

5.239 map

	DESCRIPTION	LINKS	GRAPH
Origin	Inspired by [376]		
Constraint	map(NBCYCLE, NBTREE, NODES)		
Arguments	NBCYCLE : dvar NBTREE : dvar NODES : collection(index-int, succ-dvar)		
Restrictions	NBCYCLE ≥ 0 NBTREE ≥ 0 required(NODES, [index, succ]) NODES.index ≥ 1 NODES.index ≤ NODES distinct(NODES, index) NODES.succ ≥ 1 NODES.succ ≤ NODES		
Purpose	Number of trees and number of cycles of a map. We take the description of a map from [376, page 459]: <i>“Every map decomposes into a set of connected components, also called connected maps. Each component consists of the set of all points that wind up on the same cycle, with each point on the cycle attached to a tree of all points that enter the cycle at that point.”</i>		
Example	$2, 3, \left(\begin{array}{l} \text{index} - 1 \quad \text{succ} - 5, \\ \text{index} - 2 \quad \text{succ} - 9, \\ \text{index} - 3 \quad \text{succ} - 8, \\ \text{index} - 4 \quad \text{succ} - 2, \\ \text{index} - 5 \quad \text{succ} - 9, \\ \text{index} - 6 \quad \text{succ} - 2, \\ \text{index} - 7 \quad \text{succ} - 9, \\ \text{index} - 8 \quad \text{succ} - 8, \\ \text{index} - 9 \quad \text{succ} - 1 \end{array} \right)$		
Typical	NBCYCLE > 0 NBTREE > 0 NBCYCLE < NODES NBCYCLE < NBTREE NODES > 2		

The map constraint holds since, as shown by part (B) of Figure 5.510, the graph corresponding to the NODES collection is a map containing NBCYCLE = 2 cycles (i.e., a first cycle involving vertices 1, 5 and 9 and a second cycle involving vertex 8) and 3 trees (i.e., two trees respectively involving vertices 7 and 4, 6, 2 and attached to the first cycle, and one tree mentioning vertex 3 linked to the second cycle.)

Symmetry

Items of NODES are [permutable](#).

Arg. properties

- [Functional dependency](#): NBCYCLE determined by NODES.
- [Functional dependency](#): NBTREE determined by NODES.

See also

common keyword: [cycle](#), [graph_crossing](#), [tree](#) (*graph partitioning constraint*).

Keywords

constraint arguments: pure functional dependency.

constraint type: graph constraint, graph partitioning constraint.

filtering: DFS-bottleneck.

final graph structure: connected component.

modelling: functional dependency.

Arc input(s)	NODES
Arc generator	<i>CLIQUE</i> \mapsto collection(nodes1, nodes2)
Arc arity	2
Arc constraint(s)	nodes1.succ = nodes2.index
Graph property(ies)	<ul style="list-style-type: none"> • NCC= NBCYCLE • NTREE= NBTREE

Graph model

Note that, for the argument NBTREE of the map constraint, we consider a definition different from the one used for the argument NTREES of the `tree` constraint:

- In the map constraint the number of trees NBTREE is equal to the number of vertices of the final graph, which both do not belong to any circuit and have a successor that is located on a circuit. Therefore we count three trees in the context of the **Example** slot.
- In the `tree` constraint the number of trees NTREES is equal to the number of **connected components** of the final graph.

Parts (A) and (B) of Figure 5.510 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NCC** graph property, we display the two **connected components** of the final graph. Each of them corresponds to a connected map. The first connected map is made up from one circuit and two trees, while the second one consists of one circuit and one tree. Since we also use the **NTREE** graph property, we display with a double circle those vertices that do not belong to any circuit but for which at least one successor belongs to a circuit.

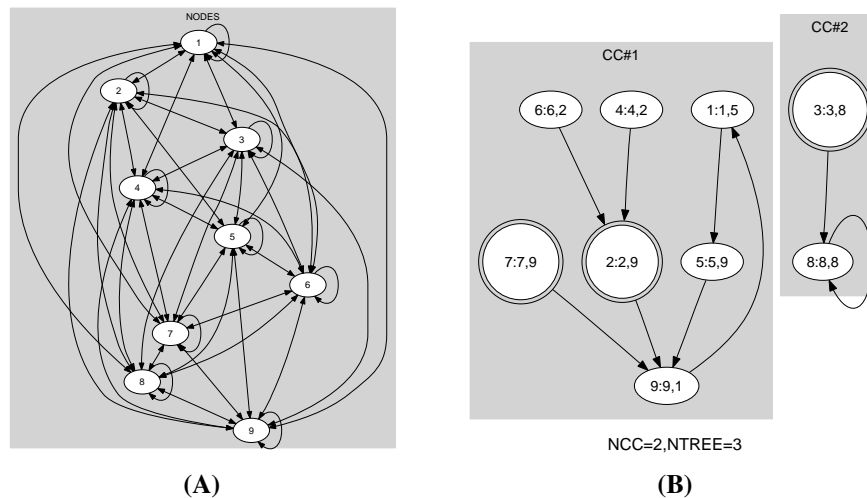


Figure 5.510: Initial and final graph of the map constraint

20000128

1603