5.404 tree_range

DESCRIPTION LINKS GRAPH

Origin

Derived from tree.

Constraint

tree_range(NTREES, R, NODES)

Arguments

NTREES	:	dvar
R	:	dvar
NODES	:	collection(index−int, succ−dvar)

Restrictions

$\mathtt{NTREES} \geq 0$
$\mathtt{R} \geq 0$
$\mathtt{R} < |\mathtt{NODES}|$
$|\mathtt{NODES}| > 0$
required(NODES, [index, succ])
$\mathtt{NODES.index} \geq 1$
$\mathtt{NODES.index} \leq |\mathtt{NODES}|$
distinct(NODES, index)
$\mathtt{NODES.succ} \geq 1$
$\mathtt{NODES.succ} \leq |\mathtt{NODES}|$

Purpose

Cover the digraph G described by the NODES collection with NTREES trees in such a way that each vertex of G belongs to one distinct tree. R is the difference between the longest and the shortest paths (from a leaf to a root) of the final graph.

Example

$$\left(2, 1, \left\langle \begin{array}{ll} \mathtt{index}-1 & \mathtt{succ}-1, \\ \mathtt{index}-2 & \mathtt{succ}-5, \\ \mathtt{index}-3 & \mathtt{succ}-5, \\ \mathtt{index}-4 & \mathtt{succ}-7, \\ \mathtt{index}-5 & \mathtt{succ}-1, \\ \mathtt{index}-6 & \mathtt{succ}-1, \\ \mathtt{index}-7 & \mathtt{succ}-7, \\ \mathtt{index}-8 & \mathtt{succ}-5 \end{array} \right\rangle \right)$$

The tree_range constraint holds since the graph associated with the items of the NODES collection corresponds to two trees (i.e., NTREES $= 2$): each tree respectively involves the vertices $\{1, 2, 3, 5, 6, 8\}$ and $\{4, 7\}$. Furthermore $\mathtt{R} = 1$ is set to the difference between the longest path (for instance $2 \rightarrow 5 \rightarrow 1$) and the shortest path (for instance $4 \rightarrow 7$) from a leaf to a root. Figure 5.766 provides the two trees associated with the example.

Typical

$\mathtt{NTREES} < |\mathtt{NODES}|$
$|\mathtt{NODES}| > 2$

Symmetry

Items of NODES are permutable.

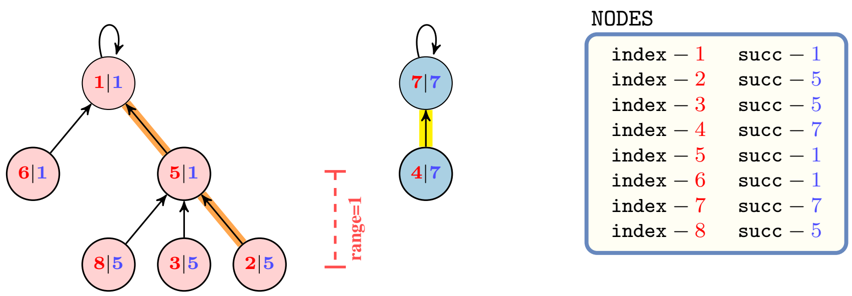


Figure 5.766: The two trees corresponding to the **Example** slot; each vertex contains the information index|succ where succ is the index of its father in the tree (by convention the father of the root is the root itself); the longest and shortest paths from a leaf to a root are respectively shown by thick orange and yellow line segments and have a length of 2 and 1; consequently the range is equal to 1.

Arg. properties

- Functional dependency: NTREES determined by NODES.
- Functional dependency: R determined by NODES.

