5.336 same_and_global_cardinality

DESCRIPTION **LINKS** **GRAPH**

Origin

Conjoin `same` and `global_cardinality`

Constraint

`same_and_global_cardinality(VARIABLES1, VARIABLES2, VALUES)`

Synonyms

`sgcc`, `same_gcc`, `same_and_gcc`, `swc`, `same_with_cardinalities`.

Arguments

```
VARIABLES1  :  collection(var-dvar)
VARIABLES2  :  collection(var-dvar)
VALUES      :  collection(val-int, noccurrence-dvar)
```

Restrictions

$|\texttt{VARIABLES1}| = |\texttt{VARIABLES2}|$
$\texttt{required}(\texttt{VARIABLES1}, \texttt{var})$
$\texttt{required}(\texttt{VARIABLES2}, \texttt{var})$
$\texttt{required}(\texttt{VALUES}, [\texttt{val}, \texttt{noccurrence}])$
$\texttt{distinct}(\texttt{VALUES}, \texttt{val})$
$\texttt{VALUES.noccurrence} \geq 0$
$\texttt{VALUES.noccurrence} \leq |\texttt{VARIABLES1}|$

Purpose

The variables of the `VARIABLES2` collection correspond to the variables of the `VARIABLES1` collection according to a permutation. In addition, each value $\texttt{VALUES}[i]\texttt{.val}$ (with $i \in [1, |\texttt{VALUES}|]$) should be taken by exactly $\texttt{VALUES}[i]\texttt{.noccurrence}$ variables of the `VARIABLES1` collection. Finally, each variable of `VARIABLES1` should be assigned a value of $\texttt{VALUES}[i]\texttt{.val}$ (with $i \in [1, |\texttt{VALUES}|]$).

Example

$$\left(\begin{array}{l}\langle 1, 9, 1, 5, 2, 1\rangle,\\ \langle 9, 1, 1, 1, 2, 5\rangle,\\ \left\langle\begin{array}{ll}\texttt{val}-1 & \texttt{noccurrence}-3,\\ \texttt{val}-2 & \texttt{noccurrence}-1,\\ \texttt{val}-5 & \texttt{noccurrence}-1,\\ \texttt{val}-7 & \texttt{noccurrence}-0,\\ \texttt{val}-9 & \texttt{noccurrence}-1\end{array}\right\rangle\end{array}\right)$$

The `same_and_global_cardinality` constraint holds since:

- The values 1, 9, 1, 5, 2, 1 assigned to `VARIABLES1` correspond to a permutation of the values 9, 1, 1, 1, 2, 5 assigned to `VARIABLES2`.
- The values 1, 2, 5, 7 and 6 are respectively used 3, 1, 1, 0 and 1 times.

Typical

$|\texttt{VARIABLES1}| > 1$
$\texttt{range}(\texttt{VARIABLES1.var}) > 1$
$\texttt{range}(\texttt{VARIABLES2.var}) > 1$
$|\texttt{VALUES}| > 1$
$\texttt{range}(\texttt{VALUES.noccurrence}) > 1$
$|\texttt{VARIABLES1}| > |\texttt{VALUES}|$

Symmetries

- Arguments are permutable w.r.t. permutation (VARIABLES1, VARIABLES2) (VALUES).
- Items of VARIABLES1 are permutable.
- Items of VARIABLES2 are permutable.
- Items of VALUES are permutable.
- An occurrence of a value of VARIABLES1.var or VARIABLES2.var that does not belong to VALUES.val can be replaced by any other value that also does not belong to VALUES.val.
- All occurrences of two distinct values in VARIABLES1.var, VARIABLES2.var or VALUES.val can be swapped; all occurrences of a value in VARIABLES1.var, VARIABLES2.var or VALUES.val can be renamed to any unused value.

Arg. properties

Contractible wrt. VALUES.

Usage

See the `same_and_global_cardinality_low_up` constraint.

Algorithm

The filtering algorithm presented in [50] can be reused for pruning the variables of the VARIABLES1 and the VARIABLES2 collection. This algorithm does not restrict the noccurrence variables of the VALUES collection.

See also

implies: `global_cardinality`, `same`.

related: `k_alldifferent` *(two overlapping* `alldifferent` *plus restriction on values).*

specialisation: `same_and_global_cardinality_low_up` *(*`variable` *replaced by* `fixed interval`*).*

Keywords

application area: assignment.

combinatorial object: permutation, multiset.

constraint arguments: constraint between two collections of variables.

constraint type: value constraint.

filtering: flow.

modelling: equality between multisets.

problems: demand profile.

Arc input(s) VARIABLES1 VARIABLES2

Arc generator $PRODUCT \mapsto$ `collection`(`variables1`, `variables2`)

Arc arity 2

Arc constraint(s) `variables1.var` = `variables2.var`

Graph property(ies)
- for all connected components: **NSOURCE**=**NSINK**
- **NSOURCE**= |`VARIABLES1`|
- **NSINK**= |`VARIABLES2`|

For all items of `VALUES`:

Arc input(s) VARIABLES1

Arc generator $SELF \mapsto$ `collection`(`variables`)

Arc arity 1

Arc constraint(s) `variables.var` = `VALUES.val`

Graph property(ies) **NVERTEX**= `VALUES.noccurrence`

Graph model Parts (A) and (B) of Figure 5.687 respectively show the initial and final graph associated with the first graph constraint of the **Example** slot. Since we use the **NSOURCE** and **NSINK** graph properties, the source and sink vertices of the final graph are stressed with a double circle. Since there is a constraint on each connected component of the final graph we also show the different connected components. Each of them corresponds to an equivalence class according to the arc constraint.

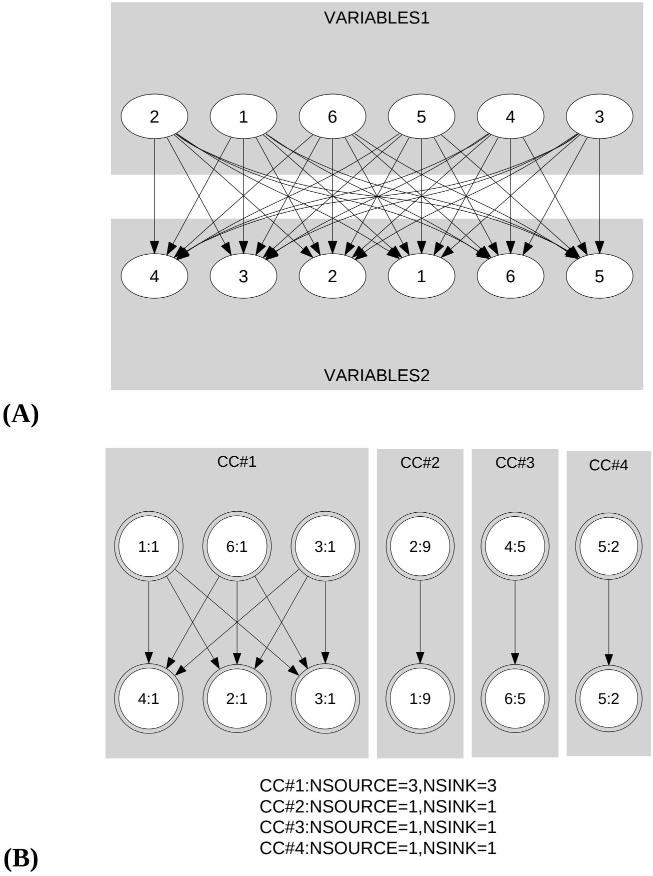

Figure 5.687: Initial and final graph of the `same_and_global_cardinality` constraint