5.249 maximum

DESCRIPTION LINKS GRAPH AUTOMATON

Origin CHIP

Constraint maximum(MAX, VARIABLES)

Synonym max.

Arguments

MAX	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

|VARIABLES| > 0
required(VARIABLES, var)

Purpose MAX is the maximum value of the collection of domain variables VARIABLES.

Example

$(7, \langle 3, 2, 7, 2, 6\rangle)$
$(1, \langle 0, 0, 1, 0, 1\rangle)$

The first maximum constraint holds since its first argument MAX = 7 is fixed to the maximum value of the collection $\langle 3, 2, 7, 2, 6\rangle$.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Symmetries

- Items of VARIABLES are permutable.
- All occurrences of two distinct values of VARIABLES.var can be swapped.
- One and the same constant can be added to MAX as well as to the var attribute of all items of VARIABLES.

Arg. properties

- Functional dependency: MAX determined by VARIABLES.
- Aggregate: MAX(max), VARIABLES(union).

Usage In some project scheduling problems one has to introduce dummy activities that correspond for instance to the completion time of a given set of activities. In this context one can use the maximum constraint to get the maximum completion time of a set of tasks.

Remark Note that maximum is a constraint and not just a function that computes the maximum value of a collection of variables: potential values of MAX influence the variables of VARIABLES, and reciprocally potential values that can be assigned to variables of VARIABLES influence MAX.

The maximum constraint is called max in **JaCoP** (http://www.jacop.eu/).

Algorithm A filtering algorithm for the `maximum` constraint is described in [27].

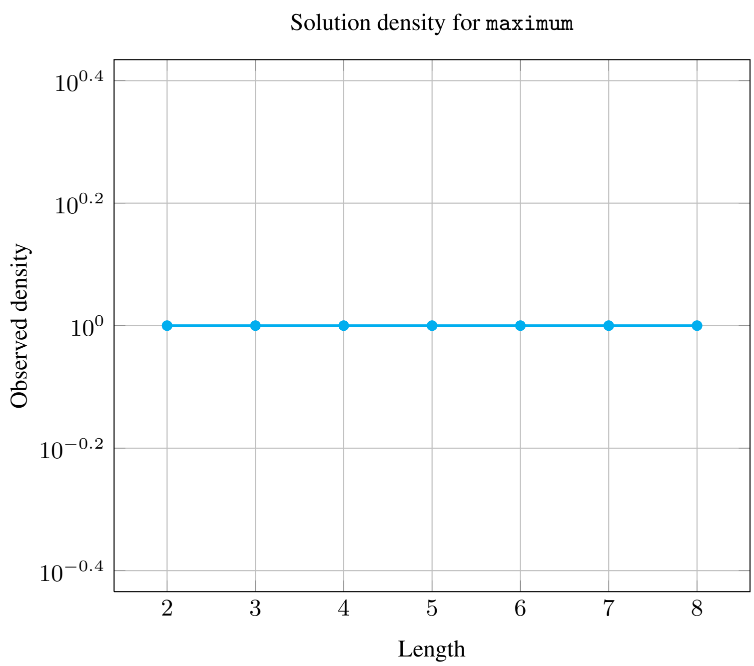
The `maximum` constraint is entailed if all the following conditions hold:

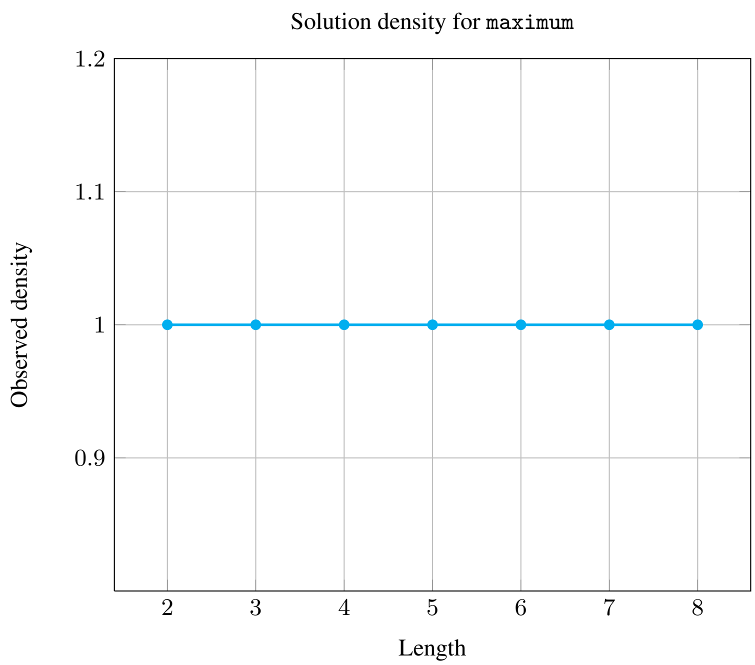
1. `MAX` is fixed.
2. At least one variable of `VARIABLES` is assigned value `MAX`.
3. All variables of `VARIABLES` have their maximum value less than or equal to value `MAX`.

