5.400 temporal_path

DESCRIPTION LINKS GRAPH

Origin

ILOG

Constraint

`temporal_path(NPATH, NODES)`

Arguments

$$\begin{array}{lll} \texttt{NPATH} & : & \texttt{dvar} \\ \texttt{NODES} & : & \texttt{collection}\left(\begin{array}{l} \texttt{index}-\texttt{int}, \\ \texttt{succ}-\texttt{dvar}, \\ \texttt{start}-\texttt{dvar}, \\ \texttt{end}-\texttt{dvar} \end{array}\right) \end{array}$$

Restrictions

$\texttt{NPATH} \geq 1$
$\texttt{NPATH} \leq |\texttt{NODES}|$
$\texttt{required}(\texttt{NODES}, [\texttt{index}, \texttt{succ}, \texttt{start}, \texttt{end}])$
$|\texttt{NODES}| > 0$
$\texttt{NODES.index} \geq 1$
$\texttt{NODES.index} \leq |\texttt{NODES}|$
$\texttt{distinct}(\texttt{NODES}, \texttt{index})$
$\texttt{NODES.succ} \geq 1$
$\texttt{NODES.succ} \leq |\texttt{NODES}|$
$\texttt{NODES.start} \leq \texttt{NODES.end}$

Purpose

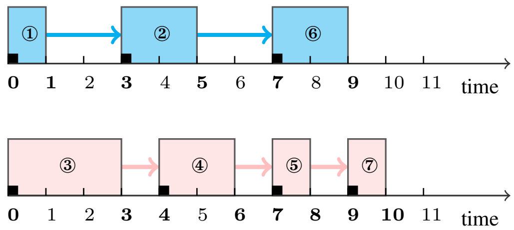
Let G be the digraph described by the `NODES` collection. Partition G with a set of disjoint paths such that each vertex of the graph belongs to a single path. In addition, for all pairs of consecutive vertices of a path we have a precedence constraint that enforces the end associated with the first vertex to be less than or equal to the start related to the second vertex.

Example

$$\left(2, \left\langle \begin{array}{llll} \texttt{index}-1 & \texttt{succ}-2 & \texttt{start}-0 & \texttt{end}-1, \\ \texttt{index}-2 & \texttt{succ}-6 & \texttt{start}-3 & \texttt{end}-5, \\ \texttt{index}-3 & \texttt{succ}-4 & \texttt{start}-0 & \texttt{end}-3, \\ \texttt{index}-4 & \texttt{succ}-5 & \texttt{start}-4 & \texttt{end}-6, \\ \texttt{index}-5 & \texttt{succ}-7 & \texttt{start}-7 & \texttt{end}-8, \\ \texttt{index}-6 & \texttt{succ}-6 & \texttt{start}-7 & \texttt{end}-9, \\ \texttt{index}-7 & \texttt{succ}-7 & \texttt{start}-9 & \texttt{end}-10 \end{array} \right\rangle \right)$$

The `temporal_path` constraint holds since:

- The items of the `NODES` collection represent the two ($\texttt{NPATH} = 2$) paths $1 \rightarrow 2 \rightarrow 6$ and $3 \rightarrow 4 \rightarrow 5 \rightarrow 7$.
- As illustrated by Figure 5.758, all precedences between adjacent vertices of a same path hold: each item i ($1 \leq i \leq 7$) of the `NODES` collection is represented by a rectangle starting and ending at instants $\texttt{NODES}[i]\texttt{.start}$ and $\texttt{NODES}[i]\texttt{.end}$; the number within each rectangle designates the index of the corresponding item within the `NODES` collection.



①	`index` − 1	`succ` − 2	`start` − 0	`end` − 1
②	`index` − 2	`succ` − 6	`start` − 3	`end` − 5
③	`index` − 3	`succ` − 4	`start` − 0	`end` − 3
④	`index` − 4	`succ` − 5	`start` − 4	`end` − 6
⑤	`index` − 5	`succ` − 7	`start` − 7	`end` − 8
⑥	`index` − 6	`succ` − 6	`start` − 7	`end` − 9
⑦	`index` − 7	`succ` − 7	`start` − 9	`end` − 10

NODES

Figure 5.758: The two paths of the **Example** slot represented as two sequences of tasks

Typical

NPATH $<$ |NODES|
|NODES| >1
NODES.start $<$ NODES.end

Symmetries

- Items of NODES are permutable.
- One and the same constant can be added to the start and end attributes of all items of NODES.

Arg. properties

Functional dependency: NPATH determined by NODES.

Remark

This constraint is related to the path constraint of **Ilog Solver**. It can also be directly expressed with the cycle [41] constraint of **CHIP** by using the *diff nodes* and the origin parameters. A generic model based on linear programming that handles paths, trees and cycles is presented in [244].

Reformulation

The temporal_path(NPATH, NODES) constraint can be expressed in term of a conjunction of one path constraint, |NODES| element constraints, and |NODES| inequalities constraints:

- We pass to the path constraint the number of path variable NPATH as well as the items of the NODES collection form which we remove the start and end attributes.
- To the i-th ($1 \leq i \leq$ |NODES|) item of the NODES collection, we create a variable $Start_{succ_i}$ and an element(NODES[i].succ, $\langle T_{i,1}, T_{i,2}, \ldots, T_{i,\text{NODES}}\rangle$, $Start_{succ_i}$) constraint, where $T_{i,j}$ = NODES[i].start if $i \neq j$ and $T_{i,i}$ = NODES[i].end otherwise.
- Finaly to the i-th ($1 \leq i \leq$ |NODES|) item of the NODES collection, we also create an inequality constraint NODES[i].end $\leq Start_{succ_i}$. Note that, since $T_{i,i}$ was initialised to NODES[i].end, the inequality NODES[i].end $\leq T_{i,j}$ holds when $i = j$.

