5.113 deepest_valley

DESCRIPTION LINKS AUTOMATON

Origin Derived from valley.

Constraint deepest_valley(DEPTH, VARIABLES)

Arguments

DEPTH	:	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 *valley* 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 $V_k < V_{k+1}$. DEPTH is the minimum value of the valley variables. If no such variable exists DEPTH is equal to the default value MAXINT.

Example $(2, \langle 5, 3, 4, 8, 8, 2, 7, 1\rangle)$
$(7, \langle 1, 3, 4, 8, 8, 8, 7, 8\rangle)$

The first deepest_valley constraint holds since 2 is the deepest valley of the sequence 5 3 4 8 8 2 7 1.

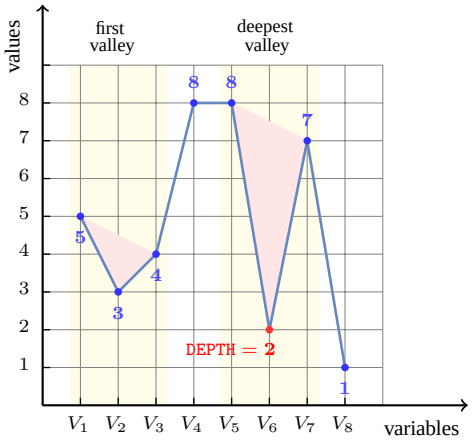


Figure 5.268: Illustration of the first example of the **Example** slot: a sequence of eight variables V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8 respectively fixed to values 5, 3, 4, 8, 8, 2, 7, 1 and its corresponding deepest valley of depth 2

Typical
|VARIABLES| > 2
range(VARIABLES.var) > 2
valley(VARIABLES.var) > 0

Symmetry

Items of `VARIABLES` can be reversed.

