5.143   element_product

Origin   [300]

Constraint   `element_product(Y, TABLE, X, Z)`

Synonym   `element.`

Arguments
```
Y     :  dvar
TABLE :  collection(value−int)
X     :  dvar
Z     :  dvar
```

Restrictions
$$Y \geq 1$$
$$Y \leq |\texttt{TABLE}|$$
$$X \geq 0$$
$$Z \geq 0$$
`required(TABLE, value)`
$$\texttt{TABLE.value} \geq 0$$

Purpose   Z is equal to the $\texttt{Y}^{th}$ item of TABLE multiplied by X.

Example $(3, \langle 6, 9, 2, 9 \rangle, 5, 10)$

The `element_product` constraint holds since its fourth argument Z $=$ 10 is equal to the 3th (Y $=$ 3) item of the collection $\langle 6, 9, 2, 9 \rangle$ multiplied by X $=$ 5.

Typical
$$X > 0$$
$$Z > 0$$
$$|\texttt{TABLE}| > 1$$
`range(TABLE.value) > 1`
$$\texttt{TABLE.value} > 0$$

Arg. properties

- Functional dependency: Z determined by Y, TABLE and X.

- Suffix-extensible wrt. TABLE.

Usage   The `element_product` constraint was originally used in configuration problems [300]. In this context, Z denotes the cost of buying X units of type Y at cost TABLE[Y].value.

Reformulation   By introducing an extra variable VAL, the `element_product(Y, TABLE, X, Z)` constraint can be expressed in term of an `element(Y, TABLE, VAL)` constraint and of a product constraint Z $=$ VAL $\cdot$ X.

See also   **common keyword:**   elem,   element,   element_greatereq, element_lesseq *(array constraint)*.

Keywords

application area: configuration problem.

constraint arguments: pure functional dependency.

constraint type: data constraint.

modelling: array constraint, table, functional dependency, variable subscript.

Derived Collection

$$\text{col} \left(\begin{array}{l} \texttt{ITEM}-\texttt{collection}(\texttt{y}-\texttt{dvar}, \texttt{x}-\texttt{dvar}, \texttt{z}-\texttt{dvar}), \\ [\texttt{item}(\texttt{y} - \texttt{Y}, \texttt{x} - \texttt{X}, \texttt{z} - \texttt{Z})] \end{array} \right)$$

Arc input(s)	ITEM TABLE
Arc generator	$PRODUCT \mapsto \texttt{collection}(\texttt{item}, \texttt{table})$
Arc arity	2
Arc constraint(s)	• item.y = table.key • item.z = item.x * table.value
Graph property(ies)	**NARC**= 1

Graph model

We use the derived collection ITEM for putting together the Y, the X and Z parameters of the element_product constraint. Within the arc constraint we use the implicit attribute key that associates to each item of a collection its position within the collection.

Parts (A) and (B) of Figure 5.314 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the unique arc of the final graph is stressed in bold.
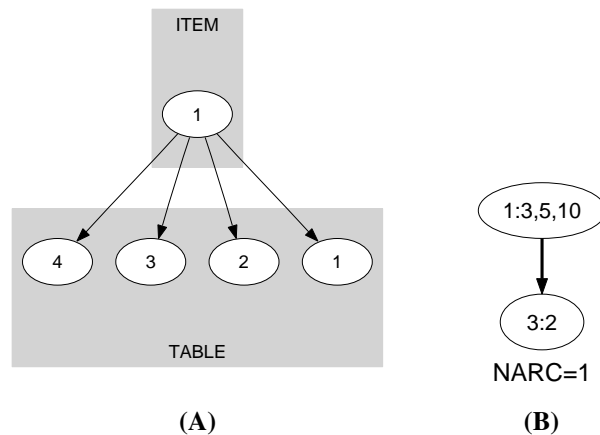

Figure 5.314: Initial and final graph of the element_product constraint

Signature

Because of the first condition of the arc constraint the final graph cannot have more than one arc. Therefore we can rewrite **NARC** $= 1$ to **NARC** ≥ 1 and simplify $\overline{\textbf{NARC}}$ to **NARC**.