## 5.195 int_value_precede

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>LINKS</th>
<th>AUTOMATON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>[258]</td>
<td></td>
</tr>
</tbody>
</table>

### Constraint

int_value_precede(S, T, VARIABLES)

### Synonyms

precede, precedence, value_precede.

### Arguments

| S    | : int |
| T    | : int |
| VARIABLES | : collection(var−dvar) |

### Restrictions

\[ S \neq T \]

required(VARIABLES, var)

### Purpose

If value T occurs in the collection of variables VARIABLES then its first occurrence should be preceded by an occurrence of value S.

### Example

\( (0, 1, (4, 0, 6, 1, 0)) \)

The int_value_precede constraint holds since the first occurrence of value 0 precedes the first occurrence of value 1.

### Typical

\[ S < T \]

\[ |\text{VARIABLES}| > 1 \]

\[ \text{atleast}(1, \text{VARIABLES}, S) \]

\[ \text{atleast}(1, \text{VARIABLES}, T) \]

### Symmetries

- An occurrence of a value of VARIABLES.var that is different from S and T can be replaced by any other value that is also different from S and T.
- All occurrences of values S and T can be swapped in S, T and VARIABLES.var.

### Arg. properties

- Suffix-contractible wrt. VARIABLES.
- Aggregate: S(id), T(id), VARIABLES(union).

### Algorithm

A filtering algorithm for maintaining value precedence is presented in [258]. Its complexity is linear to the number of variables of the collection VARIABLES.

### Systems

precede in Gecode, value_precede in MiniZinc.

### See also

generalisation: int_value_precede_chain (sequence of 2 values replaced by sequence of at least 2 values), set_value_precede (sequence of domain variables replaced by sequence of set variables).
Keywords

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

constraint network structure: Berge-acyclic constraint network.

constraint type: order constraint.

filtering: arc-consistency.

symmetry: symmetry, indistinguishable values, value precedence.
Automaton

Figure 5.435 depicts the automaton associated with the `int_value_precede` constraint. Let \( \text{VAR}_i \) be the \( i^{th} \) variable of the \( \text{VARIABLES} \) collection. To each triple \( (S, T, \text{VAR}_i) \) corresponds a signature variable \( S_i \) as well as the following signature constraint: 

\[
\begin{align*}
\text{VAR}_i = S & \iff S_i = 1 \\
\text{VAR}_i = T & \iff S_i = 2 \\
\text{VAR}_i \neq S \land \text{VAR}_i \neq T & \iff S_i = 3 
\end{align*}
\]

Figure 5.435: Automaton of the `int_value_precede` constraint (state \( s \) means that value \( S \) was not yet encountered, while state \( t \) means that value \( S \) was already encountered)

Figure 5.436: Hypergraph of the reformulation corresponding to the automaton of the `int_value_precede` constraint