## 5.220 lex2

DESCRIPTION LINKS

**Origin** [168]

**Constraint** lex2(MATRIX)

**Synonyms** double_lex, row_and_column_lex.

**Type** VECTOR : collection(var−dvar)

**Argument** MATRIX : collection(vec − VECTOR)

**Restrictions**
|VECTOR| ≥ 1
required(VECTOR, var)
required(MATRIX, vec)
same_size(MATRIX, vec)

**Purpose** Given a matrix of domain variables, enforces that both adjacent rows, and adjacent columns are lexicographically ordered (adjacent rows and adjacent columns can be equal).

**Example** (⟨vec − ⟨2, 2, 3⟩ , vec − ⟨2, 3, 1⟩⟩)

The lex2 constraint holds since:

- The first row ⟨2, 2, 3⟩ is lexicographically less than or equal to the second row ⟨2, 3, 1⟩.
- The first column ⟨2, 2⟩ is lexicographically less than or equal to the second column ⟨2, 3⟩.
- The second column ⟨2, 3⟩ is lexicographically less than or equal to the third column ⟨3, 1⟩.

**Typical**
|VECTOR| > 1
|MATRIX| > 1

**Symmetry** One and the same constant can be added to the var attribute of all items of MATRIX.vec.

**Usage** A *symmetry-breaking* constraint.

**Remark** The idea of this *symmetry-breaking* constraint can already be found in the following articles of A. Lubiw [267, 268].

In block designs you sometimes want repeated blocks, so using the non-strict order would be required in this case.

**Reformulation** The `lex2` constraint can be expressed as a conjunction of two `lex_chain_lesseq` constraints: A first `lex_chain_lesseq` constraint on the `MATRIX` argument and a second `lex_chain_lesseq` constraint on the transpose of the `MATRIX` argument.

**Systems** `lex2` in **MiniZinc**.

**See also** **common keyword:** `allperm`, `lex_lesseq` (*matrix symmetry*,*lexicographic order*).

**implied by:** `strict_lex2`.

**implies:** `lex_chain_lesseq`.

**part of system of constraints:** `lex_chain_lesseq`.

**Keywords** **constraint type:** predefined constraint, system of constraints, order constraint.

**modelling:** matrix, matrix model.

**symmetry:** symmetry, matrix symmetry, lexicographic order.