5.189 increasing_peak

DESCRIPTION LINKS AUTOMATON

Origin Derived from peak and increasing.

Constraint increasing_peak(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restrictions |VARIABLES| > 0
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 $V_k > V_{k+1}$.
When considering all the peaks of the sequence VARIABLES from left to right enforce all peaks to be increasing, i.e. the altitude of each peak is greater than or equal to the altitude of its preceding peak when it exists.

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

The increasing_peak constraint holds since the sequence 1 5 **5** 3 **5** 2 2 **7** 4 contains three peaks, in bold, that are increasing.

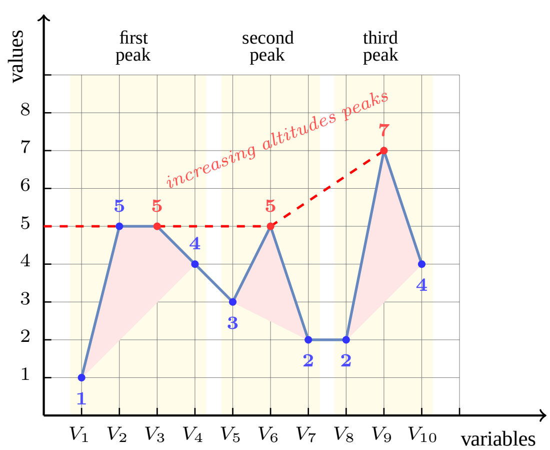


Figure 5.438: Illustration of the **Example** slot: a sequence of ten variables V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10} respectively fixed to values $1, 5, 5, 4, 3, 5, 2, 2, 7, 4$ and its corresponding three peaks, in red, respectively located at altitudes 5, 5 and 7

Typical

$|\texttt{VARIABLES}| \geq 7$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$
$\texttt{peak}(\texttt{VARIABLES.var}) \geq 3$

Symmetry

One and the same constant can be added to the `var` attribute of all items of `VARIABLES`.

