5.63 change_pair

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from change.

Constraint change_pair(NCHANGE, PAIRS, CTRX, CTRY)

Arguments

NCHANGE	:	dvar
PAIRS	:	collection(x−dvar, y−dvar)
CTRX	:	atom
CTRY	:	atom

Restrictions

$\texttt{NCHANGE} \geq 0$
$\texttt{NCHANGE} < |\texttt{PAIRS}|$
required(PAIRS, [x, y])
$\texttt{CTRX} \in [=, \neq, <, \geq, >, \leq]$
$\texttt{CTRY} \in [=, \neq, <, \geq, >, \leq]$

Purpose NCHANGE is the number of times that the following disjunction holds: $(X_1$ CTRX $X_2) \vee (Y_1$ CTRY $Y_2)$, where (X_1, Y_1) and (X_2, Y_2) correspond to consecutive pairs of variables of the collection PAIRS.

Example

$$\left(\begin{array}{cccc} & \mathtt{x}-3 & \mathtt{y}-5, & \\ & \mathtt{x}-3 & \mathtt{y}-7, & \\ & \mathtt{x}-3 & \mathtt{y}-7, & \\ & \mathtt{x}-3 & \mathtt{y}-8, & \\ 3, \left\langle \right. & \mathtt{x}-3 & \mathtt{y}-4, & \left. \right\rangle, \neq, > \\ & \mathtt{x}-3 & \mathtt{y}-7, & \\ & \mathtt{x}-1 & \mathtt{y}-3, & \\ & \mathtt{x}-1 & \mathtt{y}-6, & \\ & \mathtt{x}-1 & \mathtt{y}-6, & \\ & \mathtt{x}-3 & \mathtt{y}-7 & \end{array}\right)$$

In the example we have the following 3 changes:

- One change between pairs $\mathtt{x}-3\ \mathtt{y}-8$ and $\mathtt{x}-3\ \mathtt{y}-4$ since $3 \neq 3 \vee 8 > 4$,
- One change between pairs $\mathtt{x}-3\ \mathtt{y}-7$ and $\mathtt{x}-1\ \mathtt{y}-3$ since $3 \neq 1 \vee 7 > 3$,
- One change between pairs $\mathtt{x}-1\ \mathtt{y}-6$ and $\mathtt{x}-3\ \mathtt{y}-7$ since $1 \neq 3 \vee 6 > 7$.

Consequently the change_pair constraint holds since its first argument NCHANGE is assigned value 3.

Typical

$\texttt{NCHANGE} > 0$
$|\texttt{PAIRS}| > 1$
$\texttt{range}(\texttt{PAIRS.x}) > 1$
$\texttt{range}(\texttt{PAIRS.y}) > 1$

Symmetries

- One and the same constant can be added to the `x` attribute of all items of `PAIRS`.
- One and the same constant can be added to the `y` attribute of all items of `PAIRS`.

Arg. properties

Functional dependency: `NCHANGE` determined by `PAIRS`, `CTRX` and `CTRY`.

Usage

Here is a typical example where this constraint is useful. Assume we have to produce a set of cables. A given quality and a given cross-section that respectively correspond to the `x` and `y` attributes of the previous pairs of variables characterise each cable. The problem is to sequence the different cables in order to minimise the number of times two consecutive wire cables C_1 and C_2 verify the following property: C_1 and C_2 do not have the same quality or the cross section of C_1 is greater than the cross section of C_2.

See also

generalisation: `change_vectors` *(*`pair` *of* `variables` *replaced by* `vector`*)*.

specialisation: `change` *(*`pair` *of* `variables` *replaced by* `variable`*)*.

Keywords

characteristic of a constraint: pair, automaton, automaton with counters.

constraint arguments: pure functional dependency.

constraint network structure: sliding cyclic(2) constraint network(2).

constraint type: timetabling constraint.

final graph structure: acyclic, bipartite, no loop.

modelling: number of changes, functional dependency.

Arc input(s)	PAIRS
Arc generator	$PATH \mapsto$ collection(pairs1, pairs2)
Arc arity	2
Arc constraint(s)	pairs1.x CTRX pairs2.x $\vee$ pairs1.y CTRY pairs2.y
Graph property(ies)	NARC$=$ NCHANGE
Graph class	• ACYCLIC • BIPARTITE • NO_LOOP

Graph model Same as change, except that each item has two attributes x and y.

Parts (A) and (B) of Figure 5.175 respectively show the initial and final graph associated with the **Example** slot. Since we use the NARC graph property, the arcs of the final graph are stressed in bold.

