## 5.145 elementn

DESCRIPTION LINKS AUTOMATON

**Origin** P. Flener

**Constraint** elementn(INDEX, TABLE, ENTRIES)

**Arguments**

| | | |
|---|---|---|
| INDEX | : | dvar |
| TABLE | : | collection(value−int) |
| ENTRIES | : | collection(entry−dvar) |

**Restrictions**

$\texttt{INDEX} \geq 1$
$\texttt{INDEX} \leq |\texttt{TABLE}| - |\texttt{ENTRIES}| + 1$
$|\texttt{TABLE}| > 0$
$|\texttt{ENTRIES}| > 0$
$|\texttt{TABLE}| \geq |\texttt{ENTRIES}|$
required(TABLE, value)
required(ENTRIES, entry)

**Purpose** $\forall i \in [1, |\texttt{ENTRIES}|] : \texttt{ENTRIES}[i].\texttt{entry} = \texttt{TABLE}[\texttt{INDEX} + i - 1].\texttt{value}$

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

The elementn constraint holds since its third argument $\texttt{ENTRIES} = \langle 2, 9\rangle$ is set to the subsequence starting at the third (i.e., $\texttt{INDEX} = 3$) item of the table $\texttt{TABLE} = \langle 6, 9, 2, 9\rangle$.

**Typical**

$|\texttt{TABLE}| > 1$
$\text{range}(\texttt{TABLE.value}) > 1$
$|\texttt{ENTRIES}| > 1$

**Symmetry** All occurrences of two distinct values in TABLE.value or ENTRIES.entry can be swapped; all occurrences of a value in TABLE.value or ENTRIES.entry can be renamed to any unused value.

**Arg. properties**

Suffix-extensible wrt. TABLE.

**Usage** The elementn constraint is useful for extracting of subsequence of fixed length from a given sequence.

**Reformulation** Let $I_1 = \texttt{INDEX}, I_2 = \texttt{INDEX} + 1, \ldots, I_{|\texttt{ENTRIES}|} = \texttt{INDEX} + |\texttt{ENTRIES}| - 1$. The elementn(INDEX, TABLE, $\langle \texttt{entry} - E_1, \texttt{entry} - E_2, \ldots, \texttt{entry} - E_{|\texttt{ENTRIES}|}\rangle$) constraint can be expressed in term of a conjunction of $|\texttt{ENTRIES}|$ element constraints of the form:
element($I_1$, TABLE, $E_1$),
element($I_2$, TABLE, $E_2$),
...
element(INDEX + $|\texttt{ENTRIES}| - 1$, TABLE, $E_{|\texttt{ENTRIES}|}$).

**See also** **common keyword:** `element` *(data constraint)*.

**Keywords** **characteristic of a constraint:** automaton, automaton without counters, reified automaton constraint.

**constraint network structure:** Berge-acyclic constraint network.

**constraint type:** data constraint, sliding sequence constraint.

**filtering:** arc-consistency.

**modelling:** table.

**Automaton**

Figure 5.335 depicts the automaton associated with the `elementn` constraint of the **Example** slot. Let `I` and $\texttt{E}_k$ respectively denote the `INDEX` argument and the `entry` attribute of the $k^{th}$ item of the `ENTRIES` collection. Figure 5.336 depicts the reformulation of the `elementn` constraint.

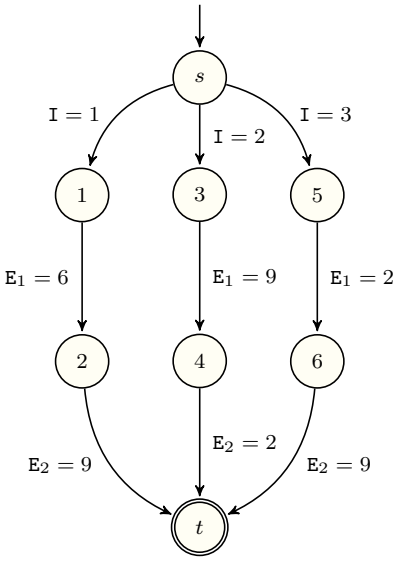


Figure 5.335: Automaton of the `elementn` constraint given in the example

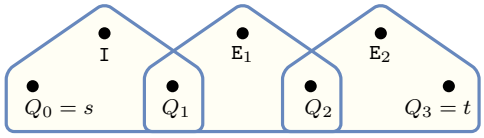


Figure 5.336: Hypergraph of the reformulation corresponding to the automaton of the `elementn` constraint