Counting

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

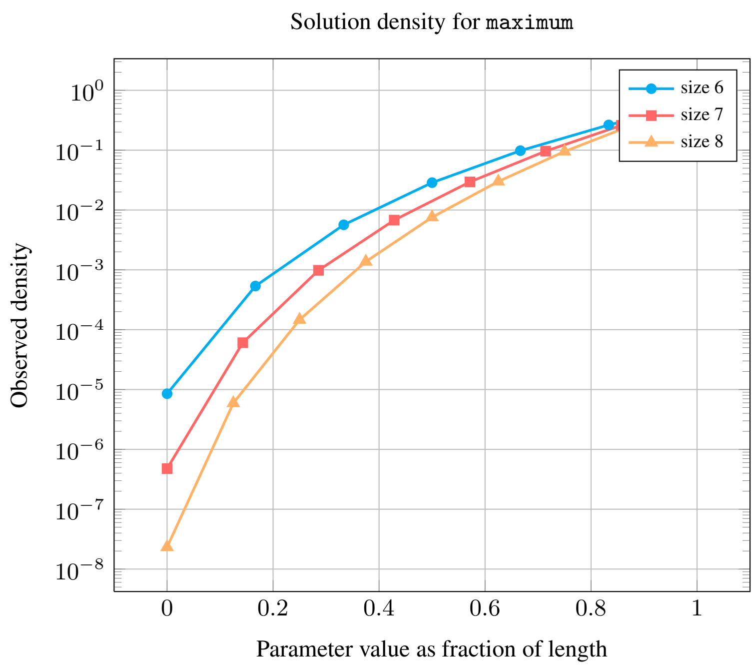
Number of solutions for `maximum`: domains $0..n$

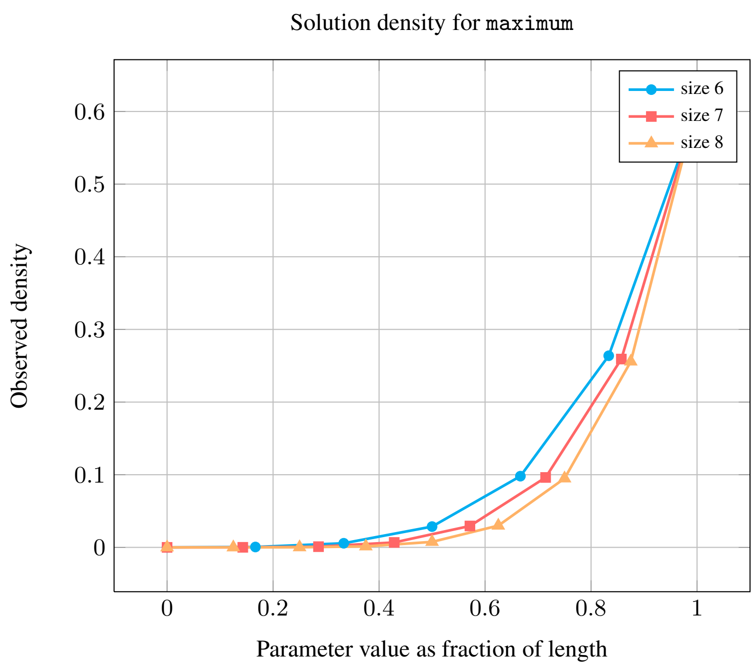




Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	0	1	1	1	1	1	1	1
	1	3	7	15	31	63	127	255
	2	5	19	65	211	665	2059	6305
	3	-	37	175	781	3367	14197	58975
	4	-	-	369	2101	11529	61741	325089
	5	-	-	-	4651	31031	201811	1288991
	6	-	-	-	-	70993	543607	4085185
	7	-	-	-	-	-	1273609	11012415
	8	-	-	-	-	-	-	26269505

Solution count for maximum: domains $0..n$





Systems max in **Choco**, max in **Gecode**, max in **JaCoP**, maximum in **MiniZinc**, maximum in **SICStus**.

