

5.121 discrepancy

	DESCRIPTION	LINKS	GRAPH
Origin	[170] and [423]		
Constraint	<code>discrepancy(VARIABLES, K)</code>		
Arguments	<pre>VARIABLES : collection(var-dvar, bad-sint) K : int</pre>		
Restrictions	<pre>required(VARIABLES, var) required(VARIABLES, bad) K ≥ 0 K ≤ VARIABLES </pre>		
Purpose	K is the number of variables of the collection VARIABLES that take their value in their respective sets of bad values.		
Example	$\left(\left\langle \begin{array}{ll} \text{var} - 4 & \text{bad} - \{1, 4, 6\}, \\ \text{var} - 5 & \text{bad} - \{0, 1\}, \\ \text{var} - 5 & \text{bad} - \{1, 6, 9\}, \\ \text{var} - 4 & \text{bad} - \{1, 4\}, \\ \text{var} - 1 & \text{bad} - \emptyset \end{array} \right\rangle, 2 \right)$ <p>The discrepancy constraint holds since exactly $K = 2$ variables (i.e., the first and fourth variables) of the VARIABLES collection take their value within their respective sets of bad values.</p>		
Typical	<pre> VARIABLES > 1 K < VARIABLES </pre>		
Symmetries	<ul style="list-style-type: none"> Items of VARIABLES are permutable. All occurrences of two distinct values in VARIABLES.var or VARIABLES.bad can be swapped; all occurrences of a value in VARIABLES.var or VARIABLES.bad can be renamed to any unused value. 		
Arg. properties	<ul style="list-style-type: none"> Functional dependency: K determined by VARIABLES. Aggregate: VARIABLES(union), K(+). 		
Remark	<p>Limited discrepancy search was first introduced by M. L. Ginsberg and W. D. Harvey as a search technique in [193]. Later on, discrepancy based filtering was presented in the PhD thesis of F. Focacci [170, pages 171–172]. Finally the discrepancy constraint was explicitly defined in the PhD thesis of W.-J. van Hoeve [423, page 104].</p>		
See also	<p>common keyword: among (<i>counting constraint</i>).</p> <p>used in graph description: in_set.</p>		

Keywords

constraint arguments: pure functional dependency.

constraint type: value constraint, counting constraint.

filtering: arc-consistency.

heuristics: heuristics, limited discrepancy search.

modelling: functional dependency.

Arc input(s)	VARIABLES
Arc generator	<i>SELF</i> \mapsto collection(variables)
Arc arity	1
Arc constraint(s)	in_set(variables.var, variables.bad)
Graph property(ies)	NARC = K

Graph model The arc constraint corresponds to the constraint `in_set(variables.var, variables.bad)` defined in this catalogue. We employ the *SELF* arc generator in order to produce an initial graph with a single loop on each vertex.

Parts (A) and (B) of Figure 5.272 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the loops of the final graph are stressed in bold.

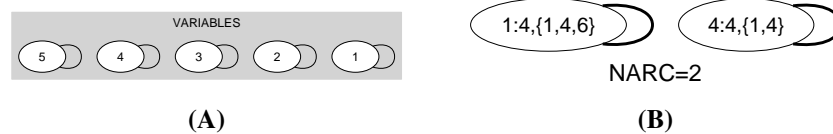


Figure 5.272: Initial and final graph of the discrepancy constraint

20050506

1065