Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

Toulouse, January 2, 2020

TouIST tutorial #4: Advanced generalized connectors

1 Prerequisites

You must have done all the previous tutorials. You must know the generalized connectors, know how to write compact nests of such connectors, and optimize your formulas so as not to generate the same constraint several times.

2 Goal of the tutorial

At the end of this tutorial, you will be able to define lists of index lists, and use them in generalized connectors in order to make your code even more compact.

3 The Map coloring problem

> The 4-color theorem, which was conjectured in 1852 by Francis Guthrie, states that one can color any geographic map using only 4 colors while ensuring that two neighboring countries receive different colors. It was (finally) demonstrated in 1976 by Appel and Haken. The demonstration required the use of the computer to study the 1478 critical cases (more than 1200 hours of calculation at the time).

3.1 Challenge

Given a set of countries *COUNTRIES* and a set of colors *COLORS*, what colors can be used to color each country so that two neighboring countries never have the same color?

Exercise 1 *The aim here is to describe the initial state. We will limit ourselves to a few European countries. Let*

$$COUNTRIES = \left\{ \begin{array}{l} austria, belgium, croatia, czechRepublic, france, germany, italy, \\ luxembourg, netherlands, poland, portugal, slovenia, spain, switzerland \end{array} \right\}$$

be a subset of European countries (we do not consider them all, but you can). We will use small caps because countries are not formulas and so, have no truth value.

Let's also define a set of (for the moment, four) possible colors:

$$COLORS = \{green, orange, pink, yellow\}.$$

1. *Define theses variables in a new* TouIST *file.*
2. *If $B(c_1, c_2)$ reads "country c_1 is bordering on c_2", describe the facts set modeling the countries map. (Note that "is bordering on" is a commutative relation: $B(france, spain)$ iff $B(spain, france)$, which* must *be modeled by a single logical using c_1 and c_2.)*

In case of doubt, to know which of these countries are bordering, consult this map.

Exercise 2 *If $C(c, r)$ reads "the country c is colored with the color r", write a formula meaning that each country $c \in COUNTRIES$ is colored with* at least *one color of $COLORS$.*

(Note this problem is similar with the Latin square problem.)

Exercise 3 *Write a formula meaning that each country $c \in COUNTRIES$ is colored with* at most *one color of $COLORS$.*

Exercise 4 *Write a formula which requires that two neighboring countries receive different colors.*

Exercise 5 *Use* TouIST *to discover a four-color solution. Then test if there is a three-color solution. Conclusion?*

3.2 To go further

Challenge: propose a modeling of the map coloring problem using only propositions $C(c, r)$.

The idea is to encode neighborhood relationships between countries using sets of sets of countries. Suppose that for all $c \in COUNTRIES$, $NEIGHBORS(c) \subseteq COUNTRIES \setminus \{c\}$ is the set of countries which border on c, and suppose also that $c_1 \in NEIGHBORS(c_2)$ if and only if $c_2 \in NEIGHBORS(c_1)$ (if two countries are neighbors, each country explicitly belongs to the list of neighbors of the other). So an expression like

$$\bigwedge_{\substack{c_1 \in COUNTRIES \\ c_2 \in COUNTRIES \\ \text{when } c_1 \neq c_2}} B(c_1, c_2) \rightarrow \ldots$$

can be written

$$\bigwedge_{\substack{c_1 \in COUNTRIES \\ c_2 \in COUNTRIES \\ \text{when } c_2 \in NEIGHBORS(c_1)}} \ldots \quad \text{or (more compact)} \quad \bigwedge_{\substack{c_1 \in COUNTRIES \\ c_2 \in NEIGHBORS(c_1)}} \ldots$$

In TouIST, you can write for instance:

```
$NEIGHBORS(austria)=[czechRepublic,germany,switzerland,italy,
                                                    slovenia]
```

Exercise 6 *In a new file, repeat Exercise 1 without using formulas like* $B(c_1, c_2)$.