See also

common keyword: `minimum` *(order constraint)*.

comparison swapped: `minimum`.

generalisation: `maximum_modulo` *(*`variable` *replaced by* `variable` mod `constant`*)*.

implied by: `or`.

implies: `between_min_max`, `in`.

soft variant: `open_maximum` *(open constraint)*.

specialisation: `max_n` *(maximum or order* `n` *replaced by absolute maximum)*.

uses in its reformulation: `tree_range`.

Keywords

characteristic of a constraint: maximum, automaton, automaton without counters, reified automaton constraint.

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

constraint network structure: centered cyclic(1) constraint network(1).

constraint type: order constraint.

filtering: glue matrix, arc-consistency, entailment.

modelling: balanced assignment, functional dependency.

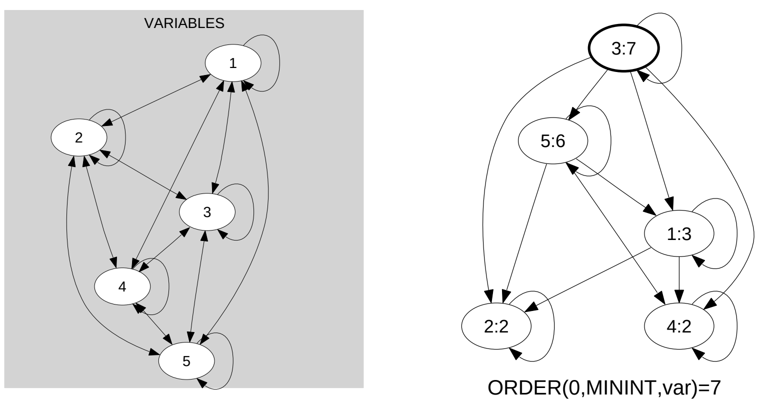
Cond. implications

`maximum`(MAX, VARIABLES)
with `first`(VARIABLES.var) $<$ MAX
and `last`(VARIABLES.var) $<$ MAX
implies `highest_peak`(HEIGHT, VARIABLES).

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	$\bigvee\left(\begin{array}{l}\texttt{variables1.key} = \texttt{variables2.key},\\ \texttt{variables1.var} > \texttt{variables2.var}\end{array}\right)$
Graph property(ies)	**ORDER**(0, MININT, var) = MAX

Graph model We use a similar definition that the one that was utilised for the minimum constraint. Within the arc constraint, we replace the comparison operator $<$ by $>$.

Parts (A) and (B) of Figure 5.520 respectively show the initial and final graph associated with the first example of the **Example** slot. Since we use the **ORDER** graph property, the vertex of rank 0 (without considering the loops) of the final graph is outlined with a thick circle.



(A) **(B)**

Figure 5.520: Initial and final graph of the maximum constraint

Automaton Figure 5.521 depicts the automaton associated with the `maximum` constraint. Let $\texttt{VAR}_i$ be the i^{th} variable of the `VARIABLES` collection. To each pair $(\texttt{MAX}, \texttt{VAR}_i)$ corresponds a signature variable S_i as well as the following signature constraint: $(\texttt{MAX} > \texttt{VAR}_i \Leftrightarrow S_i = 0) \wedge (\texttt{MAX} = \texttt{VAR}_i \Leftrightarrow S_i = 1) \wedge (\texttt{MAX} < \texttt{VAR}_i \Leftrightarrow S_i = 2)$.

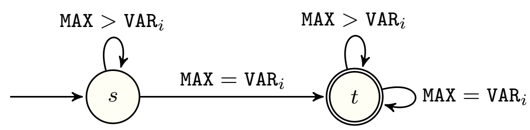


Figure 5.521: Counter free automaton of the `maximum` constraint

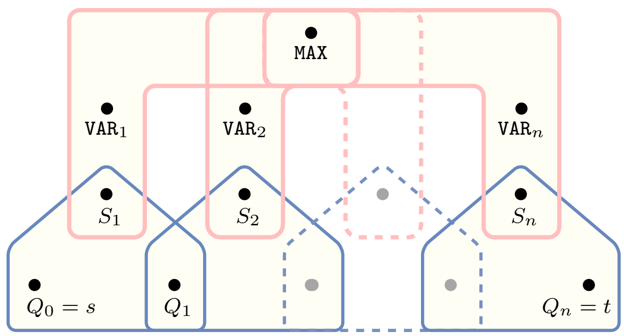


Figure 5.522: Hypergraph of the reformulation corresponding to the automaton of the `maximum` constraint

Figure 5.522 depicts a second counter free non deterministic automaton associated with the `maximum` constraint, where the argument `MAX` is also part of the sequence passed to the automaton.

