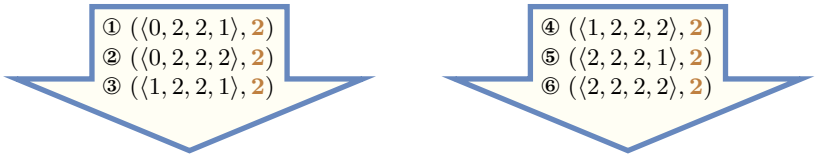

Figure 5.84: All solutions corresponding to the non ground example of the arith constraint of the **All solutions** slot

Typical

|VARIABLES| > 1
RELOP $\in [=]$

Symmetries

- Items of VARIABLES are permutable.
- An occurrence of a value of VARIABLES.var can be replaced by any value of VARIABLES.var.

Arg. properties

Contractible wrt. VARIABLES.

Systems eq in **Choco**, neq in **Choco**, geq in **Choco**, gt in **Choco**, leq in **Choco**, lt in **Choco**, rel in **Gecode**, #< in **SICStus**, #=< in **SICStus**, #> in **SICStus**, #>= in **SICStus**, #= in **SICStus**, #\= in **SICStus**.

Used in arith_sliding.

See also **common keyword:** among, count *(value constraint)*.

generalisation: arith_or *(*variable RELOP VALUE *replaced by* variable RELOP VALUE ∨ variable RELOP VALUE*)*.

system of constraints: arith_sliding.

Keywords **characteristic of a constraint:** automaton, automaton without counters, reified automaton constraint.

constraint network structure: Berge-acyclic constraint network.

constraint type: decomposition, value constraint.

filtering: arc-consistency.

modelling: domain definition.

Cond. implications arith(VARIABLES, RELOP, VALUE)
with RELOP ∈ [<]
and minval(VARIABLES.var) ≥ 0
implies range_ctr(VARIABLES, CTR, R)
when CTR ∈ [<].

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

Graph model Parts (A) and (B) of Figure 5.85 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.

(A) **(B)**

Figure 5.85: Initial and final graph of the arith constraint

Automaton

Figure 5.86 depicts the automaton associated with the `arith` 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: $\texttt{VAR}_i$ `RELOP VALUE` $\Leftrightarrow S_i$. The automaton enforces for each variable $\texttt{VAR}_i$ the condition $\texttt{VAR}_i$ `RELOP VALUE`.

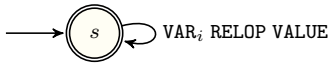

Figure 5.86: Automaton of the `arith` constraint

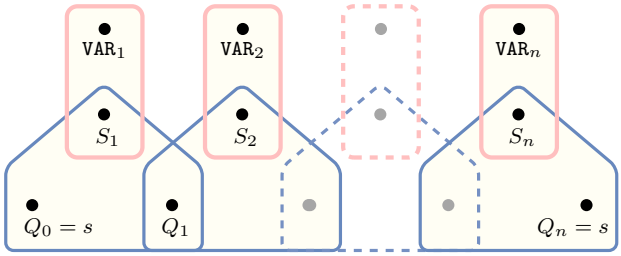

Figure 5.87: Hypergraph of the reformulation corresponding to the automaton of the `arith` constraint