5.7 all_equal_peak_max

DESCRIPTION LINKS AUTOMATON

Origin Derived from `peak` and `all_equal`.

Constraint `all_equal_peak_max(VARIABLES)`

Argument `VARIABLES : collection(var-dvar)`

Restrictions $|\texttt{VARIABLES}| > 0$
`required(VARIABLES, var)`

Purpose A variable V_k $(1 < k < m)$ of the sequence of variables $\texttt{VARIABLES} = V_1, \ldots, V_m$ 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 $V_k > V_{k+1}$.
Enforce all the peaks of the sequence VARIABLES to be assigned the same value, i.e. to be located at the same altitude corresponding to the maximum value of the sequence VARIABLES.

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

The `all_equal_peak_max` constraint holds since the two peaks, in bold, of the sequence 1 5 **5** 4 3 **5** 2 5 are located at the same altitude 5 that is also the maximum value of the sequence 1 5 5 4 3 5 2 5. Figure 5.11 depicts the solution associated with the example.

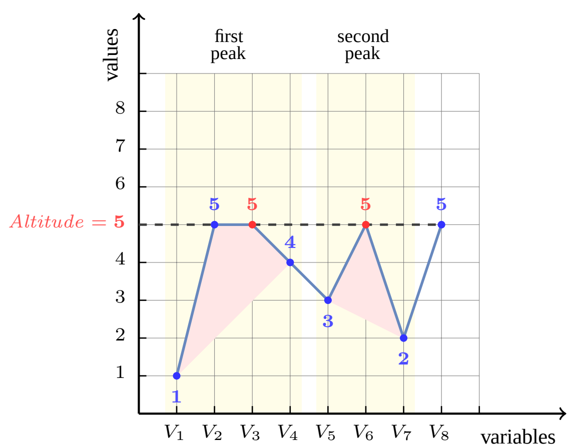


Figure 5.11: Illustration of the **Example** slot: a sequence of eight variables V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8 respectively fixed to values $1, 5, 5, 4, 3, 5, 2, 5$ and its corresponding two peaks, in red, both located at altitude 5 that also corresponds to the maximum value of the sequence

Note that the `all_equal_peak_max` constraint does not enforce that the sequence `VARIABLES` contains at least one peak.

Typical

$|\texttt{VARIABLES}| \geq 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$
$\texttt{peak}(\texttt{VARIABLES.var}) \geq 2$

