5.15 alldifferent_cst

DESCRIPTION LINKS GRAPH

Origin CHIP

Constraint alldifferent_cst(VARIABLES)

Synonyms alldiff_cst, alldistinct_cst.

Argument VARIABLES : collection(var−dvar, cst−int)

Restriction required(VARIABLES, [var, cst])

Purpose For all pairs of items $(\texttt{VARIABLES}[i], \texttt{VARIABLES}[j])$ $(i \neq j)$ of the collection VARIABLES enforce $\texttt{VARIABLES}[i].\texttt{var} + \texttt{VARIABLES}[i].\texttt{cst} \neq \texttt{VARIABLES}[j].\texttt{var} + \texttt{VARIABLES}[j].\texttt{cst}$.

Example

$$\left(\left\langle \begin{array}{ll} \texttt{var}-5 & \texttt{cst}-0, \\ \texttt{var}-1 & \texttt{cst}-1, \\ \texttt{var}-9 & \texttt{cst}-0, \\ \texttt{var}-3 & \texttt{cst}-4 \end{array} \right\rangle\right)$$

The alldifferent_cst constraint holds since all the expressions $5 + 0 = 5$, $1 + 1 = 2$, $9 + 0 = 9$ and $3 + 4 = 7$ correspond to distinct values.

All solutions Figure 5.34 gives all solutions to the following non ground instance of the alldifferent_cst constraint: $\texttt{V}_1 \in [0, 2]$, $\texttt{V}_2 \in [4, 5]$, $\texttt{V}_3 = 4$, $\texttt{V}_4 \in [0, 1]$, $\texttt{alldifferent\_cst}(\langle\langle \texttt{V}_1, 0\rangle, \langle \texttt{V}_2, 1\rangle, \langle \texttt{V}_3, 2\rangle, \langle \texttt{V}_4, 3\rangle\rangle)$.

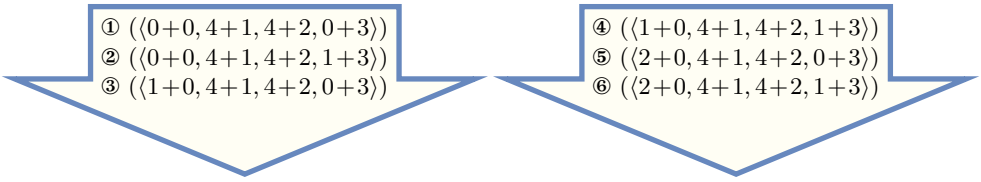

Figure 5.34: All solutions corresponding to the non ground example of the alldifferent_cst constraint of the **All solutions** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$
$2{*}\texttt{range}(\texttt{VARIABLES.var}) < 3 * |\texttt{VARIABLES}|$
$\texttt{range}(\texttt{VARIABLES.cst}) > 1$

Symmetries

- Items of `VARIABLES` are permutable.
- Attributes of `VARIABLES` are permutable w.r.t. permutation (`var`, `cst`) (*permutation not necessarily applied to all items*).
- One and the same constant can be added to the `var` attribute of all items of `VARIABLES`.
- One and the same constant can be added to the `cst` attribute of all items of `VARIABLES`.

Arg. properties

Contractible wrt. `VARIABLES`.

Usage

The `alldifferent_cst` constraint was originally introduced in **CHIP** in order to express the n-queen problem with 3 global constraints (see the **Usage** slot of the `alldifferent` constraint).

Algorithm

See the filtering algorithms of the `alldifferent` constraint.

Systems

`linear` in **Gecode**.

See also

implies (items to collection): `lex_alldifferent`.

specialisation: `alldifferent` (`variable` + `constant` *replaced by* `variable`).

Keywords

characteristic of a constraint: all different, disequality, sort based reformulation.

constraint type: value constraint.

filtering: bipartite matching, bipartite matching in convex bipartite graphs, convex bipartite graph, arc-consistency.

final graph structure: one_succ.

modelling exercises: n-Amazons.

puzzles: n-Amazons, n-queens.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var + variables1.cst = variables2.var + variables2.cst
Graph property(ies)	MAX_NSCC ≤ 1
Graph class	ONE_SUCC

Graph model We generate a *clique* with an *equality* constraint between each pair of vertices (including a vertex and itself) and state that the size of the largest strongly connected component should not exceed one.

Parts (A) and (B) of Figure 5.35 respectively show the initial and final graph associated with the **Example** slot. Since we use the MAX_NSCC graph property we show one of the largest strongly connected components of the final graph. The alldifferent_cst holds since all the strongly connected components have at most one vertex: a value is used at most once.

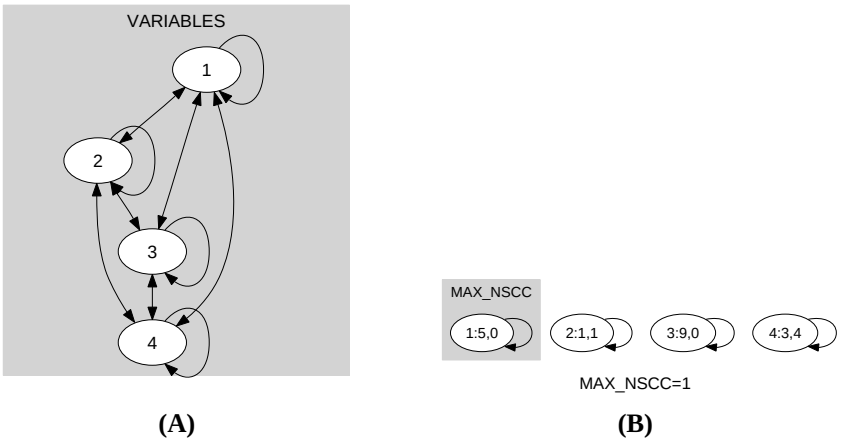

Figure 5.35: Initial and final graph of the alldifferent_cst constraint