

5.277 roots

	DESCRIPTION	LINKS	GRAPH
Origin	[57]		
Constraint	<code>roots(S, T, VARIABLES)</code>		
Arguments	S : <code>svar</code> T : <code>svar</code> VARIABLES : <code>collection(var-dvar)</code>		
Restrictions	$S \leq \text{VARIABLES} $ <code>required(VARIABLES, var)</code>		
Purpose	S is the set of indices of the variables in the collection VARIABLES taking their values in T; $S = \{i \mid \text{VARIABLES}[i].\text{var} \in T\}$. Positions are numbered from 1.		
Example	$\left(\begin{array}{l} \{2, 4, 5\}, \\ \{2, 3, 8\}, \\ \langle 1, 3, 1, 2, 3 \rangle \end{array} \right)$ <p>The <code>roots</code> constraint holds since values 2 and 3 in T occur in the collection $\langle 1, 3, 1, 2, 3 \rangle$ only at positions $S = \{2, 4, 5\}$. The value $8 \in T$ does not occur within the collection $\langle 1, 3, 1, 2, 3 \rangle$.</p>		
Usage	<p>Bessière et al. showed [57] that many counting and occurrence constraints can be specified with two <i>global primitives</i>: <code>roots</code> and <code>range</code>. For instance, the <code>count</code> constraint can be decomposed into one <code>roots</code> constraint: <code>count(VAL, VARS, OP, NVAR)</code> iff <code>roots(S, {VAL}, VARS) \wedge S OP NVAR</code>.</p> <p><code>roots</code> does not count but collects the set of variables using particular values. It provides then a way of channeling. <code>roots</code> generalizes, for instance, the <code>link_set_to_booleans</code> constraint, <code>link_set_to_booleans(S, BOOLEANS)</code> iff <code>roots(S, {1}, BOOLEANS.bool)</code>, or may be used instead of the domain constraint.</p> <p>Other examples of reformulations are given in [61].</p>		
Algorithm	<p>In [60], Bessière <i>et al.</i> shows that enforcing hybrid-consistency on <code>roots</code> is NP-hard. They consider the decomposition of <code>roots</code> into a network of ternary constraints: $\forall i, i \in S \Rightarrow \text{VARIABLES}[i].\text{var} \in T$ and $\text{VARIABLES}[i].\text{var} \Rightarrow T \wedge i \in S$. Enforcing bound consistency on the decomposition achieves bound consistency on <code>roots</code>. Enforcing hybrid consistency on the decomposition achieves at least bound consistency on <code>roots</code>, until hybrid consistency in some special cases:</p> <ul style="list-style-type: none"> • $\text{dom}(\text{VARIABLES}[i].\text{var}) \subset \underline{T}, \forall i \in \underline{S}$, • $\text{dom}(\text{VARIABLES}[i].\text{var}) \cap \overline{T} = \emptyset, \forall i \notin \overline{S}$, • VARIABLES are ground, 		

- T is ground.

Enforcing hybrid consistency on the decomposition can be done in $O(nd)$ with $n = |\text{VARIABLES}|$ and d the maximum domain size of $\text{VARIABLES}[i].\text{var}$ and T.

See also

common keyword: `link_set_to_booleans` (*constraint involving set variables*).

related: `among` (*can be expressed with `roots`*), `assign_and_nvalues` (*can be expressed with `roots` and `range`*), `atleast`, `atmost` (*can be expressed with `roots`*), `common` (*can be expressed with `roots` and `range`*), `count` (*can be expressed with `roots`*), `domain_constraint`, `global_cardinality`, `global_contiguity` (*can be expressed with `roots`*), `symmetric_alldifferent`, `uses` (*can be expressed with `roots` and `range`*).

Keywords

characteristic of a constraint: disequality.

constraint arguments: constraint involving set variables.

constraint type: counting constraint, value constraint, decomposition.

filtering: hybrid-consistency.

Derived Collection

$$\text{col}(\text{SETS} - \text{collection}(s - \text{svar}, t - \text{svar}), [\text{item}(s - S, t - T)])$$

Arc input(s)

SETS VARIABLES

Arc generator*PRODUCT* \mapsto `collection(sets, variables)`**Arc arity**

2

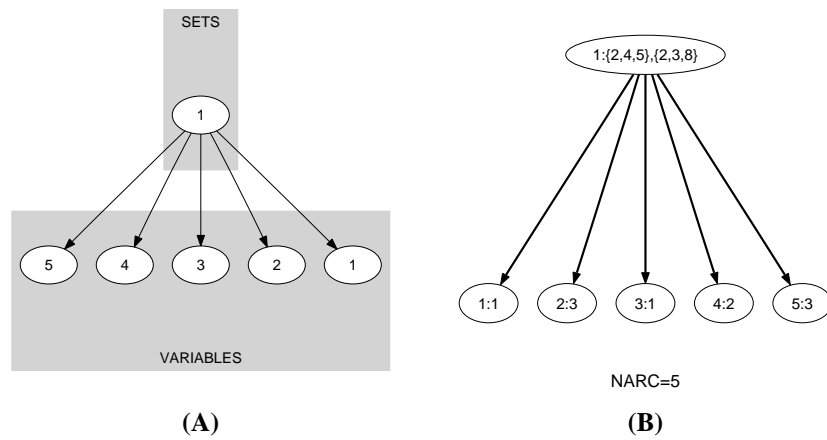
Arc constraint(s)`in_set(variables.key, sets.s) \Leftrightarrow in_set(variables.var, sets.t)`**Graph property(ies)****NARC** = |VARIABLES|

Figure 5.506: Initial and final graph of the roots constraint

Graph model

20070620

1471