

5.144 `element_sparse`

| | DESCRIPTION | LINKS | GRAPH | AUTOMATON |
|--------------|---|-------|-------|-----------|
| Origin | CHIP | | | |
| Constraint | <code>element_sparse</code> (ITEM, TABLE, DEFAULT) | | | |
| Usual name | element | | | |
| Arguments | ITEM : <code>collection</code> (index-dvar, value-dvar) TABLE : <code>collection</code> (index-int, value-int) DEFAULT : <code>int</code> | | | |
| Restrictions | <code>required</code> (ITEM, [index, value]) ITEM.index ≥ 1 ITEM = 1 TABLE > 0 <code>required</code> (TABLE, [index, value]) TABLE.index ≥ 1 <code>distinct</code> (TABLE, index) | | | |
| Purpose | ITEM[1].value is equal to one of the entries of the table TABLE or to the default value DEFAULT if the entry ITEM[1].index does not exist in TABLE. | | | |
| Example | $\left(\begin{array}{l} \langle \text{index} - 2 \text{ value} - 5 \rangle, \\ \text{index} - 1 \quad \text{value} - 6, \\ \left\langle \begin{array}{l} \text{index} - 2 \quad \text{value} - 5, \\ \text{index} - 4 \quad \text{value} - 2, \\ \text{index} - 8 \quad \text{value} - 9 \end{array} \right\rangle, 5 \end{array} \right)$ | | | |
| | The <code>element_sparse</code> constraint holds since its first argument ITEM corresponds to the second item of the TABLE collection. | | | |
| Typical | TABLE > 1 <code>range</code> (TABLE.value) > 1 | | | |
| Symmetries | <ul style="list-style-type: none">Items of TABLE are <code>permutable</code>.All occurrences of two distinct values in ITEM.value, TABLE.value or DEFAULT can be <code>swapped</code>; all occurrences of a value in ITEM.value, TABLE.value or DEFAULT can be <code>renamed</code> to any unused value. | | | |
| Usage | A sometimes more compact form of the <code>element</code> constraint: we are not obliged to specify explicitly the table entries that correspond to the specified default value. This can sometimes reduce drastically memory utilisation. | | | |
| Remark | The original constraint of <code>CHIP</code> had an additional parameter SIZE giving the maximum value of ITEM.index. | | | |

Reformulation

Let I and V respectively denote $ITEM[1].index$ and $ITEM[1].value$. The `element_sparse`($ITEM, TABLE, DEFAULT$) constraint can be expressed in term of a reified constraint of the form:

$$\begin{aligned} & ((I = TABLE[1].index \wedge V = TABLE[1].value) \vee \\ & (I = TABLE[2].index \wedge V = TABLE[2].value) \vee \\ & \dots \\ & (I = TABLE[|TABLE|].index \wedge V = TABLE[|TABLE|].value)) \vee \\ & ((I \neq TABLE[1].index) \wedge \\ & (I \neq TABLE[2].index) \wedge \\ & \dots \\ & (I \neq TABLE[|TABLE|].index) \wedge \\ & (V = DEFAULT)). \end{aligned}$$
See also

common keyword: `elem`, `element` (*array constraint*), `elements_sparse` (*sparse table*).

implies: `elements_sparse`.

system of constraints: `elements_sparse`.

Keywords

characteristic of a constraint: `automaton`, `automaton without counters`, reified automaton constraint, derived collection.

constraint arguments: binary constraint.

constraint network structure: centered cyclic(2) constraint network(1).

constraint type: data constraint.

filtering: arc-consistency.

modelling: `array constraint`, `table`, `sparse table`, `sparse functional dependency`, `variable indexing`.

Derived Collections

```
col ( DEF-collection(index-int, value-int), )  
      [item(index - 0, value - DEFAULT)]  
col ( TABLE_DEF-collection(index-dvar, value-dvar),  
      [ item(index - TABLE.index, value - TABLE.value),  
        item(index - DEF.index, value - DEF.value) ] )
```

| | |
|---------------------|---|
| Arc input(s) | ITEM TABLE_DEF |
| Arc generator | <i>PRODUCT</i> \mapsto collection(item, table_def) |
| Arc arity | 2 |
| Arc constraint(s) | <ul style="list-style-type: none">• item.value = table_def.value• item.index = table_def.index \vee table_def.index = 0 |
| Graph property(ies) | $\overline{\text{NARC}} \geq 1$ |

Graph model

The final graph has between one and two arc constraints: it has two arcs when the default value DEFAULT occurs also in the table TABLE; otherwise it has only one arc.

Parts (A) and (B) of Figure 5.332 respectively show the initial and final graph associated with the **Example** slot. Since we use the $\overline{\text{NARC}}$ graph property the arcs of the final graph are outline with thick lines.

