5.294 open_alldifferent

DESCRIPTION LINKS GRAPH

Origin [427]

Constraint open_alldifferent(S, VARIABLES)

Synonyms open_alldiff, open_alldistinct, open_distinct.

Arguments

S	:	svar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{S} \geq 1$
$\texttt{S} \leq |\texttt{VARIABLES}|$
required(VARIABLES, var)

Purpose Let $\mathcal{V}$ be the variables of the collection VARIABLES for which the corresponding position belongs to the set S. Positions are numbered from 1. Enforce all variables of $\mathcal{V}$ to take distinct values.

Example $(\{2,3,4\}, \langle 9,1,9,3\rangle)$

The open_alldifferent constraint holds since the last three (i.e., $\texttt{S} = \{2,3,4\}$) values of the collection $\langle 9,1,9,3\rangle$ are distinct.

Typical $|\texttt{VARIABLES}| > 2$

Symmetry All occurrences of two distinct values of VARIABLES.var can be swapped; all occurrences of a value of VARIABLES.var can be renamed to any unused value.

Arg. properties

Suffix-contractible wrt. VARIABLES.

Usage In their article [427], W.-J. van Hoeve and J.-C. Régin motivate the open_alldifferent constraint by the following scheduling problem. Consider a set of activities (where each activity has a fixed duration 1 and a start variable) that can be processed on two factory lines such that all the activities that will be processed on a given line must be pairwise distinct. This can be modelled by using one open_alldifferent constraint for each line, involving all the start variables as well as a set variable whose final value specifies the set of activities assigned to that specific factory line.

Note that this can also be directly modelled by a single diffn constraint. This is done by introducing an assignment variable for each activity. The initial domain of each assignment variable consists of two values that respectively correspond to the two factory lines.

Algorithm A slight adaptation of the flow model that handles the original global_cardinality constraint [342] is described in [427]. The rightmost part of Figure 3.29 illustrates this flow model.

See also

common keyword: `size_max_seq_alldifferent`, `size_max_starting_seq_alldifferent` *(all different,disequality)*.

generalisation: `open_global_cardinality` *(control the number of occurrence of each active value*[13] *with a counter variable)*, `open_global_cardinality_low_up` *(control the number of occurrence of each active value with an interval)*.

hard version: `alldifferent`.

used in graph description: `in_set`.

Keywords

characteristic of a constraint: all different, disequality.

constraint arguments: constraint involving set variables.

constraint type: open constraint, soft constraint, value constraint.

filtering: flow.

[13] An *active value* corresponds to a value occuring at a position mentionned in the set `S`.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	• variables1.var = variables2.var • in_set(variables1.key, S) • in_set(variables2.key, S)
Graph property(ies)	MAX_NSCC ≤ 1
Graph class	ONE_SUCC

Graph model We generate a *clique* with an *equality* constraint between each pair of vertices (including a vertex and itself) and state that the size of the largest strongly connected component should not exceed one. Variables for which the corresponding position does not belong to the set S are removed from the final graph by the second and third conditions of the arc-constraint.

Parts (A) and (B) of Figure 5.611 respectively show the initial and final graph associated with the **Example** slot. Since we use the MAX_NSCC graph property we show one of the largest strongly connected components of the final graph. The open_alldifferent holds since all the strongly connected components have at most one vertex: a value is used at most once.

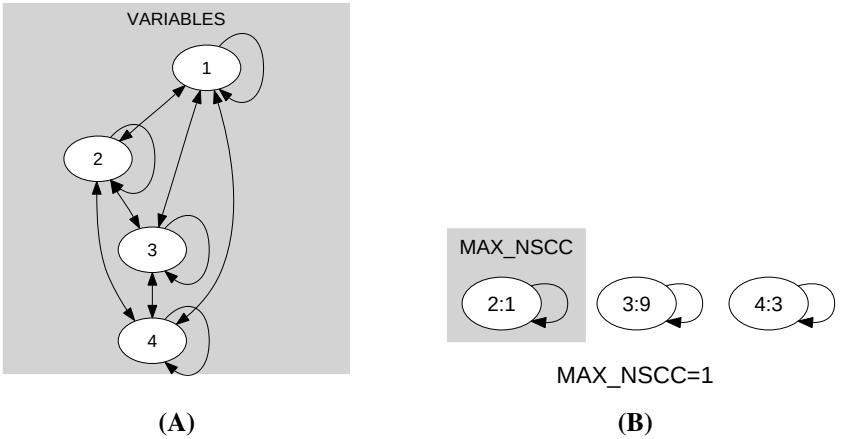

Figure 5.611: Initial and final graph of the open_alldifferent constraint