Symmetries

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

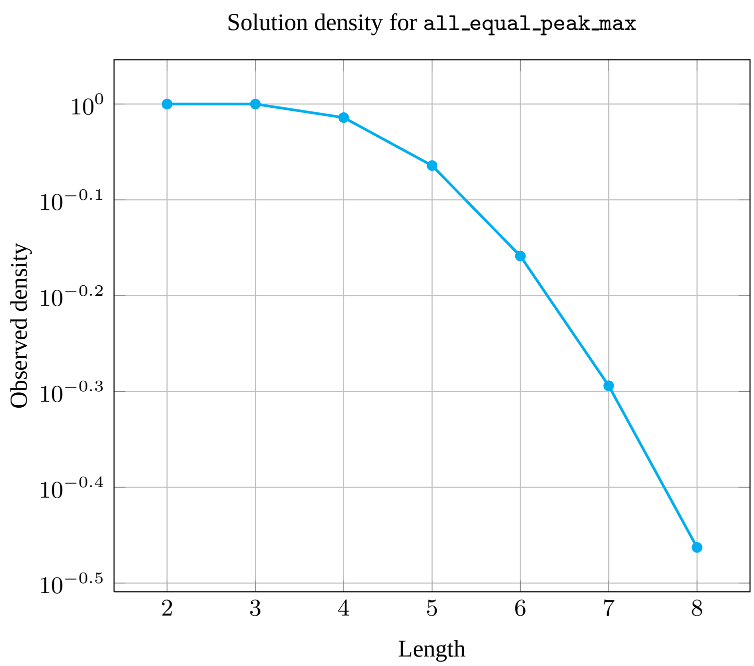
Arg. properties

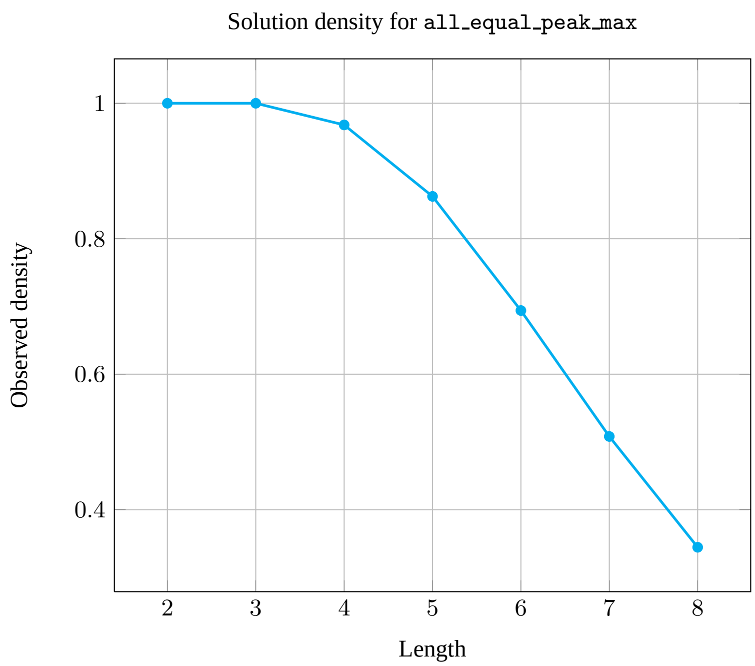
- Prefix-contractible wrt. `VARIABLES`.
- Suffix-contractible wrt. `VARIABLES`.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	9	64	605	6707	81648	1065542	14829903

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





See also

implied by: no_peak.

implies: all_equal_peak.

related: all_equal_valley_min, 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

- all_equal_peak_max(VARIABLES)
  with peak(VARIABLES.var) > 1
  implies some_equal(VARIABLES).

- all_equal_peak_max(VARIABLES)
  with peak(VARIABLES.var) > 0
  implies not_all_equal(VARIABLES).

Automaton Figure 5.12 depicts the automaton associated with the `all_equal_peak_max` 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: $(\mathtt{VAR}_i < \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 0) \wedge (\mathtt{VAR}_i = \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 1) \wedge (\mathtt{VAR}_i > \mathtt{VAR}_{i+1} \Leftrightarrow S_i = 2)$.

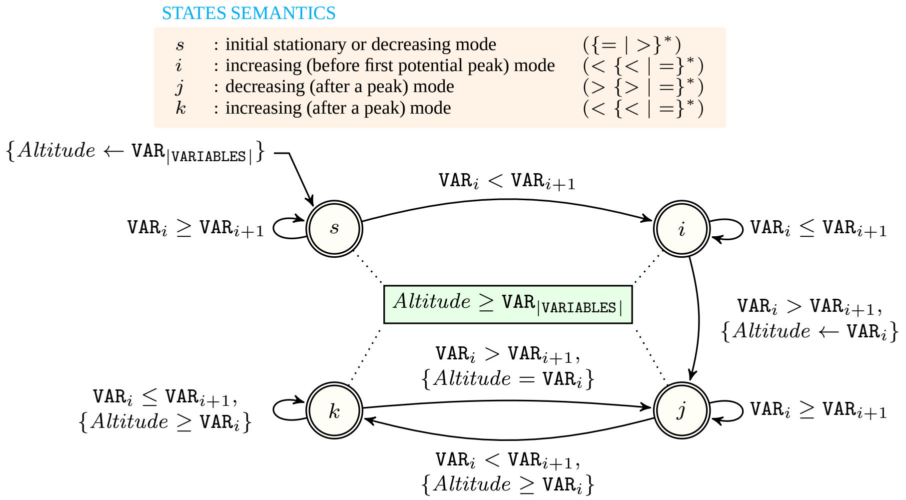


Figure 5.12: Automaton for the `all_equal_peak_max` constraint; note the conditional transition from state k to state j testing that the counter *Altitude* is equal to $\mathtt{VAR}_i$ for enforcing that all peaks are located at the same altitude; the conditional transitions from j to k and from k to k and the final check $Altitude \geq \mathtt{VAR}_{|\mathtt{VARIABLES}|}$ enforce the maximum value of the sequence `VARIABLES` to not exceed the altitude of the eventual peaks.

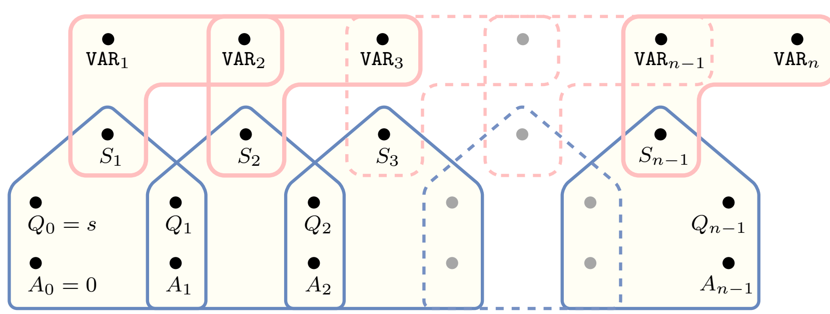


Figure 5.13: Hypergraph of the reformulation corresponding to the automaton of the `all_equal_peak_max` constraint where A stands for the value of the counter *Altitude* (since all states of the automaton are accepting there is no restriction on the last variable Q_{n-1})