5.79 compare_and_count

DESCRIPTION LINKS

Origin Generalise discrepancy

Constraint compare_and_count(VARIABLES1, VARIABLES2, COMPARE, COUNT, LIMIT)

Arguments

VARIABLES1	:	collection(var−dvar)
VARIABLES2	:	collection(var−dvar)
COMPARE	:	atom
COUNT	:	atom
LIMIT	:	dvar

Restrictions

$|\texttt{VARIABLES1}| = |\texttt{VARIABLES2}|$
required(VARIABLES1, var)
required(VARIABLES2, var)
$\texttt{COMPARE} \in [=, \neq, <, \geq, >, \leq]$
$\texttt{COUNT} \in [=, \neq, <, \geq, >, \leq]$
$\texttt{LIMIT} \geq 0$

Purpose Enforce the condition $\left(\sum_{i=1}^{|\texttt{VARIABLES1}|} \texttt{VARIABLES1}[i].\texttt{var}\ \texttt{COMPARE}\ \texttt{VARIABLES2}[i].\texttt{var}\right)\ \texttt{COUNT}\ \texttt{LIMIT}$.

Example $(\langle 4,5,5,4,5\rangle, \langle 4,2,5,1,5\rangle, =, \leq, 3)$

The compare_and_count constraint holds since no more than $\texttt{LIMIT} = 3$ pairs of variables are equal, i.e., the first, third and fifth pairs.

Typical

$|\texttt{VARIABLES1}| > 1$
range(VARIABLES1.var) > 1
range(VARIABLES2.var) > 1
$\texttt{COMPARE} \in [=]$
$\texttt{COUNT} \in [=, <, \geq, >, \leq]$
$\texttt{LIMIT} > 0$
$\texttt{LIMIT} < |\texttt{VARIABLES1}|$

Arg. properties

- Contractible wrt. VARIABLES1 and VARIABLES2 (*remove items from same position*) when $\texttt{COUNT} \in [<, \leq]$.
- Extensible wrt. VARIABLES1 and VARIABLES2 (*add items at same position*) when $\texttt{COUNT} \in [\geq, >]$.

See also **common keyword:** count (*counting constraint*).

Keywords **constraint type:** predefined constraint, counting constraint.