Figure 5.525 depicts a third deterministic automaton with one counter associated with the `maximum` constraint, where the argument `MAX` is unified to the final value of the counter.

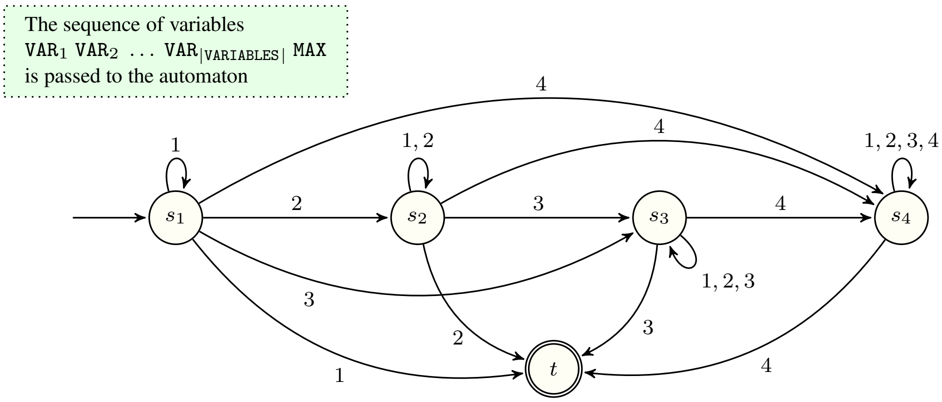


Figure 5.523: Counter free non deterministic automaton of the maximum(MAX, VARIABLES) constraint assuming that the union of the domain of the variables is the set $\{1, 2, 3, 4\}$ and that the elements of VARIABLES are first passed to the automaton followed by MAX (state s_i means that no value strictly greater than value i was found and that value i was already encountered at least once)

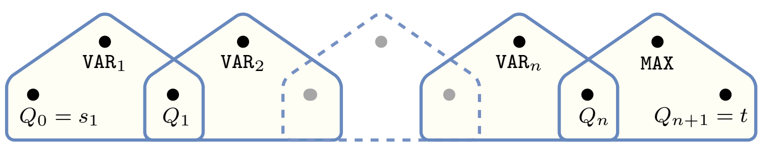


Figure 5.524: Hypergraph of the reformulation corresponding to the counter free non deterministic automaton of the maximum constraint

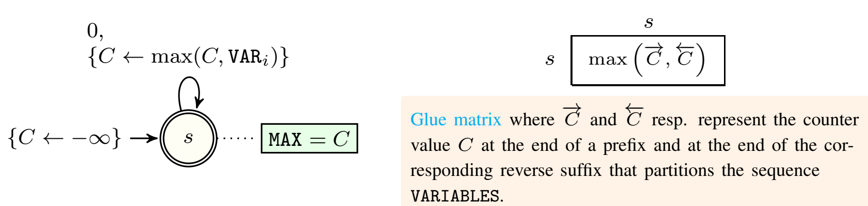


Figure 5.525: Automaton (with one counter) of the maximum constraint and its glue constraint

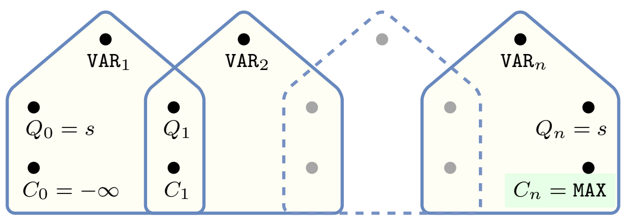


Figure 5.526: Hypergraph of the reformulation corresponding to the automaton (with one counter) of the `maximum` constraint: since all states variables $Q_0, Q_1, \ldots, Q_n$ are fixed to the unique state s of the automaton, the transitions constraints share only the counter variable C and the constraint network is Berge-acyclic