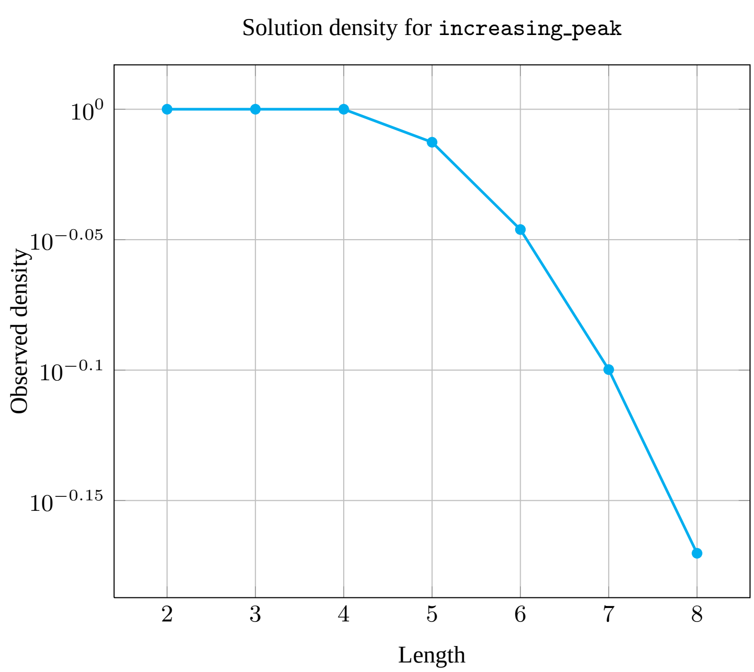
Arg. properties

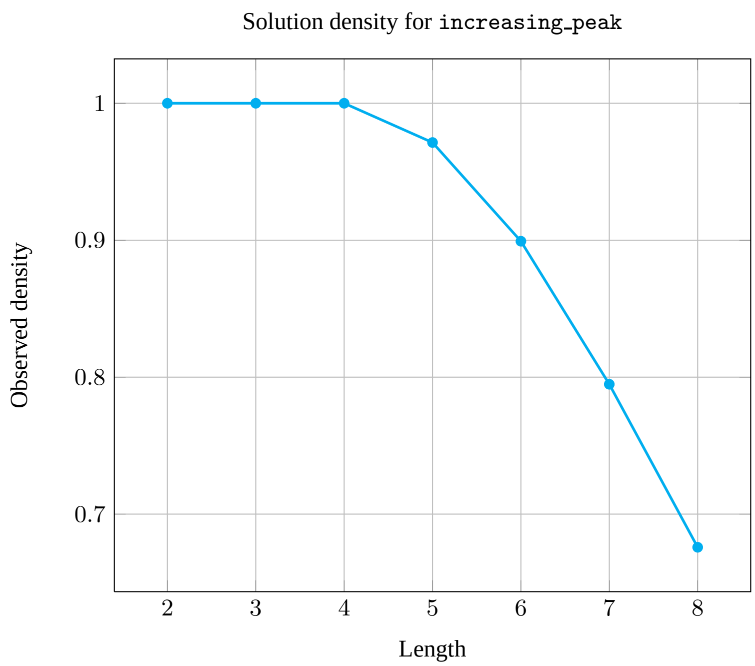
- Prefix-contractible wrt. `VARIABLES`.
- Suffix-contractible wrt. `VARIABLES`.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	9	64	625	7553	105798	1666878	29090469

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





See also

implied by: all_equal_peak.

related: decreasing_peak, peak.

Keywords

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

combinatorial object: sequence.

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

Cond. implications

increasing_peak(VARIABLES)
with peak(VARIABLES.var) > 0
implies not_all_equal(VARIABLES).

Automaton Figure 5.439 depicts the automaton associated with the increasing_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: $(\mathtt{VAR}_i < \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 0) \wedge (\mathtt{VAR}_i = \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 1) \wedge (\mathtt{VAR}_i > \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 2)$.

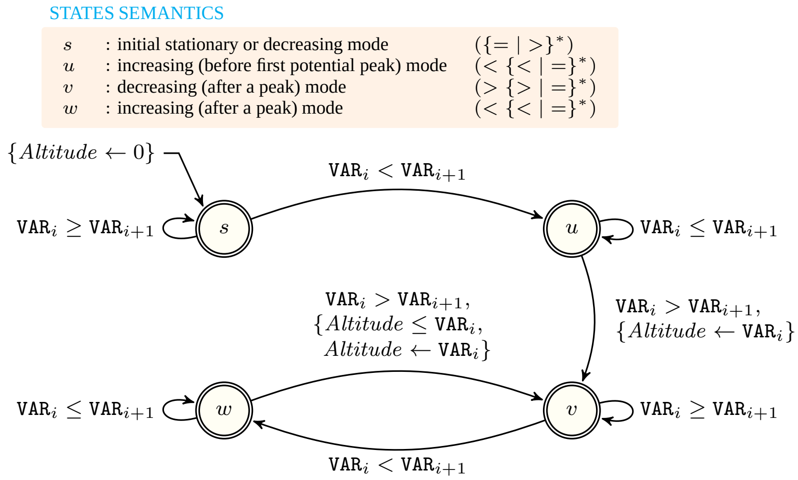


Figure 5.439: Automaton for the increasing_peak constraint (note the conditional transition from state w to state v testing that the counter *Altitude* is less than or equal to $\mathtt{VAR}_i$ for enforcing that all peaks from left to right are in increasing altitude)

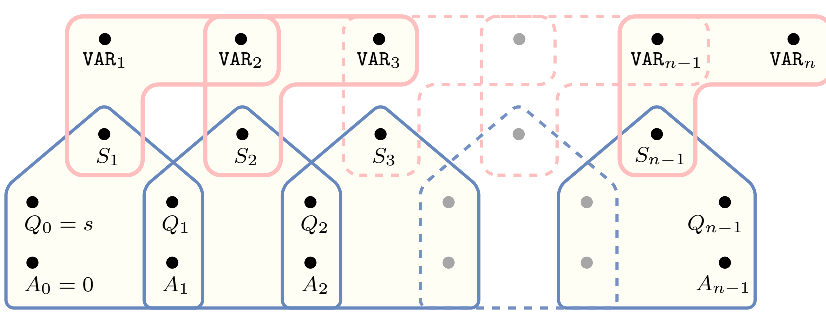


Figure 5.440: Hypergraph of the reformulation corresponding to the automaton of the increasing_peak constraint where A_i stands for the value of the counter *Altitude* (since all states of the automaton are accepting there is no restriction on the last variable Q_{n-1})