

5.374 stage_element

	DESCRIPTION	LINKS	GRAPH	AUTOMATON
Origin	Choco, derived from <code>element</code> .			
Constraint	<code>stage_element</code> (ITEM, TABLE)			
Usual name	<code>stage_elt</code>			
Synonym	<code>stage_elem</code> .			
Arguments	ITEM : <code>collection</code> (index-dvar, value-dvar) TABLE : <code>collection</code> (low-int, up-int, value-int)			
Restrictions	<code>required</code> (ITEM, [index, value]) $ \text{ITEM} = 1$ $ \text{TABLE} > 0$ <code>required</code> (TABLE, [low, up, value]) $\text{TABLE.low} \leq \text{TABLE.up}$ <code>increasing_seq</code> (TABLE, [low])			
Purpose	<p>Let low_i, up_i and value_i respectively denote the values of the low, up and value attributes of the i^{th} item of the TABLE collection. First we have that: $\text{low}_i \leq \text{up}_i$ and $\text{up}_i + 1 = \text{low}_{i+1}$.</p> <p>Second, the <code>stage_element</code> constraint forces the following equivalence: $\text{low}_i \leq \text{ITEM.index} \wedge \text{ITEM.index} \leq \text{up}_i \Leftrightarrow \text{ITEM.value} = \text{value}_i$.</p>			
Example	$\left(\begin{array}{l} \langle \text{index} - 5 \text{ value} - 6 \rangle, \\ \left\langle \begin{array}{lll} \text{low} - 3 & \text{up} - 7 & \text{value} - 6, \\ \text{low} - 8 & \text{up} - 8 & \text{value} - 8, \\ \text{low} - 9 & \text{up} - 14 & \text{value} - 2, \\ \text{low} - 15 & \text{up} - 19 & \text{value} - 9 \end{array} \right\rangle \end{array} \right)$			
	<p>Figure 5.729 depicts the function associated with the items of the TABLE collection. The <code>stage_element</code> constraint holds since:</p> <ul style="list-style-type: none"> • The value of <code>ITEM[1].index</code> is located between the values of the <code>low</code> and <code>up</code> attributes of the first item of the TABLE collection (i.e., $5 \in [3, 7]$). • The value of <code>ITEM[1].value</code> corresponds to the <code>value</code> attribute of the first item of the TABLE collection (i.e., 6). 			
Typical	$ \text{TABLE} > 1$ <code>range</code> (TABLE.value) > 1 $\text{TABLE.low} < \text{TABLE.up}$			

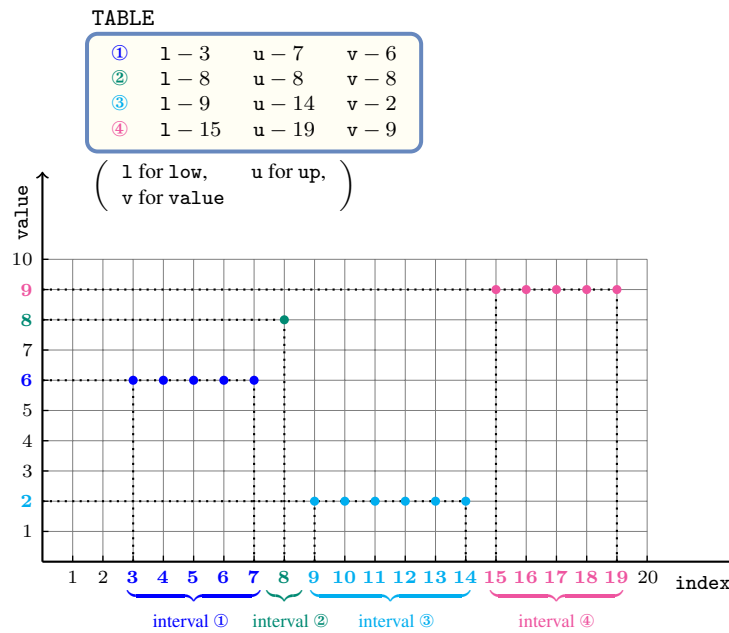


Figure 5.729: Function defined on four intervals ①, ②, ③ and ④ associated with the TABLE collection of the **Example** slot for linking the index and value attributes of the ITEM collection

Symmetry

All occurrences of two distinct values in ITEM.value or TABLE.value can be [swapped](#); all occurrences of a value in ITEM.value or TABLE.value can be [renamed](#) to any unused value.

Arg. properties

- [Functional dependency](#): ITEM.value determined by ITEM.index and TABLE.
- [Suffix-extensible](#) wrt. TABLE.

See also

[common keyword](#): [elem](#), [element](#) (*data constraint*).

Keywords

[characteristic of a constraint](#): [automaton](#), [automaton without counters](#), [reified automaton constraint](#).

[constraint arguments](#): [binary constraint](#), [pure functional dependency](#).