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

$\texttt{path}(2, \langle \texttt{index} - 1\ \texttt{succ} - 2, \texttt{index} - 2\ \texttt{succ} - 6, \texttt{index} - 3\ \texttt{succ} - 4,$
$\texttt{index} - 4\ \texttt{succ} - 5, \texttt{index} - 5\ \texttt{succ} - 7, \texttt{index} - 6\ \texttt{succ} - 6,$
$\texttt{index} - 7\ \texttt{succ} - 7\rangle),$
$\texttt{element}(2, \langle 1, 3, 0, 4, 7, 7, 9\rangle, 3),$
$\texttt{element}(6, \langle 1, 5, 0, 4, 7, 7, 9\rangle, 7),$
$\texttt{element}(4, \langle 1, 5, 3, 4, 7, 7, 9\rangle, 4),$
$\texttt{element}(5, \langle 1, 5, 3, 6, 7, 7, 9\rangle, 7),$
$\texttt{element}(7, \langle 1, 5, 3, 6, 8, 7, 9\rangle, 9),$
$\texttt{element}(6, \langle 1, 5, 3, 6, 8, 9, 9\rangle, 9),$
$\texttt{element}(7, \langle 1, 5, 3, 6, 8, 9, 10\rangle, 10),$
$1 \leq 3, 5 \leq 7, 3 \leq 4, 6 \leq 7, 8 \leq 9, 9 \leq 9, 10 \leq 10.$

See also

common keyword: `path_from_to` *(path)*.

implies (items to collection): `atleast_nvector`.

specialisation: `path` *(time dimension removed)*.

Keywords

combinatorial object: path.

constraint type: graph constraint, graph partitioning constraint.

final graph structure: connected component.

modelling: sequence dependent set-up, functional dependency.

modelling exercises: sequence dependent set-up.

Arc input(s)	NODES
Arc generator	$CLIQUE \mapsto$ collection(nodes1, nodes2)
Arc arity	2
Arc constraint(s)	• nodes1.succ = nodes2.index • nodes1.succ = nodes1.index $\vee$ nodes1.end $\leq$ nodes2.start • nodes1.start $\leq$ nodes1.end • nodes2.start $\leq$ nodes2.end
Graph property(ies)	• **MAX_ID** ≤ 1 • **NCC** = NPATH • **NVERTEX** = |NODES|

Graph model The arc constraint is a conjunction of four conditions that respectively correspond to:

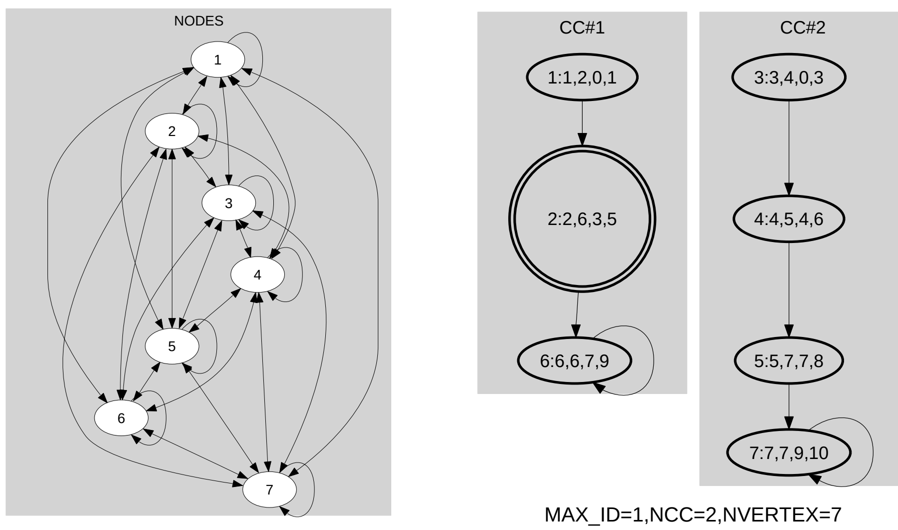
- A constraint that links the successor variable of a first vertex to the index attribute of a second vertex,
- A precedence constraint that applies on one vertex and its distinct successor,
- One precedence constraint between the start and the end of the vertex that corresponds to the departure of an arc,
- One precedence constraint between the start and the end of the vertex that corresponds to the arrival of an arc.

We use the following three graph properties in order to enforce the partitioning of the graph in distinct paths:

- The first property **MAX_ID** ≤ 1 enforces that each vertex has no more than one predecessor (**MAX_ID** does not consider loops),
- The second property **NCC** = NPATH ensures that we have the required number of paths,
- The third property **NVERTEX** = |NODES| enforces that, for each vertex, the start is not located after the end.

Parts (A) and (B) of Figure 5.759 respectively show the initial and final graph associated with the **Example** slot. Since we use the **MAX_ID**, the **NCC** and the **NVERTEX** graph properties we display the following information on the final graph:

- We show with a double circle a vertex that has the maximum number of predecessors.
- We show the two connected components corresponding to the two paths.
- We put in bold the vertices.



(A) **(B)**

Figure 5.759: Initial and final graph of the `temporal_path` constraint