Exercise 7 *Copy the answers from Exercises 2 and 3, then answer Exercise 4 without using formulas of type* $B(c_1, c_2)$. *Check the result.*

In fact, TouIST supports complex conditions in `when` clauses, such as in:

$$\text{when } c_2 \in NEIGHBORS(c_1) \vee c_1 \in NEIGHBORS(c_2)$$

for instance.

Exercise 8 *Explain why it is not necessary here to write something like that:*

$$\bigwedge_{\substack{c_1 \in COUNTRIES \\ c_2 \in COUNTRIES \\ when\ c_2 \in NEIGHBORS(c_1) \vee c_1 \in NEIGHBORS(c_2)}} \ldots$$

4 Sudoku game

Sudoku is a grid-shaped game defined in 1979 by the American Howard Garns, but inspired by the Latin square, as well as the problem of the 36 officers of the Swiss mathematician Leonhard Euler.

The object of the game is to fill the grid with a series of all different symbols, which are never found more than once in the same row, the same column or the same region (also called block, group, Sector or sub-grid). Most of the time, the symbols are numbers in $[1..N^2]$ where $N \in \mathbb{N}$, the regions then being squares of $N \times N$ cells and the Sudoku being square of $N^2 \times N^2$ cells. Some symbols are already arranged in the grid, which allows a progressive resolution of the complete problem in a unique way.

4.1 Challenge

Write the formulas to solve any Sudoku of finite dimension $N^2 \times N^2$.

First, we will take $N = 2$. The Sudoku will therefore be a 4×4 grid composed of four 2×2 regions. The set of autorized symbols is $VAL = \{1, 2, 3, 4\}$. The set of both row and column indices is $IND = \{1, 2, 3, 4\}$. Our current example is the following grid which illustrates the initial state:

1	2		
	4		
		3	2

4.2 Where it all starts like a Latin square

Exercise 9 *Describe the initial state of the grid by defining: a variable N, the sets IND and VAL (with respect to N), as well as the already known values of the grid using formulas of type $C(i, j, v)$ with $i, j \in IND$ and $v \in VAL$.*

Exercise 10 *Add the four constraints as if it was a Latin square:*

1. *any cell has at least one of the values of VAL;*
2. *any cell has at most one of the values of VAL;*
3. *the same value appears on the same line at most;*
4. *the same value appears at most on the same column.*

Now, if you press the *Solve* button, you will find that you get two different models, but one is not a Sudoku because it does not respect the constraint that the same value must appear at most 1 time in each region (white or orange).

So, we must add an additional constraint saying that the same value cannot appear more than once *in a given region* (white or orange in our above example).

Exercise 11 *First step: start by writing the constraint which says that for any pair of boxes (i_1, j_1) and (i_2, j_2) such as $i_1, j_1, i_2, j_2 \in IND$, and i_1 $neqi_2$ or j_1 $neqj_2$, it is false that they have identical values. (Not that, consequently, your program will become unsatisfiable.)*

Exercise 12 *Second step: replace the fact that $i_1, j_1, i_2, j_2 \in IND$ by the fact that for any region (ri, rj) where $ri, rj \in 1..2$ (which makes four regions well), the indices i_1, j_1, i_2, j_2 vary between the first line (respectively, column) and the last line (respectively, column) of the zone considered.*

Solve this new system, and make sure you only get one model, where each value only appears once in each area.

Exercise 13 *Modify your program to obtain a classic Sudoku 9×9 (*i.e.*, $N = 3$), then enter the initial state of the grid as described here (one of the most difficult Sudoku in the world) .*

What solution do you get?

Note that writing $\bigwedge_{i \in [1..20]} ...$ is more effective than writing $\bigwedge_{\substack{i \in \mathbb{Z} \\ i \in [1..20]}} ....$

Exercise 14 *We can note in particular a notable difference in the framework of the previous exercise. See solution proposed with the two entries, and test each of the two ways of writing the last rule.*

Authors: **Olivier GASQUET** UPS-IRIT/LILaC
Dominique LONGIN CNRS-IRIT/LILaC
Frédéric MARIS UPS-IRIT/ADRIA