5.114 derangement

DESCRIPTION LINKS GRAPH

Origin Derived from `cycle`.

Constraint `derangement(NODES)`

Argument `NODES` : `collection(index−int, succ−dvar)`

Restrictions
$|\texttt{NODES}| > 1$
`required(NODES, [index, succ])`
$\texttt{NODES.index} \geq 1$
$\texttt{NODES.index} \leq |\texttt{NODES}|$
`distinct(NODES, index)`
$\texttt{NODES.succ} \geq 1$
$\texttt{NODES.succ} \leq |\texttt{NODES}|$

Purpose Enforce to have a permutation with no cycle of length one. The permutation is depicted by the `succ` attribute of the `NODES` collection.

Example

$$\left(\left\langle\begin{array}{ll} \texttt{index}-1 & \texttt{succ}-2, \\ \texttt{index}-2 & \texttt{succ}-1, \\ \texttt{index}-3 & \texttt{succ}-5, \\ \texttt{index}-4 & \texttt{succ}-3, \\ \texttt{index}-5 & \texttt{succ}-4 \end{array}\right\rangle\right)$$

In the permutation of the example we have the following 2 cycles: $1 \rightarrow 2 \rightarrow 1$ and $3 \rightarrow 5 \rightarrow 4 \rightarrow 3$. Since these cycles have both a length strictly greater than one the corresponding `derangement` constraint holds.

Typical $|\texttt{NODES}| > 2$

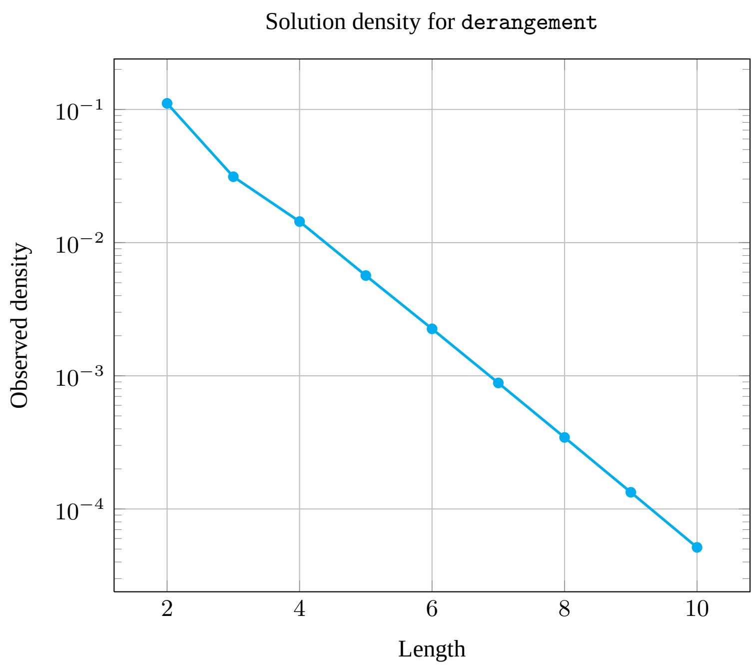
Symmetries
- Items of `NODES` are permutable.
- Attributes of `NODES` are permutable w.r.t. permutation (`index`, `succ`) (*permutation applied to all items*).

