

5.175 highest_peak

	DESCRIPTION	LINKS	AUTOMATON
Origin	Derived from <code>peak</code> .		
Constraint	<code>highest_peak(HEIGHT, VARIABLES)</code>		
Arguments	<div>HEIGHT : <code>dvar</code> VARIABLES : <code>collection(var-dvar)</code></div>		
Restriction	<code>required(VARIABLES, var)</code>		
Purpose	A variable V_k ($1 < k < m$) of the sequence of variables $VARIABLES = V_1, \dots, V_m$ is a <i>peak</i> if and only if there exists an i ($1 < i \leq k$) such that $V_{i-1} < V_i$ and $V_i = V_{i+1} = \dots = V_k$ and $V_k > V_{k+1}$. HEIGHT is the maximum value of the peak variables. If no such variable exists HEIGHT is equal to MININT.		
Example	<div><div>(8, (1, 1, 4, 8, 6, 2, 7, 1))</div><div>(1, (0, 1, 1, 0, 0, 1, 0, 1))</div></div>		

The first `highest_peak` constraint holds since 8 is the maximum peak of the sequence 1 1 4 8 6 2 7 1.

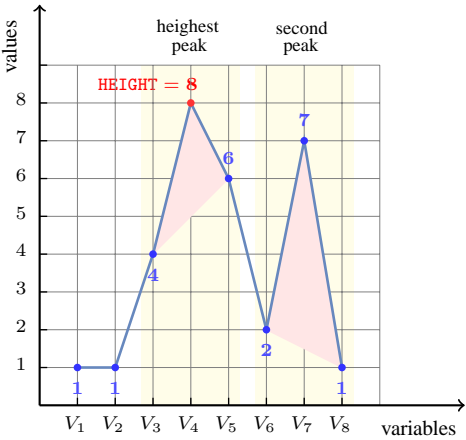


Figure 5.410: Illustration of the first constraint of the **Example** slot: a sequence of eight variables $V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8$ respectively fixed to values 1, 1, 4, 8, 6, 2, 7, 1 and its corresponding highest peak 8

Typical	<div><code> VARIABLES > 2</code> <code>range(VARIABLES.var) > 2</code> <code>peak(VARIABLES.var) > 0</code></div>
---------	---

Symmetry

Items of VARIABLES can be [reversed](#).