Reformulation

By introducing a *distance variable* D_i, an *occurrence variable* O_i and a *leave variable* L_i $(1 \leq i \leq |\texttt{NODES}|)$ for each item i of the NODES collection, where:

- D_i represents the number of vertices from i to the root of the corresponding tree,
- O_i gives the number of occurrences of value i within variables $\texttt{NODES}[1].\texttt{succ}, \texttt{NODES}[2].\texttt{succ}, \ldots, \texttt{NODES}[n].\texttt{succ}$,
- L_i is set to 1 if item i corresponds to a leave (i.e., $O_i > 0$) and 0 otherwise,

the tree_range(NTREES, R, NODES) constraint can be expressed in term of a conjunction of one tree constraint, |NODES| element constraints, |NODES| linear constraints, one global_cardinality constraint, |NODES| reified constraints, one open_minimum, one maximum and one linear constraint, where:

- The tree constraint models the fact that we have a forest of NTREES trees.
- Each element constraint provides the link between the attribute succ of the i-th item and the distance variable $D_{\texttt{NODES}[i].\texttt{succ}}$ associated with item $\texttt{NODES}[i].\texttt{succ}$.
- Each linear constraint associated with the i-th item states that the difference between the distance variable D_i and the distance variable $D_{\texttt{NODES}[i].\texttt{succ}}$ is equal to 1.
- The global_cardinality constraint provides the number of occurrences O_i of value i $(1 \leq i \leq |\texttt{NODES}|)$ within variables $\texttt{NODES}[1].\texttt{succ}, \texttt{NODES}[2].\texttt{succ}, \ldots, \texttt{NODES}[|\texttt{NODES}|].\texttt{succ}$. Note that, when O_i is equal to 0, the corresponding i-th item is a leave of one of the NTREES trees.
- Each reified constraint of the form $L_i \Leftrightarrow O_i > 0$ makes the link between the i-th occurrence variable O_i and the i-th leave variable L_i.
- The open_minimum constraint computes the minimum distance MIN from the leaves to the corresponding roots. The leave variable L_i is used in order to select only the distance variables corresponding to leaves.

- The `maximum` constraint computes the maximum distance `MAX` from the vertices to the roots. Since the maximum is achieved by a leave we do not need to focus just on the leaves as it was the case for the minimum distance `MIN`.
- The linear constraint $\texttt{MAX} - \texttt{MIN} = \texttt{R}$ links together argument `R` to the minimum and maximum distances.

With respect to the **Example** slot we get the following conjunction of constraints:

$$
\begin{array}{l}
\texttt{tree}(2, \langle \texttt{index}-1\ \texttt{succ}-1,\ \texttt{index}-2\ \texttt{succ}-5,\\
\qquad\quad \texttt{index}-3\ \texttt{succ}-5,\ \texttt{index}-4\ \texttt{succ}-7,\\
\qquad\quad \texttt{index}-5\ \texttt{succ}-1,\ \texttt{index}-6\ \texttt{succ}-1,\\
\qquad\quad \texttt{index}-7\ \texttt{succ}-7,\ \texttt{index}-8\ \texttt{succ}-5\rangle),\\
\texttt{domain}(\langle D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8\rangle, 0, 8),\\
DS_1 \in [0,8],\ \texttt{element}(1, \langle 0, D_2, D_3, D_4, D_5, D_6, D_7, D_8\rangle, DS_1),\ D_1 - 0 = 1,\\
DS_2 \in [0,8],\ \texttt{element}(5, \langle 1, 0, D_3, D_4, D_5, D_6, D_7, D_8\rangle, DS_2),\ D_2 - D_5 = 1,\\
DS_3 \in [0,8],\ \texttt{element}(5, \langle 1, D_2, 0, D_4, D_5, D_6, D_7, D_8\rangle, DS_3),\ D_3 - D_5 = 1,\\
DS_4 \in [0,8],\ \texttt{element}(7, \langle 1, D_2, D_3, 0, D_5, D_6, D_7, D_8\rangle, DS_4),\ D_4 - D_7 = 1,\\
DS_5 \in [0,8],\ \texttt{element}(1, \langle 1, D_2, D_3, D_4, 0, D_6, D_7, D_8\rangle, DS_5),\ D_5 - 1 = 1,\\
DS_6 \in [0,8],\ \texttt{element}(1, \langle 1, 3, 3, D_4, 2, 0, D_7, D_8\rangle, DS_6),\ D_6 - 1 = 1,\\
DS_7 \in [0,8],\ \texttt{element}(7, \langle 1, 3, 3, D_4, 2, 2, 0, D_8\rangle, DS_7),\ D_7 - 0 = 1,\\
DS_8 \in [0,8],\ \texttt{element}(5, \langle 1, 3, 3, 2, 2, 2, 1, 0\rangle, DS_8),\ D_8 - 2 = 1,\\
\texttt{global\_cardinality}(\langle 1,5,5,7,1,1,7,5\rangle,\ \langle \texttt{val}-1\ \texttt{noccurrence}-3,\\
\qquad \texttt{val}-2\ \texttt{noccurrence}-0,\\
\qquad \texttt{val}-3\ \texttt{noccurrence}-0,\\
\qquad \texttt{val}-4\ \texttt{noccurrence}-0,\\
\qquad \texttt{val}-5\ \texttt{noccurrence}-3,\\
\qquad \texttt{val}-6\ \texttt{noccurrence}-0,\\
\qquad \texttt{val}-7\ \texttt{noccurrence}-2,\\
\qquad \texttt{val}-8\ \texttt{noccurrence}-0\rangle),\\
1 \Leftrightarrow 3 > 0, 0 \Leftrightarrow 0 > 0, 0 \Leftrightarrow 0 > 0, 0 \Leftrightarrow 0 > 0,\\
1 \Leftrightarrow 3 > 0, 0 \Leftrightarrow 0 > 0, 1 \Leftrightarrow 2 > 0, 0 \Leftrightarrow 0 > 0,\\
\texttt{open\_minimum}(\texttt{MIN}, \langle \texttt{var}-3\ \texttt{bool}-1, \texttt{var}-0\ \texttt{bool}-0,\\
\qquad \texttt{var}-0\ \texttt{bool}-0, \texttt{var}-0\ \texttt{bool}-0,\\
\qquad \texttt{var}-3\ \texttt{bool}-1, \texttt{var}-0\ \texttt{bool}-0,\\
\qquad \texttt{var}-2\ \texttt{bool}-1, \texttt{var}-0\ \texttt{bool}-0\rangle),\\
\texttt{maximum}(\texttt{MAX}, \langle 1,3,3,2,2,2,1,3\rangle),\\
\texttt{MAX} - \texttt{MIN} = \texttt{R} = 1.
\end{array}
$$