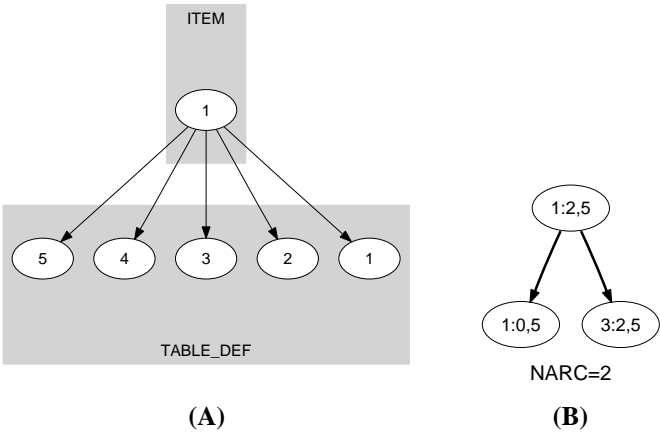


Figure 5.332: Initial and final graph of the *element_sparse* constraint

Automaton

Figure 5.333 depicts the automaton associated with the `element_sparse` constraint. Let `INDEX` and `VALUE` respectively be the index and the value attributes of the unique item of the `ITEM` collection. Let `INDEXi` and `VALUEi` respectively be the index and the value attributes of the i^{th} item of the `TABLE` collection. To each quintuple $(\text{INDEX}, \text{VALUE}, \text{DEFAULT}, \text{INDEX}_i, \text{VALUE}_i)$ corresponds a signature variable S_i as well as the following signature constraint:

$$\begin{cases} (\text{INDEX} \neq \text{INDEX}_i \wedge \text{VALUE} \neq \text{DEFAULT}) & \Leftrightarrow S_i = 0 \wedge \\ (\text{INDEX} = \text{INDEX}_i \wedge \text{VALUE} = \text{VALUE}_i) & \Leftrightarrow S_i = 1 \wedge . \\ (\text{INDEX} \neq \text{INDEX}_i \wedge \text{VALUE} = \text{DEFAULT}) & \Leftrightarrow S_i = 2 \end{cases}$$

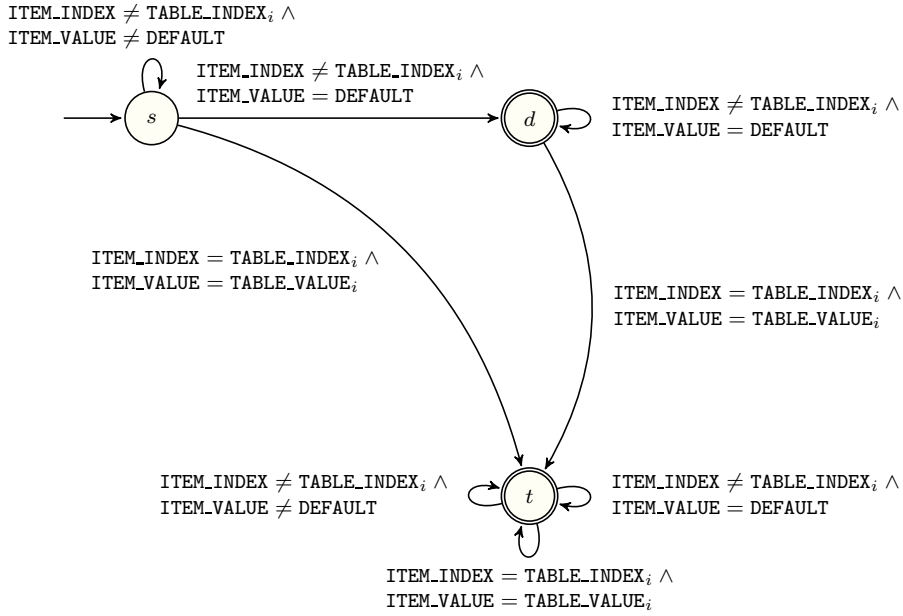


Figure 5.333: Automaton of the `element_sparse` constraint

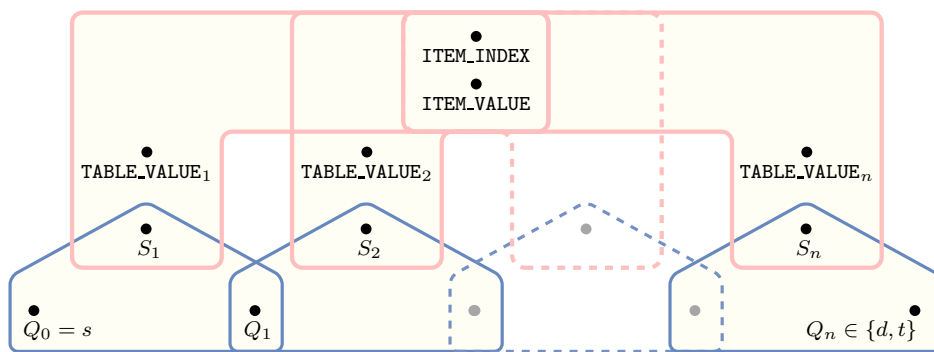


Figure 5.334: Hypergraph of the reformulation corresponding to the automaton of the `element_sparse` constraint

20030820

1171