5.223 lex_chain_greater

DESCRIPTION LINKS GRAPH

Origin Derived from lex_chain_less

Constraint lex_chain_greater(VECTORS)

Usual name lex_chain

Type VECTOR : collection(var−dvar)

Argument VECTORS : collection(vec − VECTOR)

Restrictions
|VECTOR| ≥ 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 strictly greater than $\texttt{VECTOR}_{i+1}$. 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 strictly greater than* $\vec{Y}$ if and only if $X_0 > Y_0$ or $X_0 = Y_0$ and $\langle X_1, \ldots, X_{n-1}\rangle$ is lexicographically strictly greater than $\langle Y_1, \ldots, Y_{n-1}\rangle$.

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

The lex_chain_greater constraint holds since:

- The first vector $\langle 5,2,6,3\rangle$ of the VECTORS collection is lexicographically strictly greater than 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 strictly greater than the third vector $\langle 5,2,3,9\rangle$ of the VECTORS collection.

Typical
|VECTOR| > 1
|VECTORS| > 1

Arg. properties

- Contractible wrt. VECTORS.
- Suffix-extensible wrt. VECTORS.vec (*add items at 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].

See also

common keyword: lex_between, lex_greatereq, lex_less, lex_lesseq *(lexicographic order)*.

implies: lex_alldifferent, lex_chain_greatereq.

part of system of constraints: lex_greater.

used in graph description: lex_greater.

Keywords

application area: floor planning problem.

characteristic of a constraint: vector.

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

filtering: arc-consistency.

heuristics: heuristics and lexicographical ordering.

modelling: degree of diversity of a set of solutions.

modelling exercises: degree of diversity of a set of solutions.

symmetry: symmetry, matrix symmetry, lexicographic order.

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

Graph model Parts (A) and (B) of Figure 5.478 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold. The lex_chain_greater constraint holds since all the arc constraints of the initial graph are satisfied.

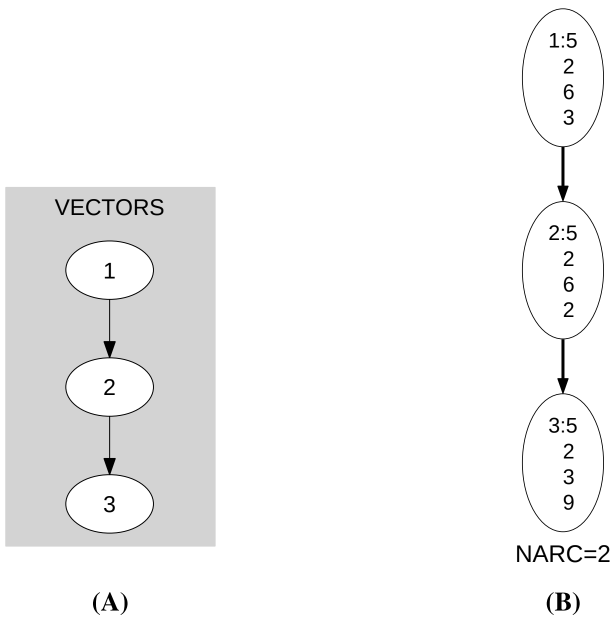

Figure 5.478: Initial and final graph of the lex_chain_greater constraint

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