5.210 maximum

DESCRIPTION LINKS GRAPH AUTOMATON

Origin CHIP

Constraint maximum(MAX, VARIABLES)

Synonym max.

Arguments

MAX	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

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

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

Example $(7, \langle 3, 2, 7, 2, 6 \rangle)$

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

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.

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 [26].

Systems max in **Choco**, max in **Gecode**, max in **JaCoP**, 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 network structure: centered cyclic(1) constraint network(1).

constraint type: order constraint.

filtering: arc-consistency.

modelling: balanced assignment.

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)	$\mathbf{ORDER}(0, \texttt{MININT}, \texttt{var}) = \texttt{MAX}$

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 5.405 respectively show the initial and final graph associated with 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.

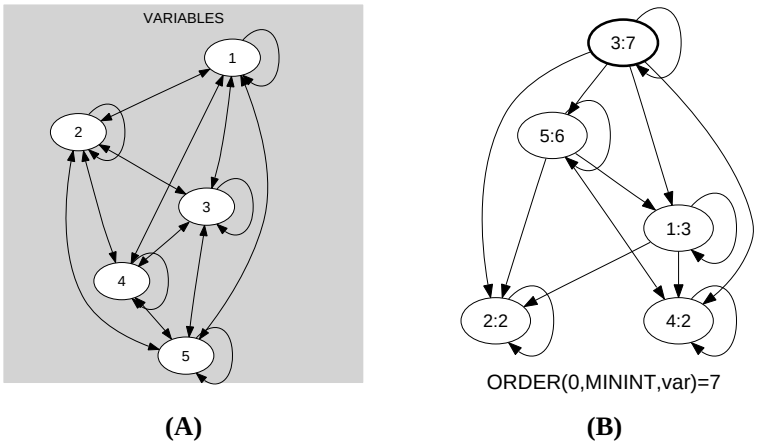

Figure 5.405: Initial and final graph of the maximum constraint

Automaton

Figure 5.406 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 $\texttt{S}_i$ as well as the following signature constraint: $(\texttt{MAX} > \texttt{VAR}_i \Leftrightarrow \texttt{S}_i = 0) \wedge (\texttt{MAX} = \texttt{VAR}_i \Leftrightarrow \texttt{S}_i = 1) \wedge (\texttt{MAX} < \texttt{VAR}_i \Leftrightarrow \texttt{S}_i = 2)$.

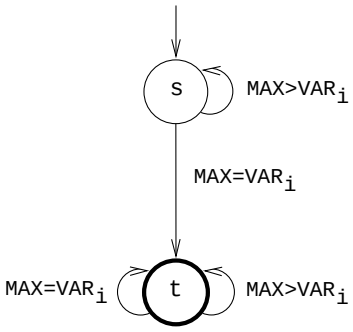

Figure 5.406: Automaton of the `maximum` constraint

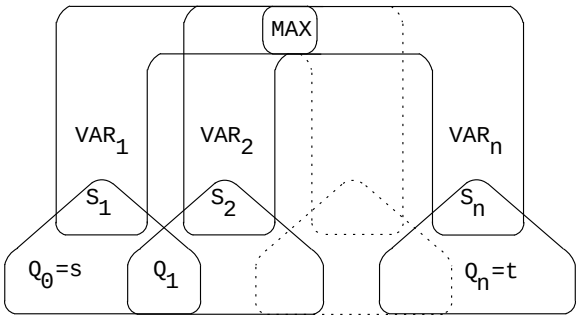

Figure 5.407: Hypergraph of the reformulation corresponding to the automaton of the `maximum` constraint