

5.38 balance_modulo

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Derived from balance .			
Constraint	<code>balance_modulo(BALANCE, VARIABLES, M)</code>			
Arguments	BALANCE : dvar VARIABLES : collection(var-dvar) M : int			
Restrictions	$BALANCE \geq 0$ $BALANCE \leq \max(0, VARIABLES - 2)$ required (VARIABLES, var) $M > 0$			
Purpose	<div style="border: 1px solid pink; padding: 5px;"> Consider the largest set \mathcal{S}_1 (respectively the smallest set \mathcal{S}_2) of variables of the collection VARIABLES that have the same remainder when divided by M. BALANCE is equal to the difference between the cardinality of \mathcal{S}_2 and the cardinality of \mathcal{S}_1. </div>			
Example	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> $(2, \langle 6, 1, 7, 1, 5 \rangle, 3)$ </div> <p>In this example values 6, 1, 7, 1, 5 are respectively associated with the equivalence classes $6 \bmod 3 = 0$, $1 \bmod 3 = 1$, $7 \bmod 3 = 1$, $1 \bmod 3 = 1$, $5 \bmod 3 = 2$. Therefore the equivalence classes 0, 1 and 2 are respectively used 1, 3 and 1 times. The <code>balance_modulo</code> constraint holds since its first argument BALANCE is assigned to the difference between the maximum and minimum number of the previous occurrences (i.e., $3 - 1$).</p>			
Typical	$ VARIABLES > 2$ $M > 1$ $M < \text{maxval}(VARIABLES.var)$			
Symmetries	<ul style="list-style-type: none"> Items of VARIABLES are permutable. An occurrence of a value u of VARIABLES.var can be replaced by any other value v such that v is congruent to u modulo M. 			
Usage	An application of the <code>balance_modulo</code> constraint is to enforce a <i>balanced assignment</i> of values, no matter how many distinct equivalence classes will be used. In this case one will <i>push down</i> the maximum value of the first argument of the <code>balance_modulo</code> constraint.			
See also	specialisation: balance (variable mod constant replaced by variable).			
Keywords	application area: assignment . characteristic of a constraint: modulo, automaton, automaton with array of counters .			

constraint type: value constraint.

final graph structure: equivalence.

modelling: balanced assignment.

Arc input(s)	VARIABLES
Arc generator	<i>CLIQUE</i> \mapsto collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var mod M = variables2.var mod M
Graph property(ies)	RANGE_NSCC = BALANCE
Graph class	EQUIVALENCE

Graph model

The graph property **RANGE_NSCC** constrains the difference between the sizes of the largest and smallest strongly connected components.

Parts (A) and (B) of Figure 5.70 respectively show the initial and final graph associated with the **Example** slot. Since we use the **RANGE_NSCC** graph property, we show the largest and smallest strongly connected components of the final graph.

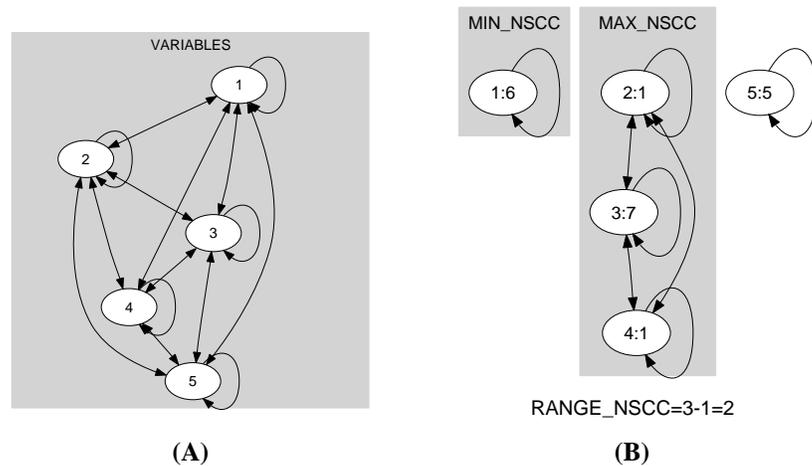


Figure 5.70: Initial and final graph of the balance_modulo constraint

Automaton

Figure 5.71 depicts the automaton associated with the `balance_modulo` constraint. To each item of the collection `VARIABLES` corresponds a signature variable S_i that is equal to 1.

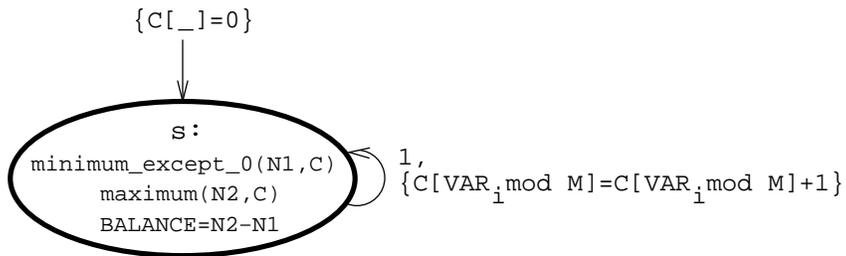


Figure 5.71: Automaton of the `balance_modulo` constraint