5.175 highest_peak

DESCRIPTION LINKS AUTOMATON

Origin Derived from peak.

Constraint highest_peak(HEIGHT, VARIABLES)

Arguments

HEIGHT	:	dvar
VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose A variable V_k $(1 < k < m)$ of the sequence of variables VARIABLES $= V_1, \ldots, V_m$ is a *peak* if and only if there exists an i $(1 < i \leq k)$ such that $V_{i-1} < V_i$ and $V_i = V_{i+1} = \cdots = V_k$ 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

$(8, \langle 1, 1, 4, 8, 6, 2, 7, 1\rangle)$
$(1, \langle 0, 1, 1, 0, 0, 1, 0, 1\rangle)$

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

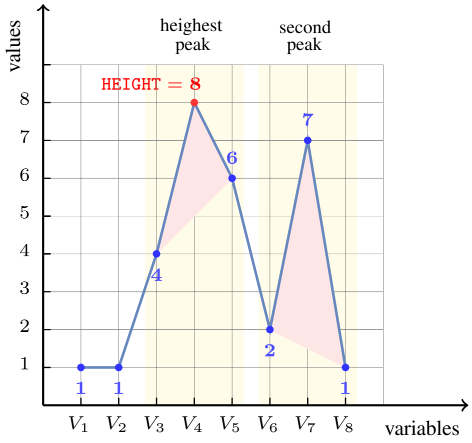


Figure 5.393: Illustration of the first constraint of the **Example** slot: a sequence of eight variables V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8 respectively fixed to values 1, 1, 4, 8, 6, 2, 7, 1 and its corresponding highest peak 8

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 2
peak(VARIABLES.var) > 0

Symmetry

Items of `VARIABLES` can be reversed.

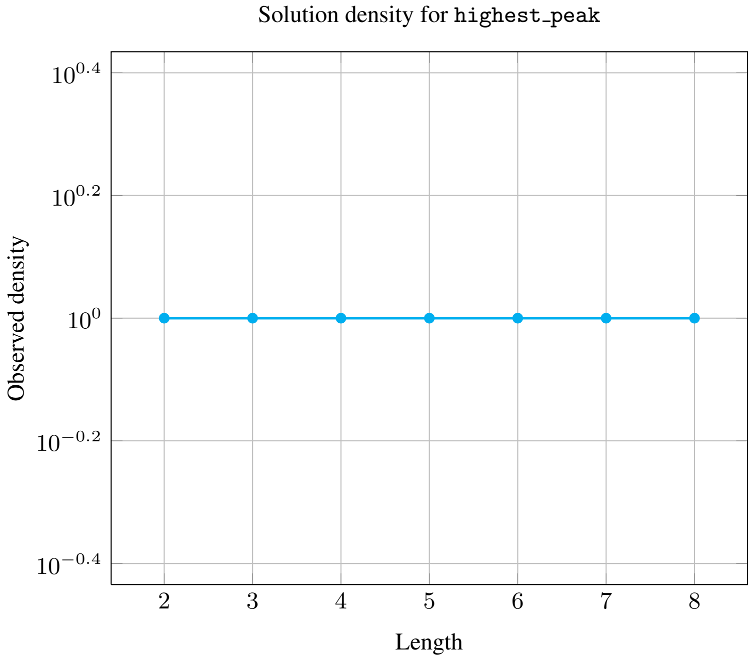
Arg. properties

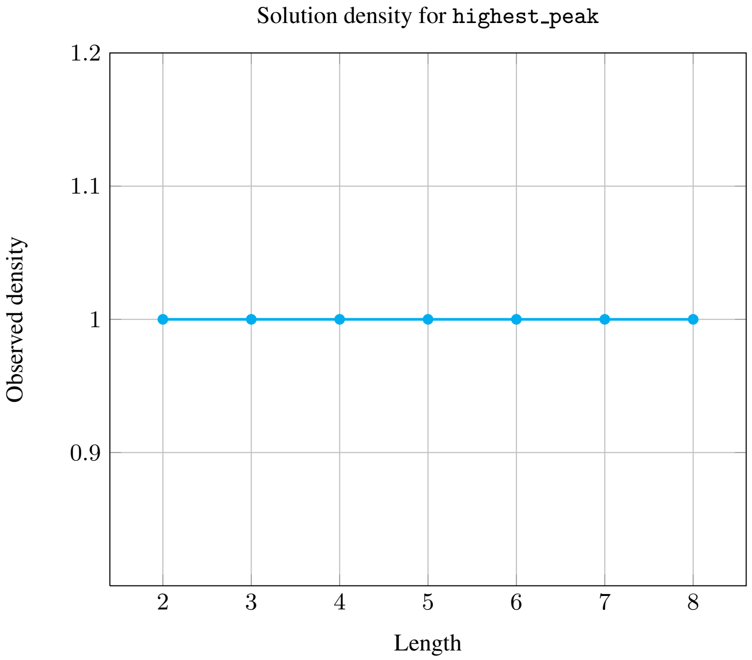
Functional dependency: `HEIGHT` determined by `VARIABLES`.