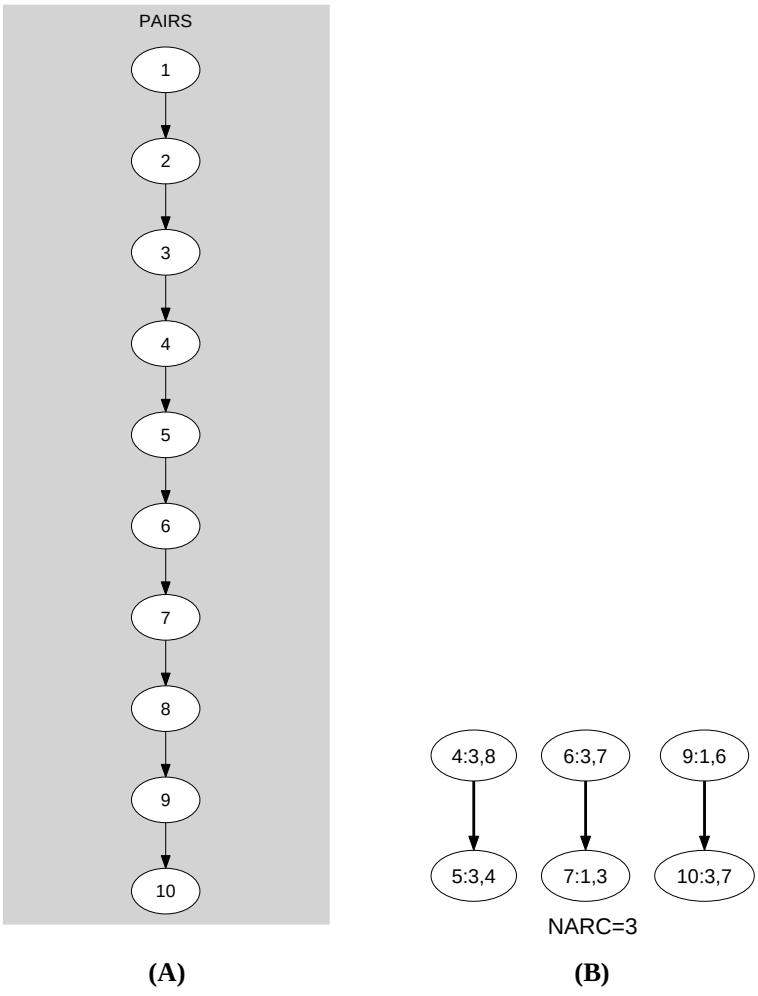

Figure 5.175: Initial and final graph of the `change_pair` constraint

Automaton

Figure 5.176 depicts the automaton associated with the change_pair constraint. To each pair of consecutive pairs $((\mathtt{X}_i, \mathtt{Y}_i), (\mathtt{X}_{i+1}, \mathtt{Y}_{i+1}))$ of the collection PAIRS corresponds a 0-1 signature variable S_i. The following signature constraint links $\mathtt{X}_i$, $\mathtt{Y}_i$, $\mathtt{X}_{i+1}$, $\mathtt{Y}_{i+1}$ and S_i: $(\mathtt{X}_i \text{ CTRX } \mathtt{X}_{i+1}) \vee (\mathtt{Y}_i \text{ CTRY } \mathtt{Y}_{i+1}) \Leftrightarrow S_i$.

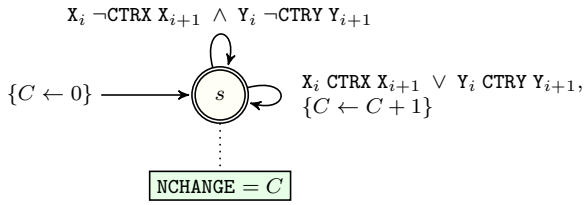

Figure 5.176: Automaton of the change_pair constraint

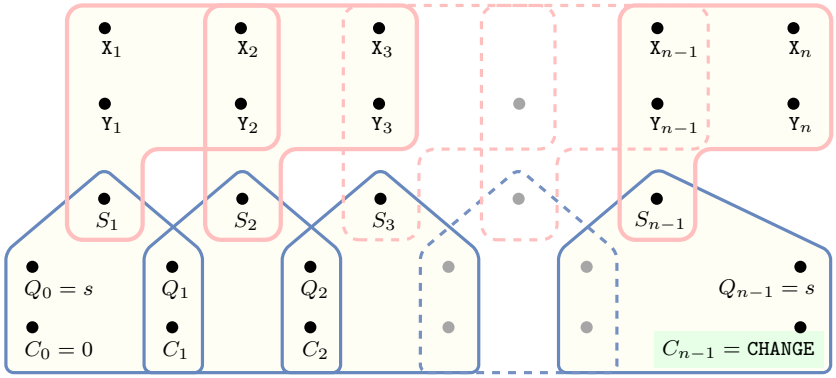

Figure 5.177: Hypergraph of the reformulation corresponding to the automaton of the change_pair constraint