

5.51 change_pair

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Derived from change .			
Constraint	<code>change_pair(NCHANGE, PAIRS, CTRX, CTRY)</code>			
Arguments	NCHANGE : <code>dvar</code> PAIRS : <code>collection(x-dvar, y-dvar)</code> CTRX : <code>atom</code> CTRY : <code>atom</code>			
Restrictions	NCHANGE ≥ 0 NCHANGE $< \text{PAIRS} $ <code>required(PAIRS, [x, y])</code> CTRX $\in [=, \neq, <, \geq, >, \leq]$ CTRY $\in [=, \neq, <, \geq, >, \leq]$			
Purpose	NCHANGE is the number of times that the following disjunction holds: $(X_1 \text{ CTRX } X_2) \vee (Y_1 \text{ CTRY } Y_2)$, where (X_1, Y_1) and (X_2, Y_2) correspond to consecutive pairs of variables of the collection PAIRS.			
Example	$3, \left\langle \begin{array}{l} x - 3 \quad y - 5, \\ x - 3 \quad y - 7, \\ x - 3 \quad y - 7, \\ x - 3 \quad y - 8, \\ x - 3 \quad y - 4, \\ x - 3 \quad y - 7, \\ x - 1 \quad y - 3, \\ x - 1 \quad y - 6, \\ x - 1 \quad y - 6, \\ x - 3 \quad y - 7 \end{array} \right\rangle, \neq, >$			
	In the example we have the following 3 changes: <ul style="list-style-type: none"> • One change between pairs $x - 3 \quad y - 8$ and $x - 3 \quad y - 4$ since $3 \neq 3 \vee 8 > 4$, • One change between pairs $x - 3 \quad y - 7$ and $x - 1 \quad y - 3$ since $3 \neq 1 \vee 7 > 3$, • One change between pairs $x - 1 \quad y - 6$ and $x - 3 \quad y - 7$ since $1 \neq 3 \vee 6 > 7$. Consequently the <code>change_pair</code> constraint holds since its first argument NCHANGE is assigned to 3.			
Typical	NCHANGE > 0 $ \text{PAIRS} > 1$ <code>range(PAIRS.x) > 1</code> <code>range(PAIRS.y) > 1</code>			

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.

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

[specialisation: change](#) (pair of variables replaced by variable).

Keywords

[characteristic of a constraint: pair, automaton, automaton with counters.](#)

[constraint network structure: sliding cyclic\(2\) constraint network\(2\).](#)

[constraint type: timetabling constraint.](#)

[final graph structure: acyclic, bipartite, no loop.](#)

[modelling: number of changes.](#)

Arc input(s)	PAIRS
Arc generator	<i>PATH</i> \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	<ul style="list-style-type: none"> • 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 5.112 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.

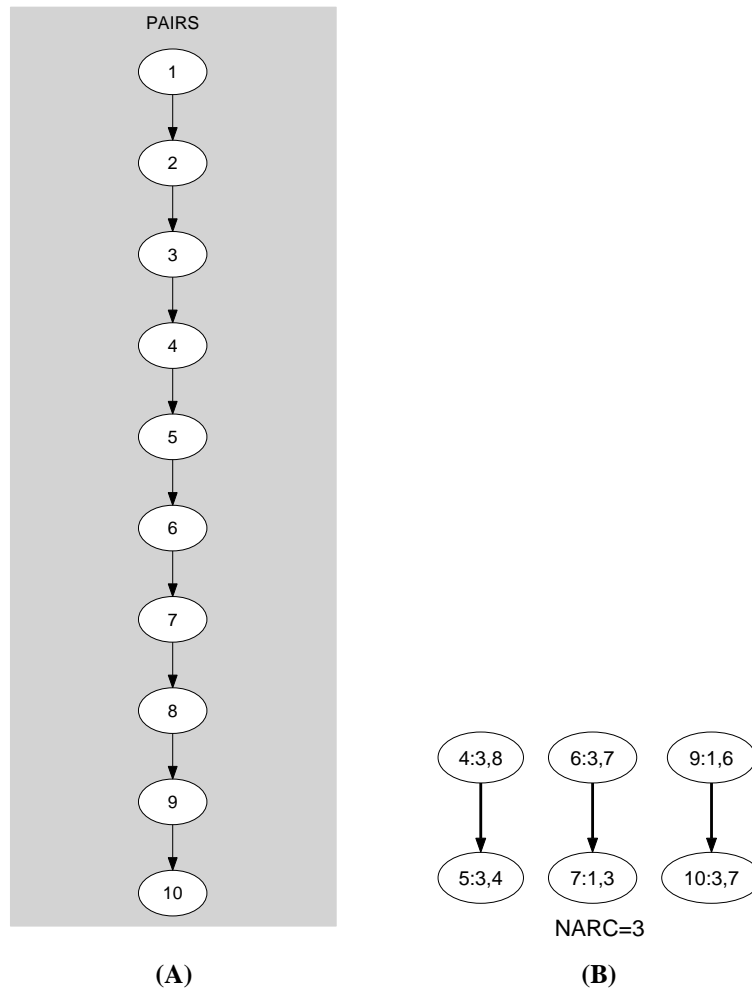


Figure 5.112: Initial and final graph of the change_pair constraint

Automaton

Figure 5.113 depicts the automaton associated with the `change_pair` constraint. To each pair of consecutive pairs $((X_i, Y_i), (X_{i+1}, Y_{i+1}))$ of the collection `PAIRS` corresponds a 0-1 signature variable S_i . The following signature constraint links $X_i, Y_i, X_{i+1}, Y_{i+1}$ and S_i : $(X_i \text{ CTRX } X_{i+1}) \vee (Y_i \text{ CTRY } Y_{i+1}) \Leftrightarrow S_i$.

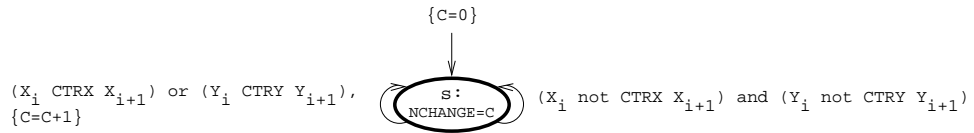


Figure 5.113: Automaton of the `change_pair` constraint

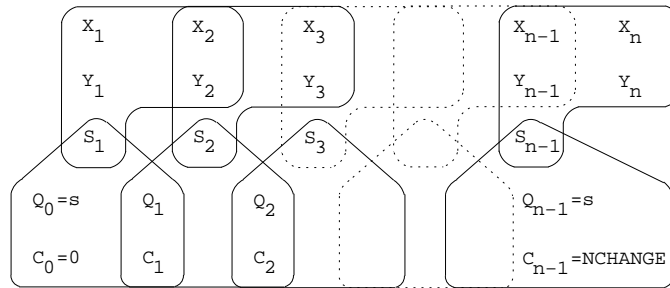


Figure 5.114: Hypergraph of the reformulation corresponding to the automaton of the `change_pair` constraint

20030820

595