Counting

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

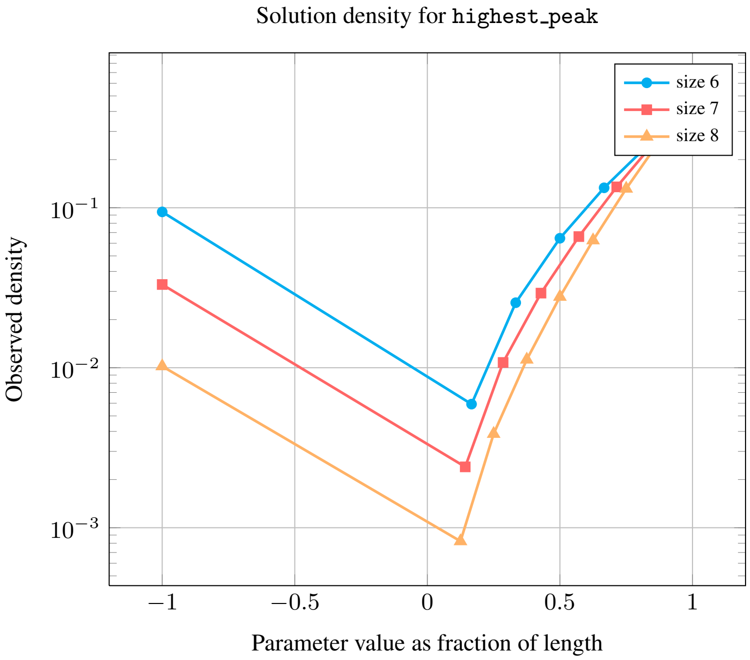
Number of solutions for `highest_peak`: domains $0..n$

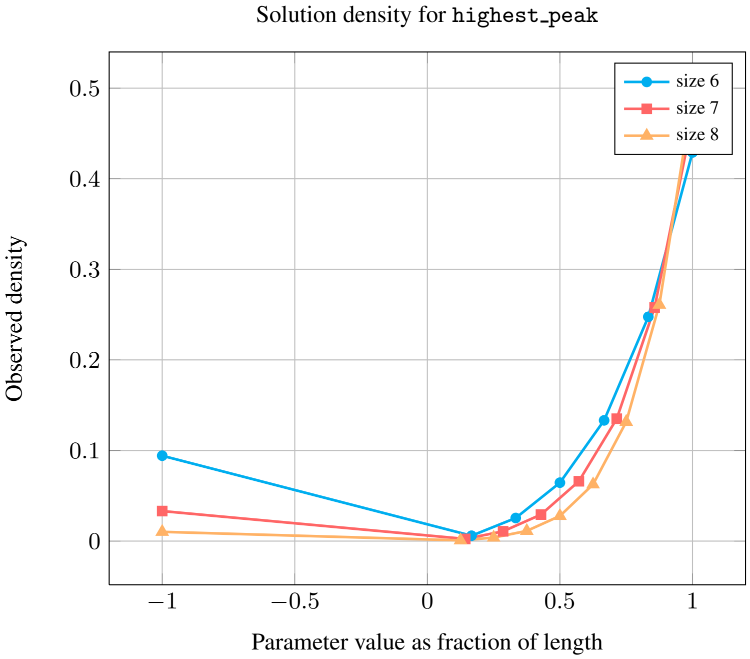




Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	-1000000	9	50	295	1792	11088	69498	439791
	1	-	1	11	92	697	5036	35443
	2	-	4	44	380	3000	22632	166208
	3	-	9	99	900	7587	61389	484020
	4	-	-	176	1712	15680	138544	1195056
	5	-	-	-	2900	29125	283250	2693425
	6	-	-	-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

