5.102 cutset

DESCRIPTION LINKS GRAPH

Origin [156]

Constraint cutset(SIZE_CUTSET, NODES)

Arguments

SIZE_CUTSET	:	dvar
NODES	:	collection(index−int, succ−sint, bool−dvar)

Restrictions

SIZE_CUTSET ≥ 0
SIZE_CUTSET $\leq$ |NODES|
required(NODES, [index, succ, bool])
NODES.index ≥ 1
NODES.index $\leq$ |NODES|
distinct(NODES, index)
NODES.bool ≥ 0
NODES.bool ≤ 1

Purpose

Consider a digraph G with n vertices described by the NODES collection. Enforces that the subset of kept vertices of cardinality $n-$ SIZE_CUTSET and their corresponding arcs form a graph without circuit.

Example

$$\left(1, \left\langle \begin{array}{lll} \texttt{index}-1 & \texttt{succ}-\{2,3,4\} & \texttt{bool}-1, \\ \texttt{index}-2 & \texttt{succ}-\{3\} & \texttt{bool}-1, \\ \texttt{index}-3 & \texttt{succ}-\{4\} & \texttt{bool}-1, \\ \texttt{index}-4 & \texttt{succ}-\{1\} & \texttt{bool}-0 \end{array} \right\rangle \right)$$

The cutset constraint holds since the vertices of the NODES collection for which the bool attribute is set to 1 correspond to a graph without circuit and since exactly one (SIZE_CUTSET $= 1$) vertex has its bool attribute set to 0.

Typical

SIZE_CUTSET > 0
SIZE_CUTSET $\leq$ |NODES|
|NODES| > 1

Symmetry Items of NODES are permutable.

Usage The article [156] introducing the cutset constraint mentions applications from various areas such that deadlock breaking or program verification.

Remark The undirected version of the cutset constraint corresponds to the minimum feedback vertex set problem.

Algorithm The filtering algorithm presented in [156] uses graph reduction techniques inspired from Levy and Low [260] as well as from Lloyd, Soffa and Wang [264].

Keywords

application area: deadlock breaking, program verification.

constraint type: graph constraint.

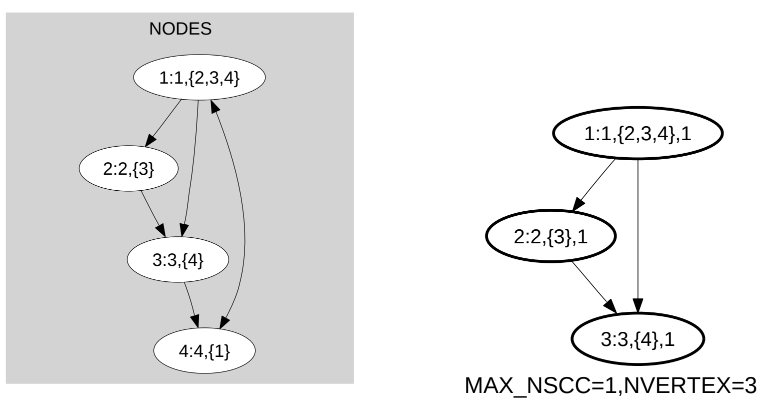
final graph structure: circuit, directed acyclic graph, acyclic, no loop.

problems: minimum feedback vertex set.

Arc input(s)	`NODES`
Arc generator	*CLIQUE*$\mapsto$`collection`(`nodes1`, `nodes2`)
Arc arity	2
Arc constraint(s)	• `in_set`(`nodes2.index`, `nodes1.succ`) • `nodes1.bool` $= 1$ • `nodes2.bool` $= 1$
Graph property(ies)	• **MAX_NSCC** ≤ 1 • **NVERTEX** $= \|$`NODES`$\| -$ `SIZE_CUTSET`
Graph class	• `ACYCLIC` • `NO_LOOP`

Graph model We use a set of integers for representing the successors of each vertex. Because of the arc constraint, all arcs such that the `bool` attribute of one extremity is equal to 0 are eliminated; Therefore all vertices for which the `bool` attribute is equal to 0 are also eliminated (since they will correspond to isolated vertices). The graph property **MAX_NSCC** ≤ 1 enforces the size of the largest strongly connected component to not exceed 1; Therefore, the final graph cannot contain any circuit.

Part (A) of Figure 5.227 shows the initial graph from which we have chosen to start. It is derived from the set associated with each vertex. Each set describes the potential values of the `succ` attribute of a given vertex. Part (B) of Figure 5.227 gives the final graph associated with the **Example** slot. Since we use the **NVERTEX** graph property, the vertices of the final graph are stressed in bold. The `cutset` constraint holds since the final graph does not contain any circuit and since the number of removed vertices `SIZE_CUTSET` is equal to 1.



(A) **(B)**

Figure 5.227: Initial and final graph of the `cutset` set constraint