5.326 sum_ctr

DESCRIPTION **LINKS** **GRAPH**

Origin Arithmetic constraint.

Constraint `sum_ctr(VARIABLES, CTR, VAR)`

Synonyms `constant_sum`, `sum`, `linear`, `scalar_product`.

Arguments

```
VARIABLES : collection(var−dvar)
CTR       : atom
VAR       : dvar
```

Restrictions

`required(VARIABLES, var)`
`CTR` $\in [=, \neq, <, \geq, >, \leq]$

Purpose Constraint the sum of a set of domain variables. More precisely, let `S` denote the sum of the variables of the `VARIABLES` collection (when the collection is empty the corresponding sum is equal to 0). Enforce the following constraint to hold: `S` `CTR` `VAR`.

Example $(\langle 1, 1, 4\rangle, =, 6)$

The `sum_ctr` constraint holds since the condition $1 + 1 + 4 = 6$ is satisfied.

Symmetry Items of `VARIABLES` are permutable.

Remark When `CTR` corresponds to $=$ this constraint is referenced under the names `constant_sum` in KOALOG (`http://www.koalog.com/php/index.php`) and `sum` in **JaCoP** (`http://www.jacop.eu/`).

Systems `equation` in **Choco**, `linear` in **Gecode**, `scalar_product` in **SICStus**.

Used in `bin_packing`, `cumulative`, `cumulative_convex`, `cumulative_with_level_of_priority`, `cumulatives`, `indexed_sum`, `interval_and_sum`, `relaxed_sliding_sum`, `sliding_sum`, `sliding_time_window_sum`.

See also

assignment dimension added: `interval_and_sum` (*assignment dimension corresponding to intervals is added*).

common keyword: `arith_sliding`, `product_ctr`, `range_ctr` (*arithmetic constraint*), `sum` (*sum*), `sum_set` (*arithmetic constraint*).

generalisation: `scalar_product` (*arithmetic constraint where all coefficients are not necessarly equal to* 1).

system of constraints: `sliding_sum`.

Keywords

characteristic of a constraint: sum.

constraint type: arithmetic constraint.

Arc input(s)	`VARIABLES`
Arc generator	$SELF \mapsto$ `collection(variables)`
Arc arity	1
Arc constraint(s)	`TRUE`
Graph property(ies)	**SUM**(`VARIABLES`, `var`) `CTR` `VAR`

Graph model Since we want to keep all the vertices of the initial graph we use the $SELF$ arc generator together with the `TRUE` arc constraint. This predefined arc constraint always holds.

Parts (A) and (B) of Figure 5.591 respectively show the initial and final graph associated with the **Example** slot. Since we use the `TRUE` arc constraint both graphs are identical.

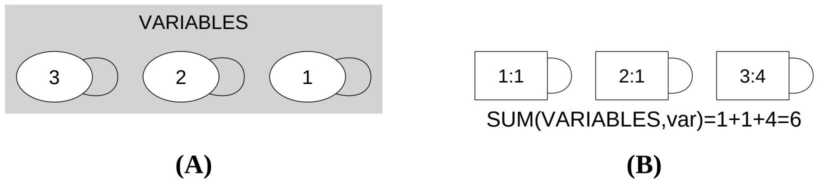

Figure 5.591: Initial and final graph of the `sum_ctr` constraint