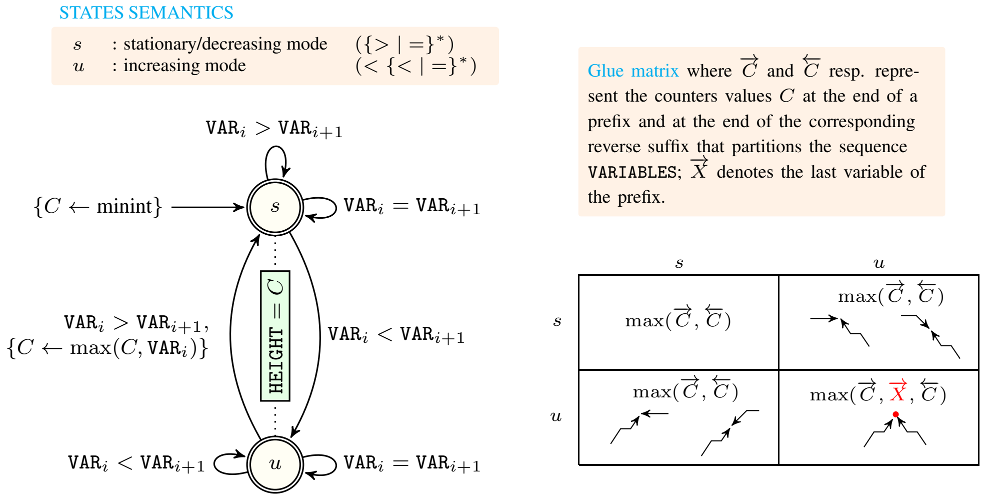
constraint network structure: sliding cyclic(1) constraint network(2).

filtering: glue matrix.

modelling: functional dependency.

Automaton Figure 5.394 depicts the automaton associated with the `highest_peak` constraint. To each pair of consecutive variables $(\mathtt{VAR}_i, \mathtt{VAR}_{i+1})$ of the collection `VARIABLES` corresponds a signature variable S_i. The following signature constraint links $\mathtt{VAR}_i$, $\mathtt{VAR}_{i+1}$ and S_i:

$\mathtt{VAR}_i < \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 0 \wedge \mathtt{VAR}_i = \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 1 \wedge \mathtt{VAR}_i > \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 2.$



STATES SEMANTICS

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

Glue matrix where $\overrightarrow{C}$ 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`; $\overrightarrow{X}$ denotes the last variable of the prefix.

$\mathtt{VAR}_i > \mathtt{VAR}_{i+1}$
$\{C \leftarrow \text{minint}\}$
s
$\mathtt{VAR}_i = \mathtt{VAR}_{i+1}$
$\mathtt{VAR}_i > \mathtt{VAR}_{i+1}$, $\{C \leftarrow \max(C, \mathtt{VAR}_i)\}$
$\mathtt{HEIGHT} = C$
$\mathtt{VAR}_i < \mathtt{VAR}_{i+1}$
$\mathtt{VAR}_i < \mathtt{VAR}_{i+1}$
u
$\mathtt{VAR}_i = \mathtt{VAR}_{i+1}$
s
u
s
$\max(\overrightarrow{C}, \overleftarrow{C})$
$\max(\overrightarrow{C}, \overleftarrow{C})$
u
$\max(\overrightarrow{C}, \overleftarrow{C})$
$\max(\overrightarrow{C}, \overrightarrow{X}, \overleftarrow{C})$

Figure 5.394: 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

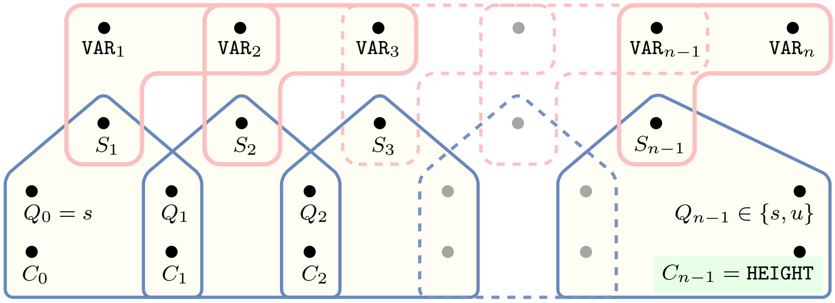


Figure 5.395: Hypergraph of the reformulation corresponding to the automaton of the `highest_peak` constraint (C_0 is set to minint the largest integer that can be represented on a machine)