

5.91 count

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	[99]			
Constraint	count(VALUE, VARIABLES, RELOP, LIMIT)			
Synonyms	occurencemax, occurencemin, occurrence.			
Arguments	VALUE : int VARIABLES : collection(var-dvar) RELOP : atom LIMIT : dvar			
Restrictions	required(VARIABLES, var) RELOP ∈ [=, ≠, <, ≥, >, ≤]			
Purpose	Let N be the number of variables of the VARIABLES collection assigned to value VALUE; Enforce condition N RELOP LIMIT to hold.			
Example	$(5, \langle 4, 5, 5, 4, 5 \rangle, \geq, 2)$			
	The count constraint holds since value VALUE = 5 occurs 3 times within the items of the collection VARIABLES = $\langle 4, 5, 5, 4, 5 \rangle$, which is greater than or equal to (RELOP is set to \geq) LIMIT = 2.			
Typical	$ \text{VARIABLES} > 1$ $\text{range}(\text{VARIABLES.var}) > 1$ RELOP ∈ [=, <, ≥, >, ≤] LIMIT > 0 LIMIT < VARIABLES			
Symmetries	<ul style="list-style-type: none"> Items of VARIABLES are permutable. An occurrence of a value of VARIABLES.var that is different from VALUE can be replaced by any other value that is also different from VALUE. 			
Arg. properties	<ul style="list-style-type: none"> Contractible wrt. VARIABLES when RELOP ∈ [<, ≤]. Extensible wrt. VARIABLES when RELOP ∈ [≥, >]. Aggregate: VALUE(id), VARIABLES(union), RELOP(id), LIMIT(+) when RELOP ∈ [<, ≤, ≥, >]. 			
Remark	Similar to the among constraint. Both, in JaCoP (http://www.jacop.eu/) and in MiniZinc (http://www.minizinc.org/) RELOP is implicitly set to =.			

Reformulation	The <code>count(VALUE, VARIABLES, RELOP, LIMIT)</code> constraint can be expressed in term of the conjunction <code>among(N, VARIABLES, (VALUE)) \wedge N RELOP LIMIT</code> .
Systems	<code>occurence</code> in Choco , <code>count</code> in Gecode , <code>count</code> in JaCoP , <code>count</code> in MiniZinc , <code>count</code> in SICStus .
See also	<p>assignment dimension added: <code>assign_and_counts</code> (<code>variable=VALUE</code> replaced by <code>variable \in VALUES</code> and <i>assignment dimension introduced</i>).</p> <p>common keyword: <code>among</code> (<i>value constraint, counting constraint</i>), <code>arith</code> (<i>value constraint</i>), <code>compare_and_count</code> (<i>counting constraint</i>), <code>global_cardinality</code>, <code>max_nvalue</code>, <code>min_nvalue</code> (<i>value constraint, counting constraint</i>), <code>nvalue</code> (<i>counting constraint</i>).</p> <p>generalisation: <code>counts</code> (<code>variable=VALUE</code> replaced by <code>variable \in VALUES</code>).</p> <p>related: <code>roots</code>.</p> <p>used in reformulation: <code>among</code>.</p>
Keywords	<p>characteristic of a constraint: <code>automaton</code>, <code>automaton with counters</code>.</p> <p>constraint network structure: <code>alpha-acyclic constraint network(2)</code>.</p> <p>constraint type: <code>value constraint</code>, <code>counting constraint</code>.</p> <p>filtering: <code>arc-consistency</code>.</p>

Arc input(s)	VARIABLES
Arc generator	<i>SELF</i> \mapsto collection(variables)
Arc arity	1
Arc constraint(s)	variables.var = VALUE
Graph property(ies)	NARC RELOP LIMIT

Graph model

Parts (A) and (B) of Figure 5.203 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.

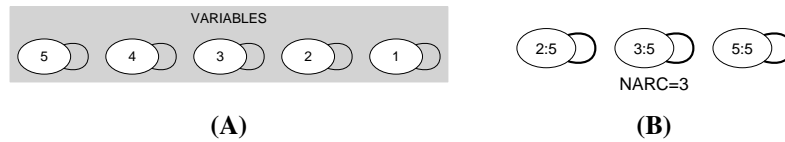


Figure 5.203: Initial and final graph of the count constraint

Automaton

Figure 5.204 depicts the automaton associated with the count constraint. To each variable VAR_i of the collection VARIABLES corresponds a 0-1 signature variable S_i . The following signature constraint links VAR_i and S_i : $VAR_i = VALUE \Leftrightarrow S_i$.

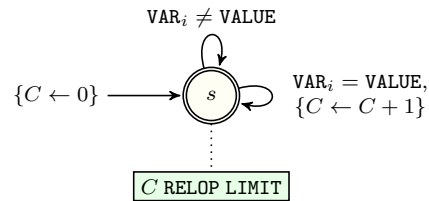


Figure 5.204: Automaton of the count constraint

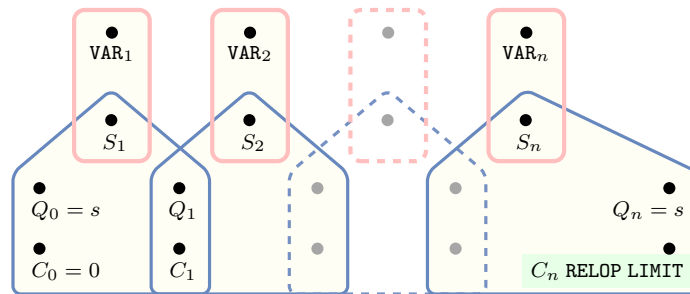


Figure 5.205: Hypergraph of the reformulation corresponding to the automaton (with one counter) of the count constraint: since all states variables Q_0, Q_1, \dots, Q_n are fixed to the unique state s of the automaton, the transitions constraints share only the counter variable C and the constraint network is Berge-acyclic