5.383 sum

DESCRIPTION LINKS GRAPH

Origin [444].

Constraint sum(INDEX, SETS, CONSTANTS, S)

Synonym sum_pred.

Arguments

INDEX	:	dvar
SETS	:	collection(ind−int, set−sint)
CONSTANTS	:	collection(cst−int)
S	:	dvar

Restrictions

$|\texttt{SETS}| \geq 1$
required(SETS, [ind, set])
distinct(SETS, ind)
$|\texttt{CONSTANTS}| \geq 1$
required(CONSTANTS, cst)

Purpose S is equal to the sum of the constants of CONSTANTS corresponding to the $\texttt{INDEX}^{th}$ set of the SETS collection.

Example

$$\left(8, \left\langle \begin{array}{ll} \texttt{ind} - 8 & \texttt{set} - \{2,3\}, \\ \texttt{ind} - 1 & \texttt{set} - \{3\}, \\ \texttt{ind} - 3 & \texttt{set} - \{1,4,5\}, \\ \texttt{ind} - 6 & \texttt{set} - \{2,4\} \end{array} \right\rangle, \langle 4,9,1,3,1 \rangle, 10 \right)$$

The sum constraint holds since its last argument $\texttt{S} = 10$ is equal to the sum of the 2^{th} and 3^{th} items of the collection $\langle 4,9,1,3,1 \rangle$. As illustrated by Figure 5.744, this stems from the fact that its first argument $\texttt{INDEX} = 8$ corresponds to the value of the ind attribute of the first item of the SETS collection. Consequently the corresponding set $\{2,3\}$ is used for summing the 2^{th} and 3^{th} items of the CONSTANTS collection.

Typical

$|\texttt{SETS}| > 1$
$|\texttt{CONSTANTS}| > |\texttt{SETS}|$
$\texttt{range}(\texttt{CONSTANTS.cst}) > 1$

Symmetry Items of SETS are permutable.

Arg. properties

Functional dependency: S determined by INDEX, SETS and CONSTANTS.

Usage In his article introducing the sum constraint, Tallys H. Yunes mentions the *Sequence Dependent Cumulative Cost Problem* as the subproblem that originally motivates this constraint.

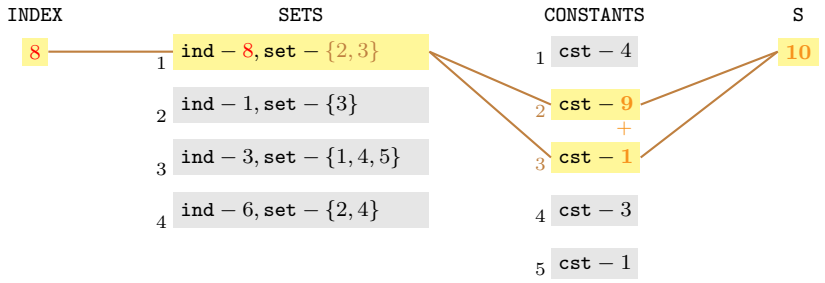


Figure 5.744: Illustration of the correspondence between the arguments of the sum(INDEX, SETS, CONSTANTS, S) constraint in the context of the **Example** slot (from right to left, S = 10 is equal to the sum of the constants 9 and 1 corresponding to the indices 2 and 3 of the set for which the ind attribute is equal to INDEX = 8)

Remark The sum constraint is called sum_pred in **MiniZinc** (http://www.minizinc.org/).

Algorithm The article [444] gives the convex hull relaxation of the sum constraint.

Systems sum_pred in **MiniZinc**.

See also **common keyword:** element (*data constraint*), sum_ctr, sum_set (*sum*).
used in graph description: in_set.

Keywords **characteristic of a constraint:** convex hull relaxation, sum.
constraint type: data constraint.
filtering: linear programming.
modelling: functional dependency.

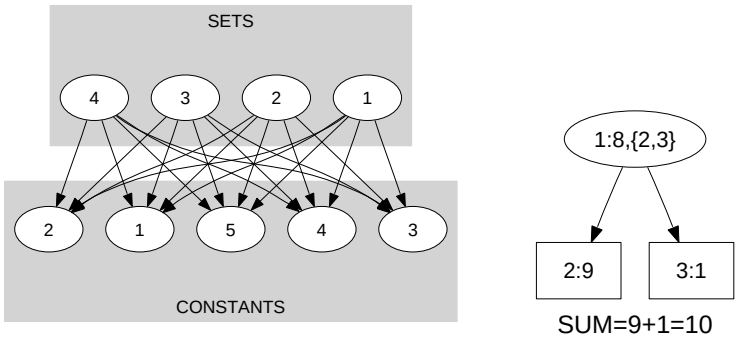
Arc input(s)	SETS CONSTANTS
Arc generator	$PRODUCT \mapsto$ collection(sets, constants)
Arc arity	2
Arc constraint(s)	• INDEX = sets.ind • in_set(constants.key, sets.set)
Graph property(ies)	SUM(CONSTANTS, cst) = S

Graph model According to the value assigned to INDEX the arc constraint selects for the final graph:

- The $\texttt{INDEX}^{th}$ item of the SETS collection,
- The items of the CONSTANTS collection for which the key correspond to the indices of the $\texttt{INDEX}^{th}$ set of the SETS collection.

Finally, since we use the **SUM** graph property on the cst attribute of the CONSTANTS collection, the last argument S of the sum constraint is equal to the sum of the constants associated with the vertices of the final graph.

Parts (A) and (B) of Figure 5.745 respectively show the initial and final graph associated with the **Example** slot. Since we use the **SUM** graph property we show the vertices from which we compute S in a box.



(A) **(B)**

Figure 5.745: Initial and final graph of the sum constraint