

5.55 circular_change

| | DESCRIPTION | LINKS | GRAPH | AUTOMATON |
|---------------------|--|-------|-------|-----------|
| Origin | Derived from change . | | | |
| Constraint | <code>circular_change(NCHANGE, VARIABLES, CTR)</code> | | | |
| Arguments | NCHANGE : dvar VARIABLES : collection (var-dvar) CTR : atom | | | |
| Restrictions | NCHANGE ≥ 0 NCHANGE $\leq VARIABLES $ required (VARIABLES, var) CTR $\in [=, \neq, <, \geq, >, \leq]$ | | | |
| Purpose | NCHANGE is the number of times that CTR holds on consecutive variables of the collection VARIABLES. The last and the first variables of the collection VARIABLES are also considered to be consecutive. | | | |
| Example | <div style="border: 1px solid black; padding: 5px; display: inline-block;"> $(4, \langle 4, 4, 3, 4, 1 \rangle, \neq)$ </div> <p>In the example the changes within the <code>VARIABLES = $\langle 4, 4, 3, 4, 1 \rangle$</code> collection are located between values 4 and 3, 3 and 4, 4 and 1, and 1 and 4 (i.e., since the third argument CTR of the <code>circular_change</code> constraint is set to <code>\neq</code>, we count one change for each disequality constraint between two consecutive variables that holds). Consequently, the corresponding <code>circular_change</code> constraint holds since its first argument NCHANGE is fixed to 4.</p> | | | |
| Typical | NCHANGE > 0 $ VARIABLES > 1$ range (VARIABLES.var) > 1 | | | |
| Symmetries | <ul style="list-style-type: none"> • Items of VARIABLES can be shifted. • One and the same constant can be added to the var attribute of all items of VARIABLES. | | | |
| See also | common keyword : change (<i>number of changes</i>). | | | |
| Keywords | characteristic of a constraint : cyclic, automaton, automaton with counters. constraint network structure : circular sliding cyclic(1) constraint network(2). constraint type : timetabling constraint. modelling : number of changes. | | | |

| | |
|----------------------------|--|
| Arc input(s) | VARIABLES |
| Arc generator | <i>CIRCUIT</i> \mapsto <code>collection(variables1, variables2)</code> |
| Arc arity | 2 |
| Arc constraint(s) | variables1.var CTR variables2.var |
| Graph property(ies) | NARC = NCHANGE |

Graph model

Since we are also interested in the constraint that links the last and the first variable we use the arc generator *CIRCUIT* to produce the arcs of the initial graph.

Parts (A) and (B) of Figure 5.121 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.

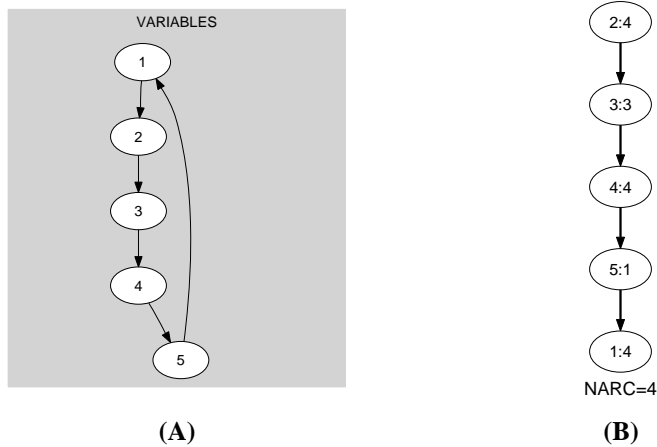


Figure 5.121: Initial and final graph of the circular_change constraint

Automaton

Figure 5.122 depicts the automaton associated with the `circular_change` constraint. To each pair of consecutive variables $(VAR_i, VAR_{(i \bmod |VARIABLES|)+1})$ of the collection `VARIABLES` corresponds a 0-1 signature variable S_i . The following signature constraint links VAR_i , $VAR_{(i \bmod |VARIABLES|)+1}$ and S_i : $VAR_i \text{ CTR } VAR_{(i \bmod |VARIABLES|)+1} \Leftrightarrow S_i$.

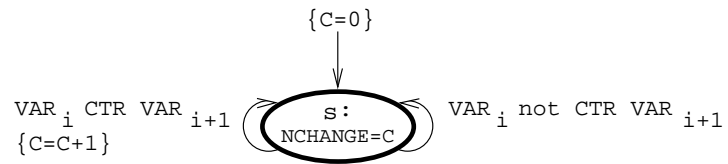


Figure 5.122: Automaton of the `circular_change` constraint

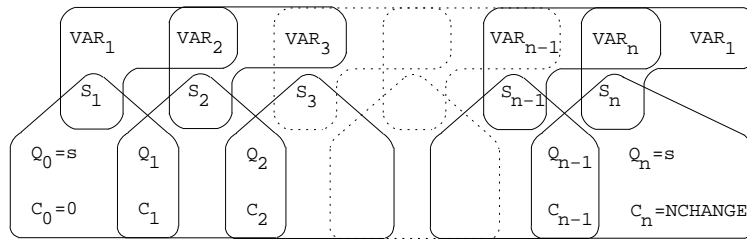


Figure 5.123: Hypergraph of the reformulation corresponding to the automaton of the `circular_change` constraint

20030820

613