## 5.220 lex_alldifferent

DESCRIPTION LINKS GRAPH

**Origin** J. Pearson

**Constraint** lex_alldifferent(VECTORS)

**Synonyms** lex_alldiff, lex_alldistinct, alldiff_on_tuples, alldifferent_on_tuples, alldistinct_on_tuples.

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

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

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

**Purpose** All the vectors of the collection VECTORS are distinct. Two vectors $(u_1, u_2, \ldots, u_n)$ and $(v_1, v_2, \ldots, v_n)$ are distinct if and only if there exists $i \in [1, n]$ such that $u_i \neq v_i$.

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

The lex_alldifferent constraint holds since:

- The first vector $\langle 5, 2, 3\rangle$ and the second vector $\langle 5, 2, 6\rangle$ of the VECTORS collection differ in their third component (i.e., $3 \neq 6$).
- The first vector $\langle 5, 2, 3\rangle$ and the third vector $\langle 5, 3, 3\rangle$ of the VECTORS collection differ in their second component (i.e., $2 \neq 3$).
- The second vector $\langle 5, 2, 6\rangle$ and the third vector $\langle 5, 3, 3\rangle$ of the VECTORS collection differ in their second and third components (i.e., $2 \neq 3$ and $6 \neq 3$).

**Typical**
$|\text{VECTOR}| > 1$
$|\text{VECTORS}| > 1$

**Symmetries**
- Items of VECTORS are permutable.
- Items of VECTORS.vec are permutable (*same permutation used*).
- All occurrences of two distinct tuples of values of VECTORS.vec can be swapped; all occurrences of a tuple of values of VECTORS.vec can be renamed to any unused tuple of values.

**Arg. properties**
- Contractible wrt. VECTORS.
- Extensible wrt. VECTORS.vec (*add items at same position*).

**Usage** When the vectors have two components, the `lex_alldifferent` constraint allows to directly enforce difference constraints between pairs of variables. Such difference constraints occur for instance in block design problems (e.g., Steiner triples, Kirkman schoolgirls problem). However, in all these problems a same variable may occur in more than one pair of variables. Consequently, arc-consistency is not achieved any more by the filtering algorithm described in [335].

**Algorithm** A filtering algorithm achieving arc-consistency for the `lex_alldifferent` constraint is proposed by C.-G. Quimper and T. Walsh in [335] and a longer version is available in [336] and in [337].

**Reformulation** The `lex_alldifferent(VECTORS)` constraint can be expressed as a clique of `lex_different` constraints. By associating a $n$-dimensional box for which all sizes are equal to 1, one can also express the `lex_alldifferent(VECTORS)` constraint as a `diffn` or a `geost` constraint. Enforcing all the $n$-dimensional boxes to not overlap is equivalent as enforcing all the vectors to be distinct. In the context of the multidimensional sweep algorithm of the `geost` constraint [38], it makes more sense to make a complete sweep over the domain of each variable in order not to only restrict the minimum and maximum value of each variable.

**See also**

**generalisation:** `diffn` (`vector` *replaced by* `orthotope`), `geost` (`vector` *replaced by* `object`).

**implied by:** `all_incomparable`, `lex_chain_greater`, `lex_chain_less`.

**implies:** `lex_alldifferent_except_0`.

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

**specialisation:** `alldifferent` (`vector` *replaced by* `variable`).

**used in graph description:** `lex_different`.

**Keywords**

**characteristic of a constraint:** vector.

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

**filtering:** bipartite matching, arc-consistency.

**modelling:** difference between pairs of variables.

| | |
|---|---|
| **Arc input(s)** | VECTORS |
| **Arc generator** | $CLIQUE(<) \mapsto$ collection(vectors1, vectors2) |
| **Arc arity** | 2 |
| **Arc constraint(s)** | lex_different(vectors1.vec, vectors2.vec) |
| **Graph property(ies)** | $\mathbf{NARC} = |\text{VECTORS}| * (|\text{VECTORS}| - 1)/2$ |

**Graph model** Parts (A) and (B) of Figure 5.475 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.

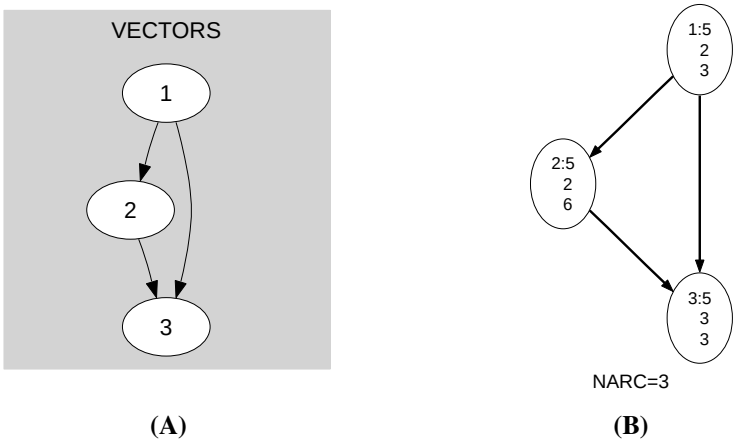


Figure 5.475: Initial and final graph of the lex_alldifferent constraint

**Signature** Since we use the $CLIQUE(<)$ arc generator on the VECTORS collection the number of arcs of the initial graph is equal to $|\text{VECTORS}| \cdot (|\text{VECTORS}| - 1)/2$. For this reason we can rewrite $\mathbf{NARC} = |\text{VECTORS}| \cdot (|\text{VECTORS}| - 1)/2$ to $\mathbf{NARC} \geq |\text{VECTORS}| \cdot (|\text{VECTORS}| - 1)/2$ and simplify $\overline{\mathbf{NARC}}$ to $\mathbf{NARC}$.