Regular Nurse

4.1.4 Regular

The `regular` constraint is used to enforce that a sequence of variables takes a value defined by a finite automaton. The usage of `regular` has the form

```
regular(array[int] of var int: x, int: Q, int: S,
        array[int,int] of int: d, int: q0, set of int: F)
```

It constrains that the sequence of values in array x (which must all be in the range $1..S$) is accepted by the DFA of Q states with input $1..S$ and transition function d (which maps $< 1..Q, 1..S >$ to $0..Q$) and initial state $q0$ (which must be in $1..Q$) and accepting states F (which all must be in $1..Q$). State 0 is reserved to be an always failing state.

Consider a nurse rostering problem. Each nurse is scheduled for each day as either: (d) on day shift, (n) on night shift, or (o) off. In each four day period a nurse must have at least one day off, and no nurse can be scheduled for 3 night shifts in a row. This can be encoded using the incomplete DFA

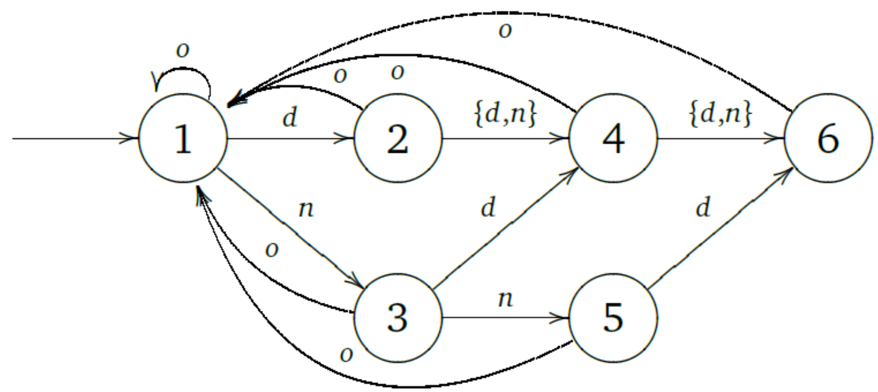
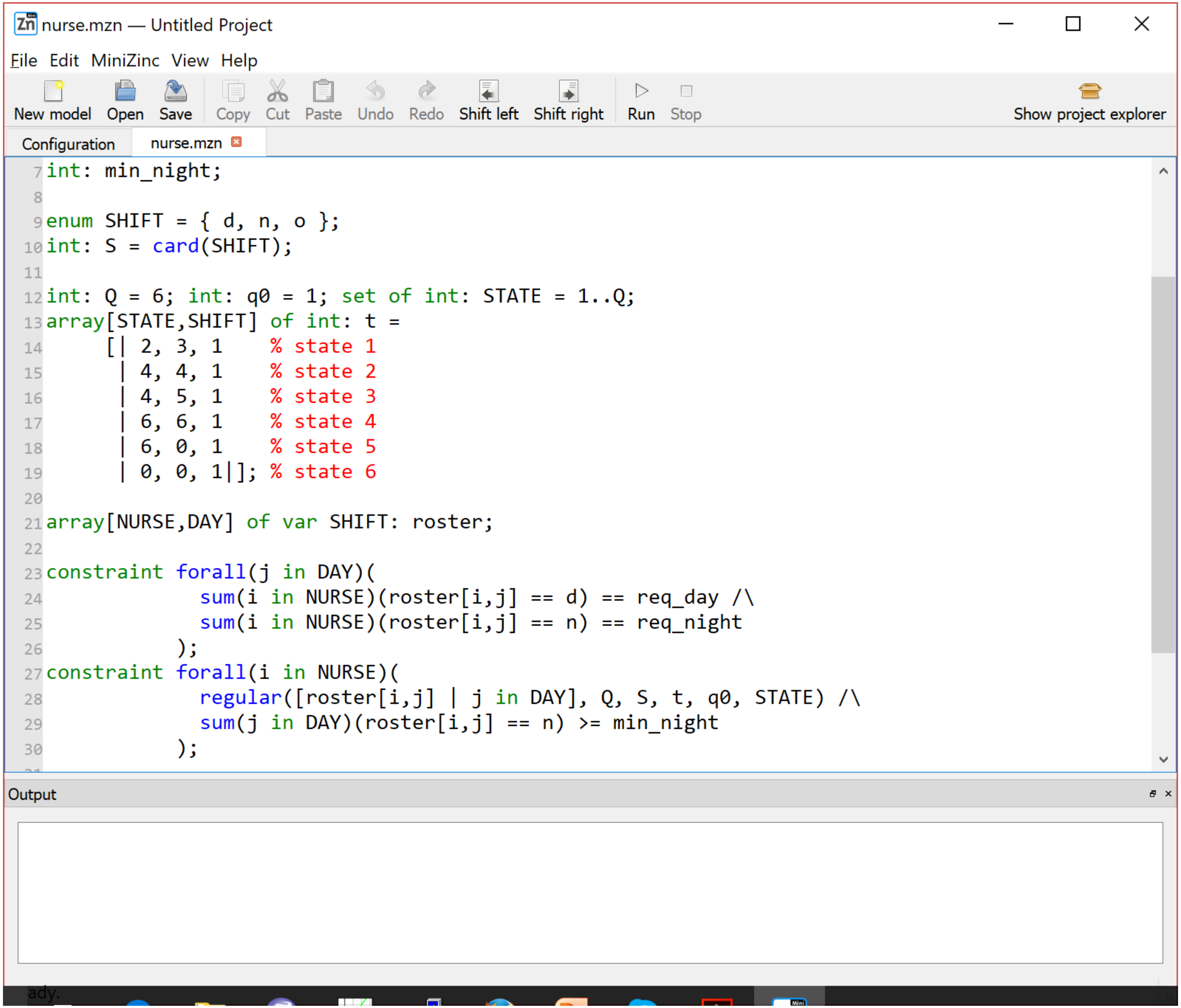