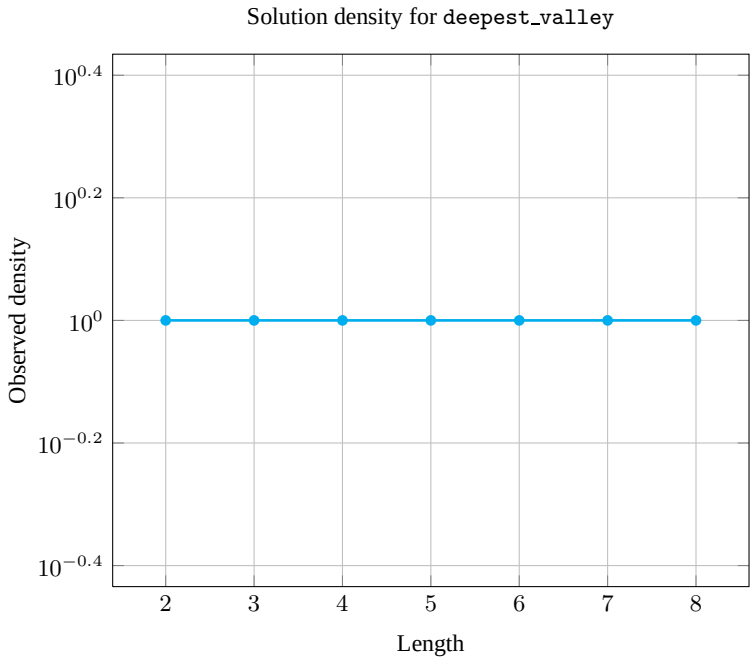
Arg. properties

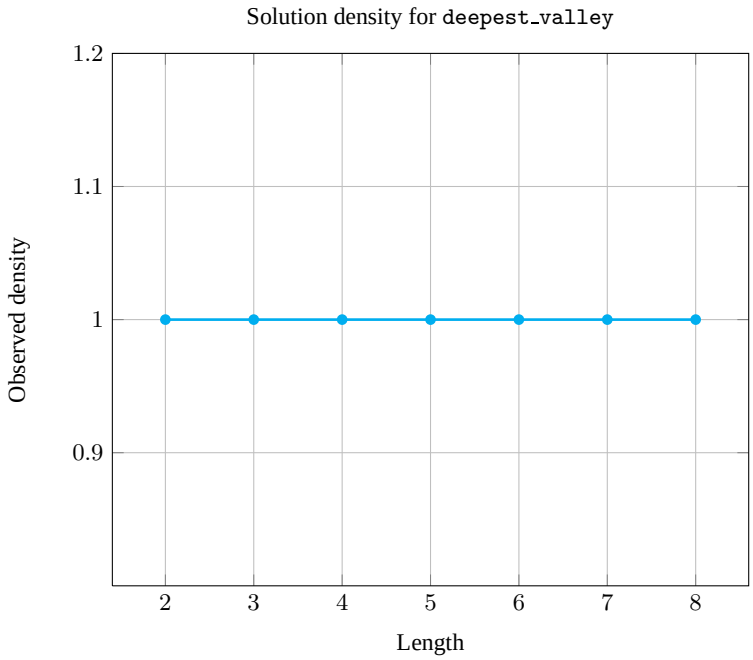
Functional dependency: `DEPTH` determined by `VARIABLES`.

Counting

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

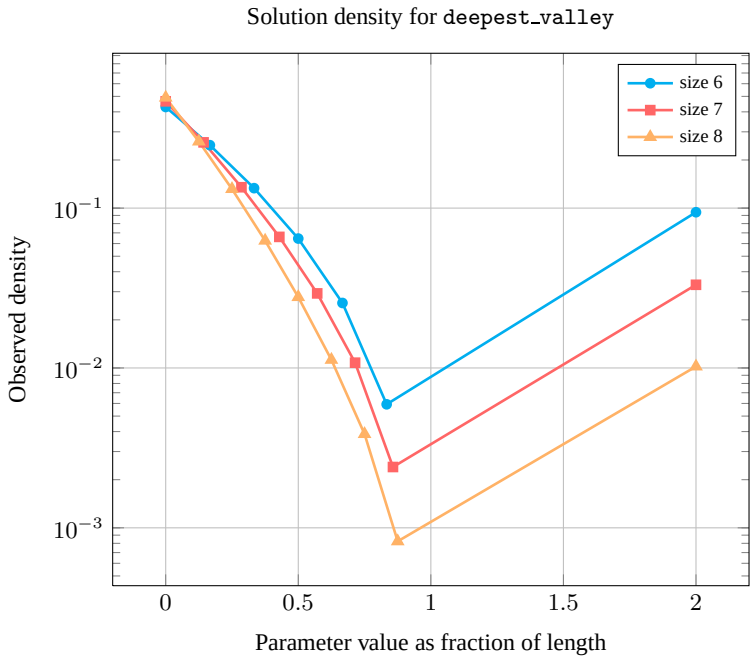
Number of solutions for `deepest_valley`: domains $0..n$

