5.227 lex_chain_lesseq

DESCRIPTION LINKS GRAPH

Origin [95]

Constraint lex_chain_lesseq(VECTORS)

Usual name lex_chain

Type VECTOR : collection(var−dvar)

Argument VECTORS : collection(vec − VECTOR)

Restrictions
$|\texttt{VECTOR}| \geq 1$
required(VECTOR, var)
required(VECTORS, vec)
same_size(VECTORS, vec)

Purpose For each pair of consecutive vectors $\texttt{VECTOR}_i$ and $\texttt{VECTOR}_{i+1}$ of the VECTORS collection we have that $\texttt{VECTOR}_i$ is lexicographically less than or equal to $\texttt{VECTOR}_{i+1}$. Given two vectors, $\vec{X}$ and $\vec{Y}$ of n components, $\langle X_0, \ldots, X_{n-1} \rangle$ and $\langle Y_0, \ldots, Y_{n-1} \rangle$, $\vec{X}$ is *lexicographically less than or equal to* $\vec{Y}$ if and only if $n = 0$ or $X_0 < Y_0$ or $X_0 = Y_0$ and $\langle X_1, \ldots, X_{n-1} \rangle$ is lexicographically less than or equal to $\langle Y_1, \ldots, Y_{n-1} \rangle$.

Example $(\langle \texttt{vec} - \langle 5,2,3,9 \rangle, \texttt{vec} - \langle 5,2,6,2 \rangle, \texttt{vec} - \langle 5,2,6,2 \rangle \rangle)$

The lex_chain_lesseq constraint holds since:

- The first vector $\langle 5,2,3,9 \rangle$ of the VECTORS collection is lexicographically less than or equal to the second vector $\langle 5,2,6,2 \rangle$ of the VECTORS collection.
- The second vector $\langle 5,2,6,2 \rangle$ of the VECTORS collection is lexicographically less than or equal to the third vector $\langle 5,2,6,2 \rangle$ of the VECTORS collection.

Typical
$|\texttt{VECTOR}| > 1$
$|\texttt{VECTORS}| > 1$

Arg. properties

- Contractible wrt. VECTORS.
- Suffix-contractible wrt. VECTORS.vec (*remove items from same position*).

Usage This constraint was motivated for breaking symmetry: more precisely when one wants to lexicographically order the consecutive columns of a matrix of decision variables. A further motivation is that using a set of lexicographic ordering constraints between two vectors does usually not allows to come up with a complete pruning.

Algorithm A filtering algorithm achieving arc-consistency for a chain of lexicographical ordering constraints is presented in [95].

Six different ways of integrating a chain of lexicographical ordering constraints within non-overlapping constraints like `diffn` or `geost` and within their corresponding necessary condition like the `cumulative` constraint are shown in [3].

Systems `lexChainEq` in **Choco**, `lex_chain` in **SICStus**.

See also **common keyword:** `allperm` *(lexicographic order)*, `geost` *(symmetry, lexicographic ordering on the origins of* `tasks`, `rectangles`, *...)*, `lex_between`, `lex_greater`, `lex_greatereq`, `lex_less` *(lexicographic order)*.

implied by: `lex2` *(columns lex ordering imposed by constraint* `lex2` *removed)*, `lex_chain_less` *(non-strict order implied by strict order)*, `ordered_atleast_nvector` *(*`NVEC` *of constraint* `ordered_atleast_nvector` *removed)*, `ordered_atmost_nvector` *(*`NVEC` *of constraint* `ordered_atmost_nvector` *removed)*, `ordered_nvector` *(*`NVEC` *of constraint* `ordered_nvector` *removed)*.

part of system of constraints: `lex_lesseq`.

related: `cumulative`, `diffn` *(lexicographic ordering on the origins of* `tasks`, `rectangles`, *...)*.

system of constraints: `lex2`.

used in graph description: `lex_lesseq`.

Keywords **characteristic of a constraint:** vector.

constraint type: system of constraints, decomposition, order constraint.

filtering: arc-consistency.

heuristics: heuristics and lexicographical ordering.

symmetry: symmetry, matrix symmetry, lexicographic order.

Arc input(s)	VECTORS
Arc generator	$PATH \mapsto$ collection(vectors1, vectors2)
Arc arity	2
Arc constraint(s)	lex_lesseq(vectors1.vec, vectors2.vec)
Graph property(ies)	$\mathbf{NARC} = \lvert\text{VECTORS}\rvert - 1$

Graph model Parts (A) and (B) of Figure 5.500 respectively show the initial and final graph associated with the **Example** slot. Since we use the $\mathbf{NARC}$ graph property, the arcs of the final graph are stressed in bold. The lex_chain_lesseq constraint holds since all the arc constraints of the initial graph are satisfied.

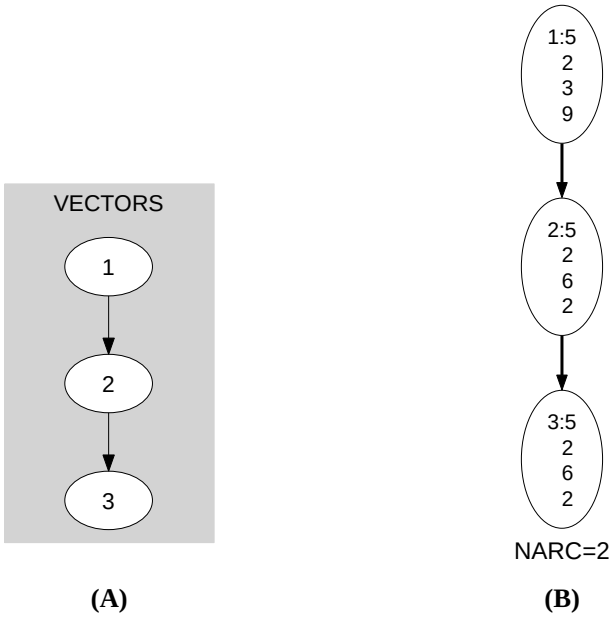

Figure 5.500: Initial and final graph of the lex_chain_lesseq constraint

Signature Since we use the $PATH$ arc generator on the VECTORS collection the number of arcs of the initial graph is equal to $\lvert\text{VECTORS}\rvert - 1$. For this reason we can rewrite $\mathbf{NARC} = \lvert\text{VECTORS}\rvert - 1$ to $\mathbf{NARC} \geq \lvert\text{VECTORS}\rvert - 1$ and simplify $\overline{\mathbf{NARC}}$ to $\overline{\mathbf{NARC}}$.