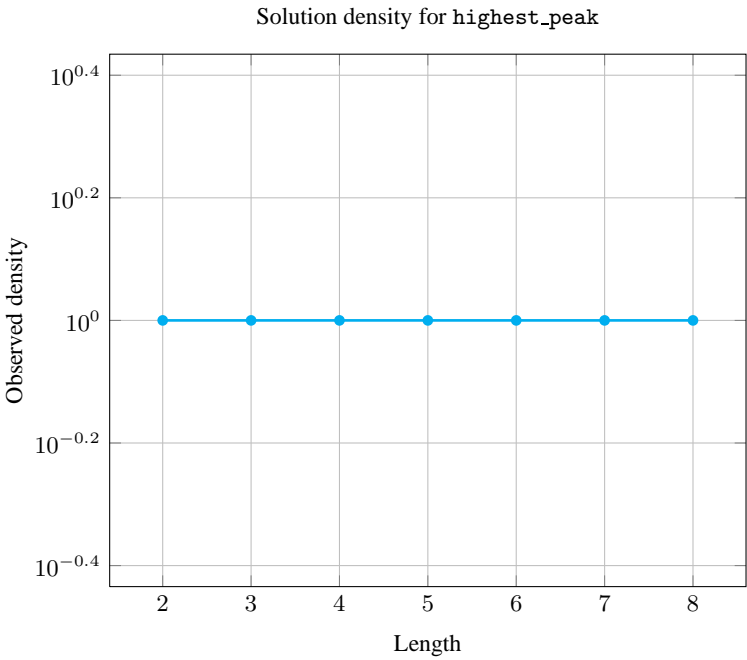
Arg. properties

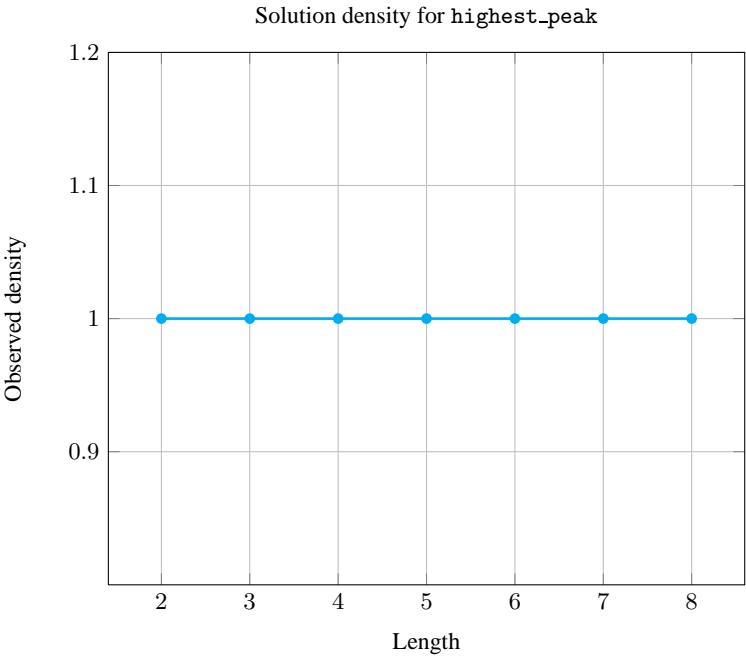
[Functional dependency](#): HEIGHT determined by VARIABLES.

Counting

Length (<i>n</i>)	2	3	4	5	6	7	8
Solutions	9	64	625	7776	117649	2097152	43046721

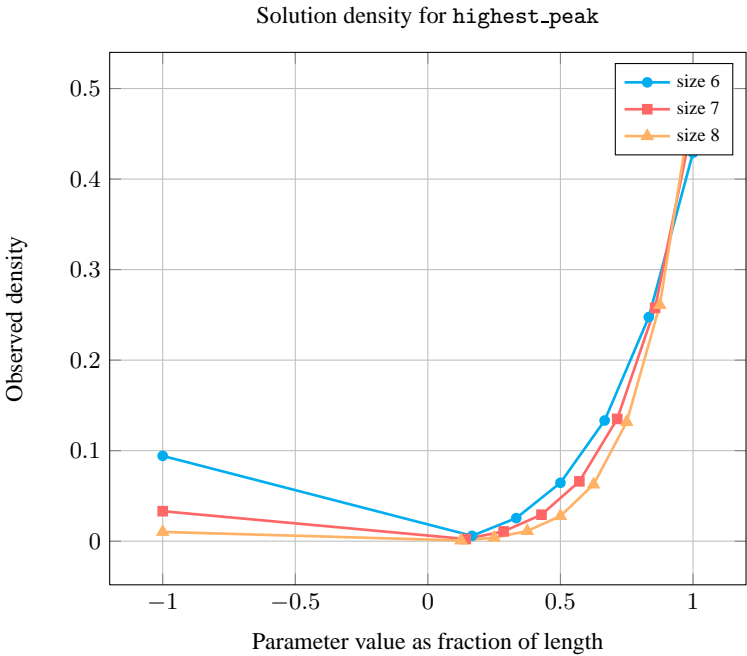
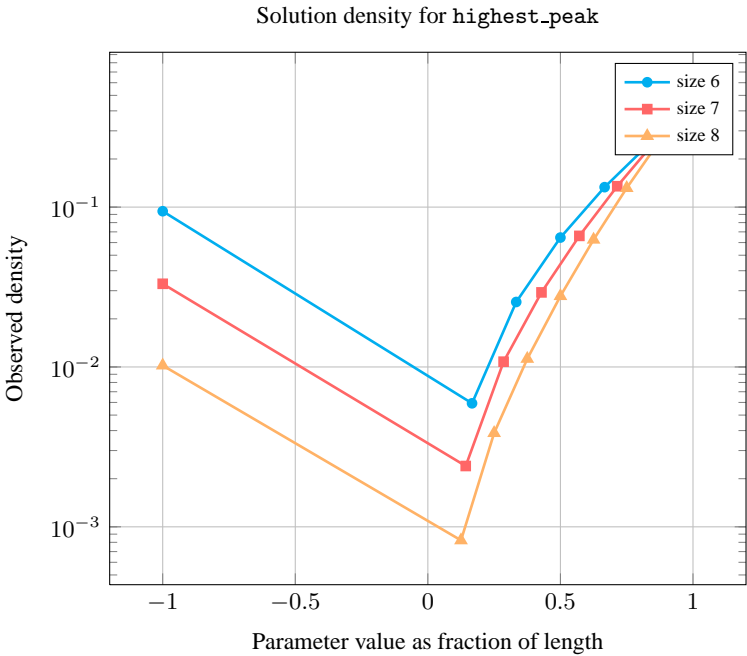
Number of solutions for highest_peak: domains 0..*n*