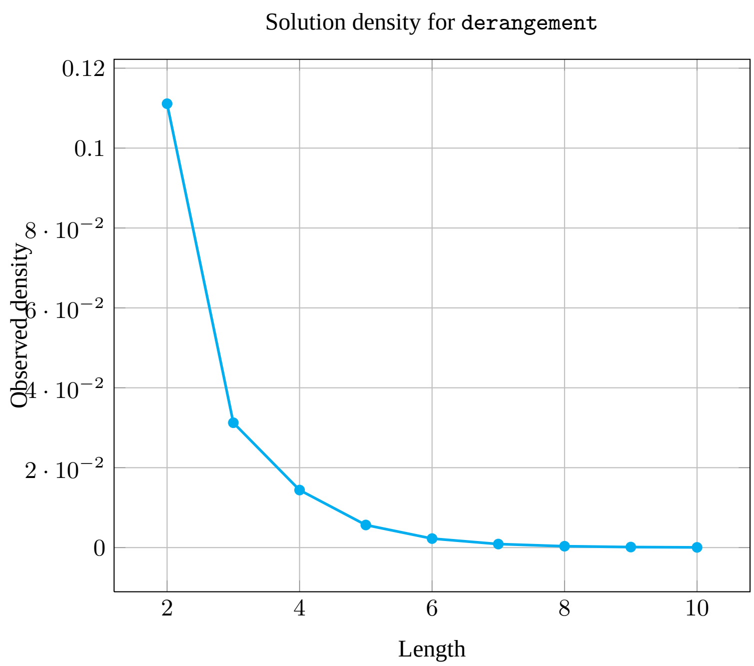
Remark A special case of the `cycle` [41] constraint.

Counting

Length (n)	2	3	4	5	6	7	8	9	10
Solutions	1	2	9	44	265	1854	14833	133496	1334961

Number of solutions for `derangement`: domains $0..n$





See also **common keyword:** `alldifferent`, `cycle` *(permutation)*.

implied by: `symmetric_alldifferent`.

implies: `twin`.

implies (items to collection): `k_alldifferent`, `lex_alldifferent`.

Keywords

characteristic of a constraint: sort based reformulation.

combinatorial object: permutation.

constraint type: graph constraint.

filtering: arc-consistency, DFS-bottleneck.

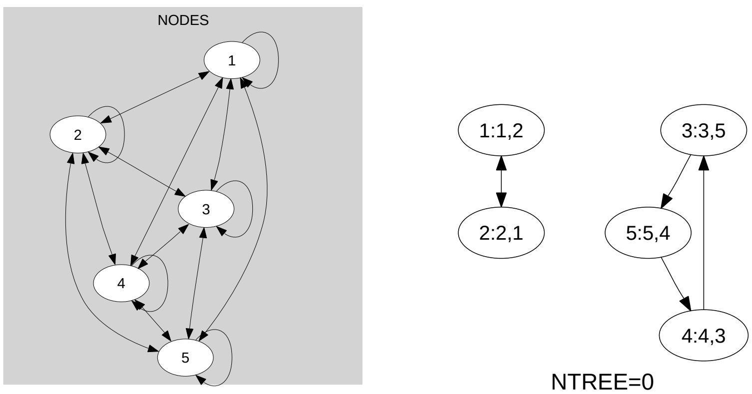
final graph structure: one_succ.

Cond. implications

`derangement`(NODES)
implies `permutation`(VARIABLES : NODES).

Arc input(s)	`NODES`
Arc generator	CLIQUE $\mapsto$ `collection`(`nodes1`, `nodes2`)
Arc arity	2
Arc constraint(s)	• `nodes1.succ` $=$ `nodes2.index` • `nodes1.succ` $\neq$ `nodes1.index`
Graph property(ies)	**NTREE** $= 0$
Graph class	`ONE_SUCC`

Graph model Parts (A) and (B) of Figure 5.271 respectively show the initial and final graph associated with the **Example** slot. The `derangement` constraint holds since the final graph does not contain any vertex that does not belong to a circuit (i.e., **NTREE** $= 0$).



(A) **(B)**

Figure 5.271: Initial and final graph of the `derangement` constraint

In order to express the binary constraint that links two vertices of the `NODES` collection one has to make explicit the index value of the vertices. This is why the `derangement` constraint considers objects that have two attributes:

- One fixed attribute `index` that is the identifier of the vertex,
- One variable attribute `succ` that is the successor of the vertex.

Forbidding cycles of length one is achieved by the second condition of the arc constraint.

Signature Since 0 is the smallest possible value of **NTREE** we can rewrite the graph property **NTREE** $= 0$ to **NTREE** ≤ 0. This leads to simplify $\overline{\mathbf{NTREE}}$ to $\underline{\mathbf{NTREE}}$.