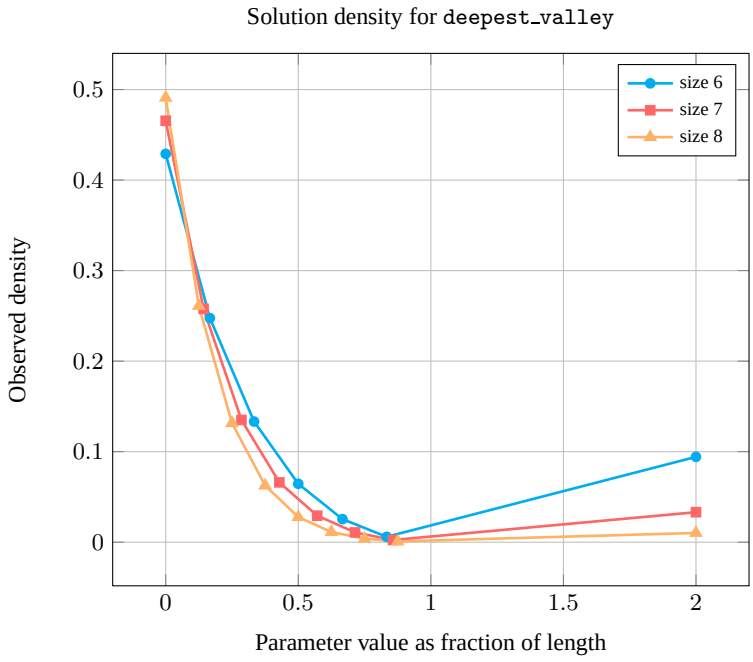




Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	0	-	9	176	2900	50472	976227	21133632
	1	-	4	99	1712	29125	540576	11233250
	2	-	1	44	900	15680	283250	5665896
	3	-	-	11	380	7587	138544	2693425
	4	-	-	-	92	3000	61389	1195056
	5	-	-	-	-	697	22632	484020
	6	-	-	-	-	-	5036	166208
	7	-	-	-	-	-	-	35443
	1000000	9	50	295	1792	11088	69498	439791

Solution count for `deepest_valley`: domains $0..n$





See also

common keyword: `highest_peak`, `valley` *(sequence)*.

implies: `between_min_max`.

Keywords

characteristic of a constraint: maxint, 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 5.269 depicts the automaton associated with the `deepest_valley` constraint. To each pair of consecutive variables $(\texttt{VAR}_i, \texttt{VAR}_{i+1})$ of the collection `VARIABLES` corresponds a signature variable S_i. The following signature constraint links $\texttt{VAR}_i$, $\texttt{VAR}_{i+1}$ and S_i:

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

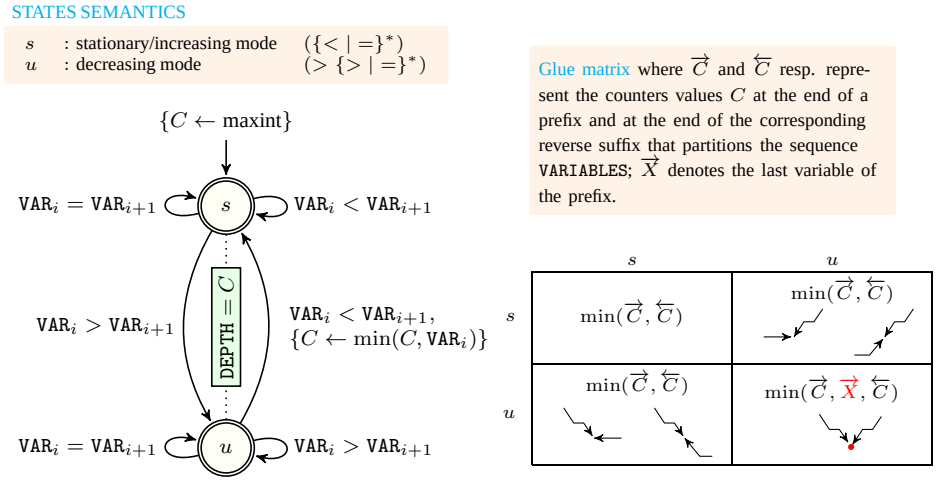


Figure 5.269: Automaton of the `deepest_valley` constraint and its glue matrix (state s means that we are in *increasing* or *stationary* mode, state u means that we are in *decreasing* mode, a new valley is detected each time we switch from decreasing to increasing mode and the counter C is updated accordingly); maxint is the largest integer that can be represented on a machine

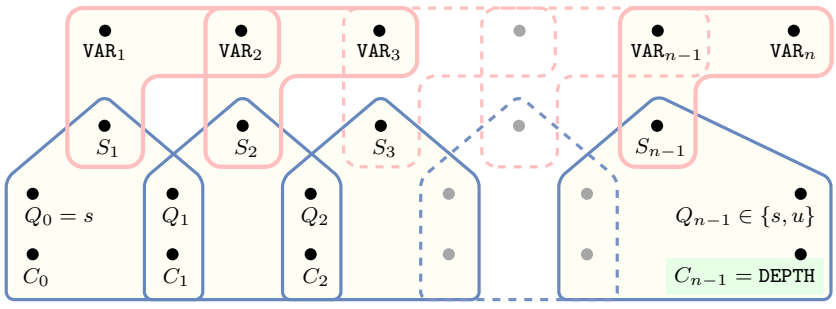


Figure 5.270: Hypergraph of the reformulation corresponding to the automaton of the `deepest_valley` constraint (C_0 is set to maxint the largest integer that can be represented on a machine)