See also

related: `balance` *(balanced tree versus balanced assignment).*

root concept: `tree`.

used in reformulation: `domain`, `element`, `global_cardinality`, `maximum`, `open_minimum`, `tree`.

Keywords

constraint type: graph constraint, graph partitioning constraint.

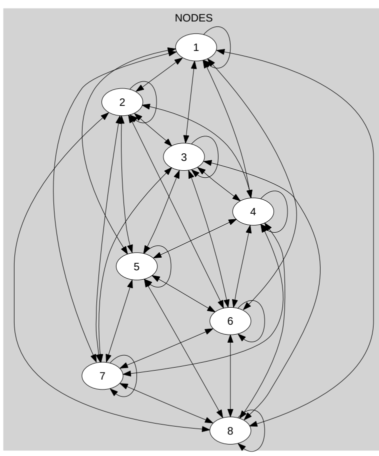
final graph structure: connected component, tree.

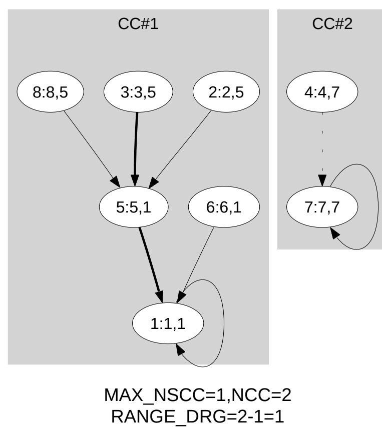
modelling: balanced tree, functional dependency.

Arc input(s)	NODES
Arc generator	*CLIQUE*↦collection(nodes1, nodes2)
Arc arity	2
Arc constraint(s)	nodes1.succ = nodes2.index
Graph property(ies)	• **MAX_NSCC** ≤ 1 • **NCC** = NTREES • **RANGE_DRG** = R

Graph model

Parts (A) and (B) of Figure 5.767 respectively show the initial and final graph associated with the **Example** slot. Since we use the RANGE_DRG graph property, we respectively display the longest and shortest paths of the final graph with a bold and a dash line.





(A) **(B)**

Figure 5.767: Initial and final graph of the tree_range constraint