Length (<i>n</i>)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	-1000000	9	50	295	1792	11088	69498	439791
	1	-	1	11	92	697	5036	35443
	2	-	4	44	380	3000	22632	166208
	3	-	9	99	900	7587	61389	484020
	4	-	-	176	1712	15680	138544	1195056
	5	-	-	-	2900	29125	283250	2693425
	6	-	-	-	-	50472	540576	5665896
	7	-	-	-	-	-	976227	11233250
	8	-	-	-	-	-	-	21133632

Solution count for highest_peak: domains 0..*n*



See also [common keyword: deepest_valley, peak\(sequence\).](#)
[implies: between_min_max.](#)

Keywords

characteristic of a constraint: automaton, automaton with counters, automaton with same input symbol.

combinatorial object: sequence.

constraint arguments: reverse of a constraint, pure functional dependency.

constraint network structure: sliding cyclic(1) constraint network(2).

filtering: glue matrix.

modelling: functional dependency.

Automaton

Figure 5.411 depicts the automaton associated with the `highest_peak` constraint. To each pair of consecutive variables (VAR_i, VAR_{i+1}) of the collection `VARIABLES` corresponds a signature variable S_i . The following signature constraint links VAR_i , VAR_{i+1} and S_i :

$$VAR_i < VAR_{i+1} \Leftrightarrow S_i = 0 \wedge VAR_i = VAR_{i+1} \Leftrightarrow S_i = 1 \wedge VAR_i > VAR_{i+1} \Leftrightarrow S_i = 2.$$

STATES SEMANTICS

s : stationary/decreasing mode ($\{> | =\}^*$)
 u : increasing mode ($\{< | =\}^*$)

Blue matrix where \vec{C} and \overleftarrow{C} resp. represent the counters values C at the end of a prefix and at the end of the corresponding reverse suffix that partitions the sequence `VARIABLES`; \vec{X} denotes the last variable of the prefix.

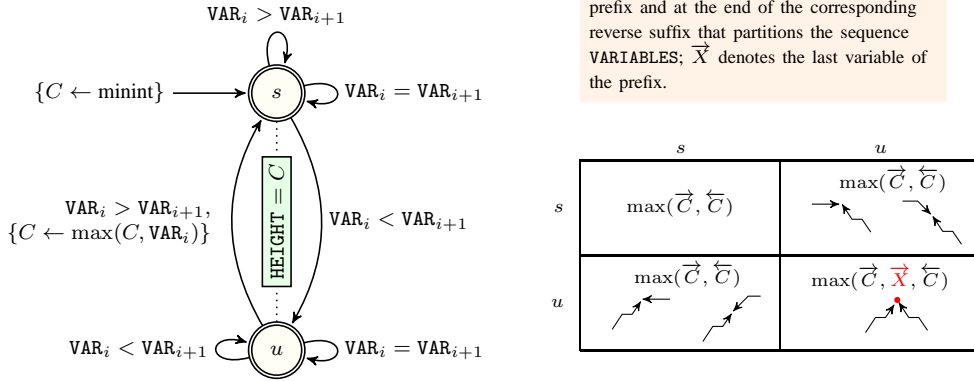


Figure 5.411: Automaton of the `highest_peak` constraint and its glue matrix (state s means that we are in *decreasing* or *stationary* mode, state u means that we are in *increasing* mode, a new peak is detected each time we switch from increasing to decreasing mode and the counter C is updated accordingly); `minint` is the smallest integer that can be represented on a machine

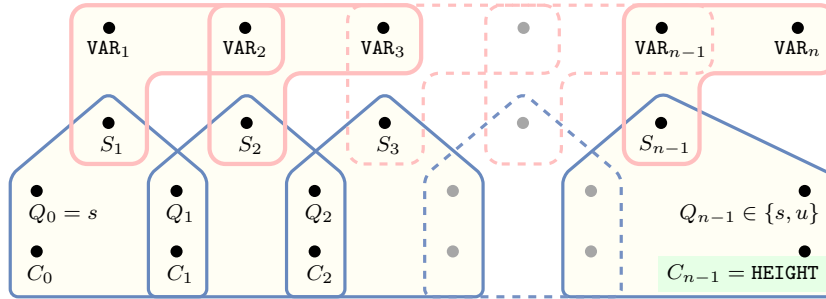


Figure 5.412: Hypergraph of the reformulation corresponding to the automaton of the `highest_peak` constraint (C_0 is set to `minint` the largest integer that can be represented on a machine)