Figure 29: A DFA determining correct rosters.

shown in Figure 29. We can encode this DFA as having start state 1, final states 1 .. 6, and transition function

	d	n	o
1	2	3	1
2	4	4	1
3	4	5	1
4	6	6	1
5	6	0	1
6	0	0	1


nurse.mzn — Untitled Project
File Edit MiniZinc View Help
New model
Open
Save
Copy
Cut
Paste
Undo
Redo
Shift left
Shift right
Run
Stop
Show project explorer
Configuration
nurse.mzn
int: min_night;
enum SHIFT = { d, n, o };
int: S = card(SHIFT);
int: Q = 6; int: q0 = 1; set of int: STATE = 1..Q;
array[STATE,SHIFT] of int: t =
[| 2, 3, 1 % state 1
| 4, 4, 1 % state 2
| 4, 5, 1 % state 3
| 6, 6, 1 % state 4
| 6, 0, 1 % state 5
| 0, 0, 1|]; % state 6
array[NURSE,DAY] of var SHIFT: roster;
constraint forall(j in DAY)(
sum(i in NURSE)(roster[i,j] == d) == req_day /\
sum(i in NURSE)(roster[i,j] == n) == req_night
);
constraint forall(i in NURSE)(
regular([roster[i,j] | j in DAY], Q, S, t, q0, STATE) /\
sum(j in DAY)(roster[i,j] == n) >= min_night
);
Output

Windows PowerShell

```
PS C:\cpM\minizincCPM\nurse> mzn-gecode .\nurse.mzn .\nurse_07.dzn
d o o d n n o
d o n d d o n
d d o d n n o
o d d n o o n
o d n n o d d
n n d o d d d
n n d o d d d
----------
PS C:\cpM\minizincCPM\nurse> mzn-gecode .\nurse.mzn .\nurse_10.dzn
o o n d n o n n o o
d o d n n o d n n o
o d d n o n d o n n
d d d o o n d d o n
d d n o d d n o d d
n n o d d d o d d d
n n o d d d o d d d
----------
PS C:\cpM\minizincCPM\nurse> _
```

MiniZinc Documentation - Standard Library

Extensional constraints (table, regular etc.)

Index reveal all hide all

```
predicate regular(array [int] of var int: x,
                  int: Q,
                  int: S,
                  array [int,int] of int: d,
                  int: q0,
                  set of int: F)
```

The sequence of values in array **x** (which must all be in the range 1..**S**) is accepted by the DFA of **Q** states with input 1..**S** and transition function **d** (which maps (1..**Q**, 1..**S**) -> 0..**Q**)) and initial state **q0** (which must be in 1..**Q**) and accepting states **F** (which all must be in 1..**Q**). We reserve state 0 to be an always failing state.

```
predicate regular_nfa(array [int] of var int: x,
                      int: Q,
                      int: S,
                      array [int,int] of set of int: d,
                      int: q0,
                      set of int: F)
```

International Conference on Principles and Practice of Constraint Programming
CP 2004: Principles and Practice of Constraint Programming – CP 2004 pp 482-495 | Cite as

A Regular Language Membership Constraint for Finite Sequences of Variables

Authors | Authors and affiliations

Gilles Pesant

Part of the Lecture Notes in Computer Science book series (LNCS, volume 3258)

Abstract

This paper describes a global constraint on a fixed-length sequence of finite-domain variables requiring that the corresponding sequence of values taken by these variables belong to a given regular language, thereby generalizing some other known global constraints. We describe and analyze a filtering algorithm achieving generalized arc consistency for this constraint. Some comparative empirical results are also given.