5.33 arith_sliding

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Used in the definition of some automaton

Constraint `arith_sliding(VARIABLES, RELOP, VALUE)`

Arguments

```
VARIABLES  :  collection(var-dvar)
RELOP      :  atom
VALUE      :  int
```

Restrictions

`required(VARIABLES, var)`

RELOP $\in [=, \neq, <, \geq, >, \leq]$

Purpose Enforce for all sequences of variables $\mathtt{var}_1, \mathtt{var}_2, \ldots, \mathtt{var}_i$ $(1 \leq i \leq |\mathtt{VARIABLES}|)$ of the VARIABLES collection to have $(\mathtt{var}_1 + \mathtt{var}_2 + \cdots + \mathtt{var}_i)$ RELOP VALUE.

Example $(\langle 0, 0, 1, 2, 0, 0, -3\rangle, <, 4)$

The arith_sliding constraint holds since all the following seven inequalities hold:

- $0 < 4$,
- $0 + 0 < 4$,
- $0 + 0 + 1 < 4$,
- $0 + 0 + 1 + 2 < 4$,
- $0 + 0 + 1 + 2 + 0 < 4$,
- $0 + 0 + 1 + 2 + 0 + 0 < 4$,
- $0 + 0 + 1 + 2 + 0 + 0 - 3 < 4$.

Typical

$|\mathtt{VARIABLES}| > 1$

RELOP $\in [<, \geq, >, \leq]$

Arg. properties

- Contractible wrt. VARIABLES when RELOP $\in [<, \leq]$ and $\mathtt{minval(VARIABLES.var)} \geq 0$.
- Suffix-contractible wrt. VARIABLES.

See also

common keyword: sum_ctr *(arithmetic constraint)*.

implies: sum_ctr.

part of system of constraints: arith.

used in graph description: arith.

Keywords

characteristic of a constraint: hypergraph, automaton, automaton with counters.

combinatorial object: sequence.

constraint type: arithmetic constraint, decomposition, sliding sequence constraint.

Arc input(s)	VARIABLES
Arc generator	$\mathit{PATH\_1} \mapsto \texttt{collection}$
Arc arity	$*$
Arc constraint(s)	$\texttt{arith}(\texttt{collection}, \texttt{RELOP}, \texttt{VALUE})$
Graph property(ies)	$\mathbf{NARC} = \vert\texttt{VARIABLES}\vert$

Automaton Figure 5.78 depicts the automaton associated with the `arith_sliding` constraint. To each item of the collection `VARIABLES` corresponds a signature variable S_i that is equal to 0.

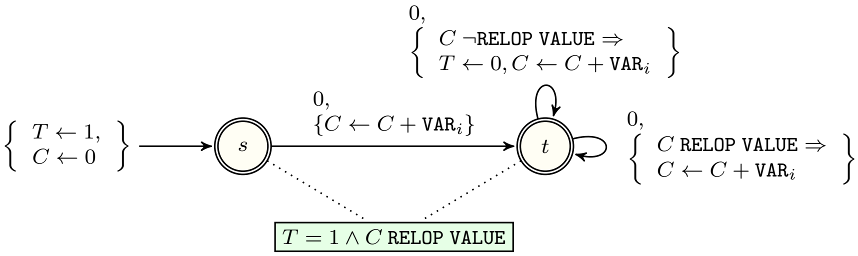
$\left\{ \begin{array}{l} T \leftarrow 1, \\ C \leftarrow 0 \end{array} \right\}$
s
t
$0, \{C \leftarrow C + \texttt{VAR}_i\}$
$0, \left\{ \begin{array}{l} C \neg\texttt{RELOP VALUE} \Rightarrow \\ T \leftarrow 0, C \leftarrow C + \texttt{VAR}_i \end{array} \right\}$
$0, \left\{ \begin{array}{l} C\ \texttt{RELOP VALUE} \Rightarrow \\ C \leftarrow C + \texttt{VAR}_i \end{array} \right\}$
$T = 1 \wedge C\ \texttt{RELOP VALUE}$

Figure 5.78: Automaton of the `arith_sliding` constraint (T is initially set to 1 and reset to 0 as soon as one of the sliding constraints does not hold)

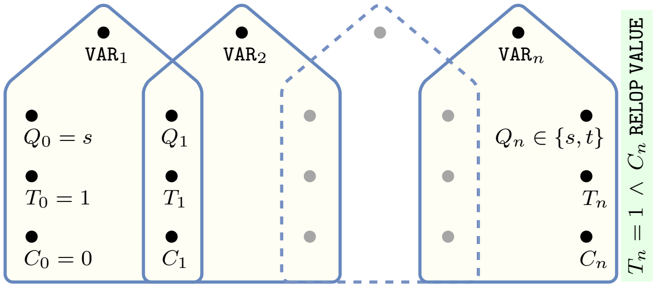

Figure 5.79: Hypergraph of the reformulation corresponding to the automaton (with two counters) of the `arith_sliding` constraint (since all states of the automaton are accepting there is no restriction on the last variable Q_n)