[constraint network structure](#): [centered cyclic\(2\) constraint network\(1\)](#).

[constraint type](#): [data constraint](#).

[filtering](#): [arc-consistency](#).

[modelling](#): [table](#), [functional dependency](#).

Arc input(s)	TABLE
Arc generator	$\text{PATH} \mapsto \text{collection}(\text{table1}, \text{table2})$
Arc arity	2
Arc constraint(s)	<ul style="list-style-type: none"> • $\text{table1.low} \leq \text{table1.up}$ • $\text{table1.up} + 1 = \text{table2.low}$ • $\text{table2.low} \leq \text{table2.up}$
Graph property(ies)	$\text{NARC} = \text{TABLE} - 1$
Arc input(s)	ITEM TABLE
Arc generator	$\text{PRODUCT} \mapsto \text{collection}(\text{item}, \text{table})$
Arc arity	2
Arc constraint(s)	<ul style="list-style-type: none"> • $\text{item.index} \geq \text{table.low}$ • $\text{item.index} \leq \text{table.up}$ • $\text{item.value} = \text{table.value}$
Graph property(ies)	$\text{NARC} = 1$

Graph model

The first graph constraint models the restrictions on the low and up attributes of the TABLE collection, while the second graph constraint is similar to the one used for defining the `element` constraint.

Parts (A) and (B) of Figure 5.730 respectively show the initial and final graph associated with the second graph constraint of the **Example** slot. Since we use the NARC graph property, the unique arc of the final graph is stressed in bold.

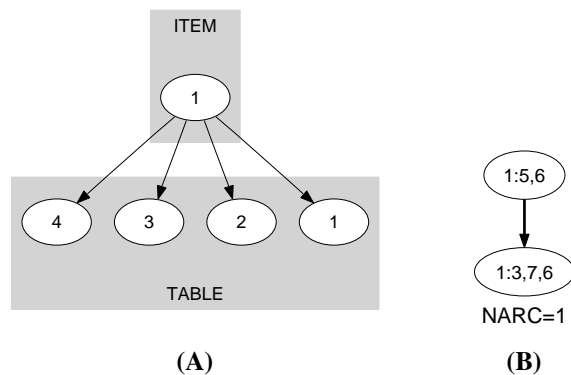


Figure 5.730: Initial and final graph of the `stage_element` constraint

Automaton

Figure 5.731 depicts the automaton associated with the `stage_element` constraint. Let `INDEX` and `VALUE` respectively be the index and the value attributes of the unique item of the `ITEM` collection. Let `LOWi`, `UPi` and `VALUEi` respectively be the low, the up and the value attributes of the *i*th item of the `TABLE` collection. To each quintuple $(INDEX, VALUE, LOW_i, UP_i, VALUE_i)$ corresponds a 0-1 signature variable S_i as well as the following signature constraint: $((LOW_i \leq INDEX) \wedge (INDEX \leq UP_i) \wedge (VALUE = VALUE_i)) \Leftrightarrow S_i$.

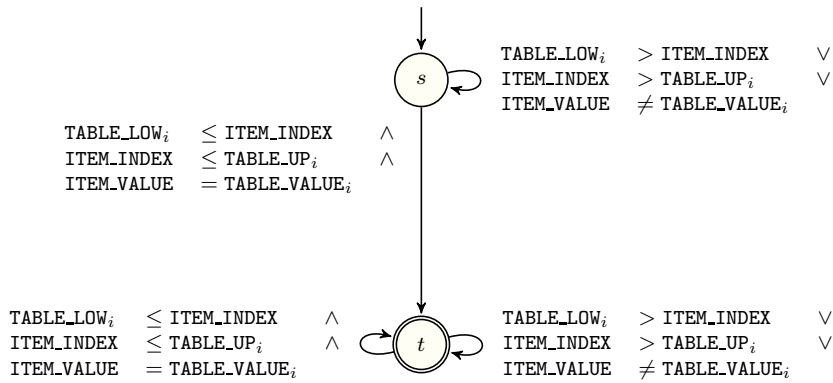


Figure 5.731: Automaton of the `stage_element` constraint

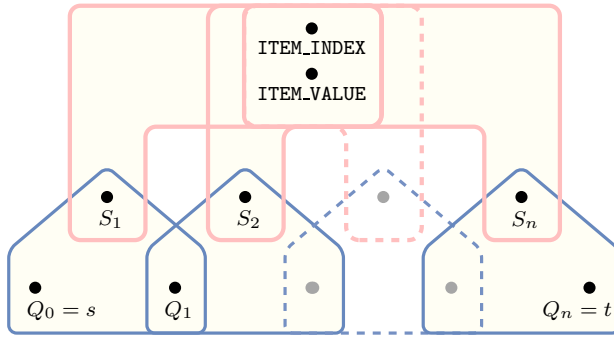


Figure 5.732: Hypergraph of the reformulation corresponding to the automaton of the `stage_element` constraint