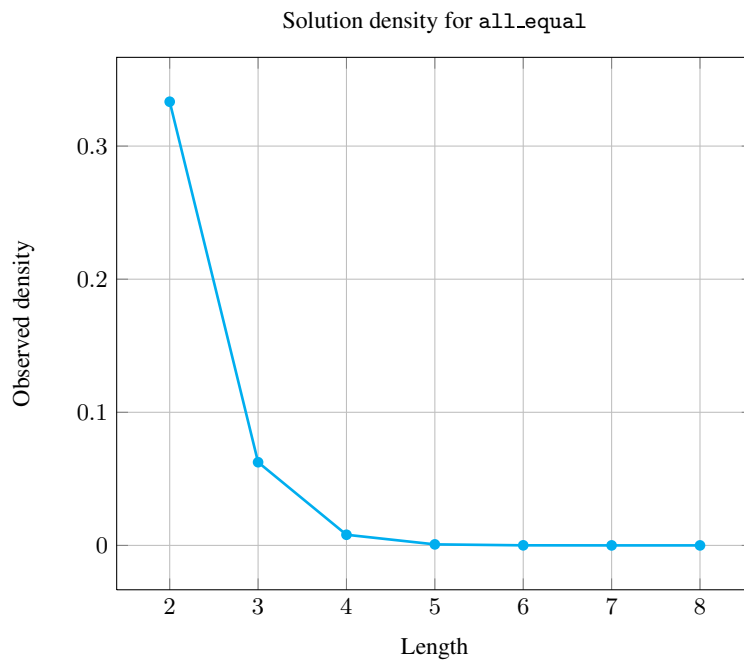
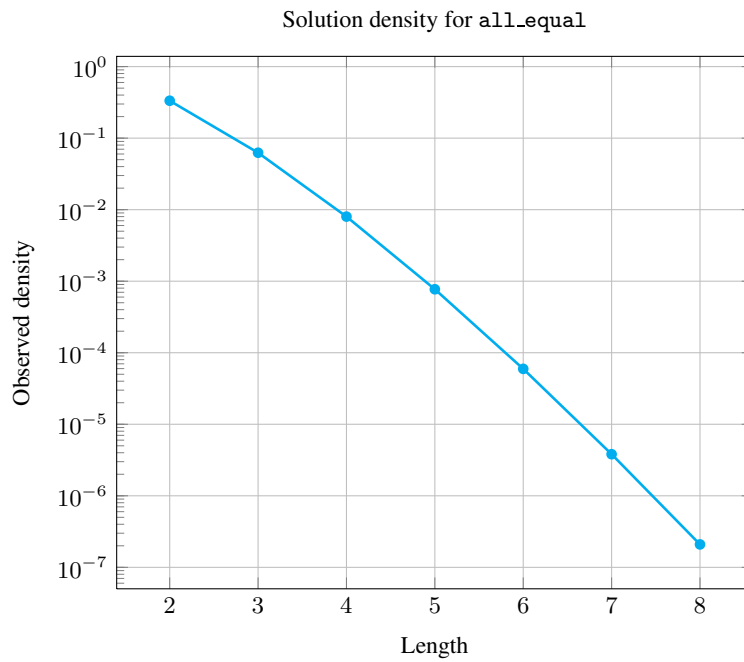


5.5 all_equal

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>soft_all_equal_min_ctr</code>		
Constraint	<code>all_equal(VARIABLES)</code>		
Synonym	<code>rel.</code>		
Argument	<code>VARIABLES : collection(var-dvar)</code>		
Restrictions	<code>required(VARIABLES, var)</code> <code> VARIABLES > 0</code>		
Purpose	Enforce all variables of the collection <code>VARIABLES</code> to take the same value.		
Example	<code>((5, 5, 5, 5))</code>		
	The <code>all_equal</code> constraint holds since all its variables are fixed to value 5.		
All solutions	Figure 5.5 gives all solutions to the following non ground instance of the <code>all_equal</code> constraint: $V_1 \in [0, 6]$, $V_2 \in [0, 2]$, $V_3 \in [0, 2]$, $V_4 \in [1, 4]$, <code>all_equal</code> ((V_1, V_2, V_3, V_4)).		
			
	Figure 5.5: All solutions corresponding to the non ground example of the <code>all_equal</code> constraint of the All solutions slot		
Typical	<code> VARIABLES > 2</code> <code>minval(VARIABLES.var) ≠ 0</code>		
Symmetries	<ul style="list-style-type: none"> Items of <code>VARIABLES</code> are <code>permutable</code>. All occurrences of a value of <code>VARIABLES.var</code> can be <code>renamed</code> to any unused value. 		
Arg. properties	<code>Contractible</code> wrt. <code>VARIABLES</code> .		
Counting			

Length (n)	2	3	4	5	6	7	8
Solutions	3	4	5	6	7	8	9

Number of solutions for `all_equal`: domains $0..n$



Systems [atMostNValue](#) in [Choco](#), [rel](#) in [Gecode](#), [all_equal](#) in [MiniZinc](#).

See also [generalisation: nvalue](#) (a variable counting the number of distinct values is introduced).

implies: `consecutive_values`, `decreasing`, `increasing`,
`multi_global_contiguity`.

negation: `not_all_equal`.

soft variant: `soft_all_equal_max_var`,
`soft_all_equal_min_ctr` (*decomposition-based violation measure*),
`soft_all_equal_min_var` (*variable-based violation measure*).

specialisation: `eq` (*equality between just two variables*).

Keywords

constraint type: value constraint.

Cond. implications

`all_equal(VARIABLES)`
 with `|VARIABLES| > 1`
implies `some_equal(VARIABLES)`.

Arc input(s)	VARIABLES
Arc generator	<i>PATH</i> \mapsto <i>collection</i> (variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	NARC = VARIABLES - 1

Graph model

We use the arc generator *PATH* in order to link consecutive variables of the collection VARIABLES by a binary equality constraint.

Parts (A) and (B) of Figure 5.6 respectively show the initial and final graph of the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold.

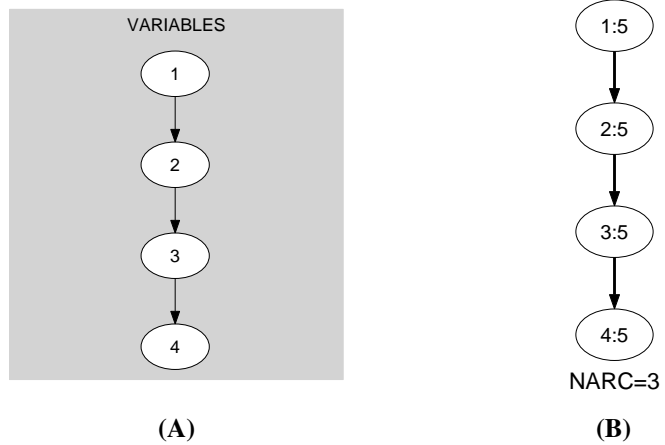


Figure 5.6: Initial and final graph of the `all_equal` constraint