SAARLAND UNIVERSITY ARTIFICIAL INTELLIGENCE
Prof. Jana Koehler
Prof. Jörg Hoffmann

Practical Exercise Sheet 3

Solutions due Sunday, June 13, 23:59.

Note: Please only use boolean variables.

Exercise 1: Valid Truth assignment 4 Points

Given is the following formula in propositional logic. Write a program in MiniZinc to find a valid truth assignment for the following formula:

$$\neg\,[(P \land Q) \Rightarrow \neg\,(P \land R)] \lor [(Q \land R) \land \neg\,(P \land \neg\,Q)]$$

Exercise 2: Tautology 4 Points

You are given the following two formulas in propositional logic. Prove **indirectly** in MiniZinc that the first formula is a tautology and the second is not a tautology:

(i) $(P \Rightarrow Q) \lor (Q \Rightarrow P)$

(ii) $P \Rightarrow ((P \lor Q) \land \neg\,Q)$

Exercise 3: European countries 5.5 Points

A new agreement between Germany, the Netherlands, Belgium, Luxembourg, France, Switzerland, and Ireland should be signed. Unfortunately, the different countries only sign the agreement under the following conditions:

1. Belgium signs the agreement if and only if France signs the agreement.
2. Switzerland does not sign the agreement.
3. Luxembourg signs the agreement if and only if Switzerland does not sign the agreement.

4. If France does not sign the agreement, then Germany also does not sign the agreement.
5. If Luxembourg and Belgium sign the agreement also the Netherlands signs the agreement.
6. Germany and Luxembourg sign both the agreement or they both do not sign it.
7. At least one of Ireland, Germany, and Luxembourg signs the agreement.

Write a program in MiniZinc to find out if a country signs the agreement or not. Use the first letter of each country as an abbreviation that this country signs the agreement (e.g., `G` denotes Germany signs the agreement).
You should make MiniZinc print all solutions (see "How to display all possible solutions" on the cheat sheet for how to do so). You should have multiple solutions. This shows that we currently do not have enough information to tell for each country whether it signs the agreement or not.

Exercise 4: European countries Part II 2 Points

We are considering the scenario from Exercise 3 again. Say we have the additional information that at most two of the following countries sign the agreement: Ireland, the Netherlands, and Belgium.

Write a new MiniZinc program (make use of your solution from Exercise 3) to check if the problem is still satisfiable and whether we can now say clearly which of the countries will sign the agreement or not.

Submission Instructions

Solutions need to be packaged into a `.zip` file and uploaded in the AI CMS. The `.zip` file has to contain a single folder with name:
`AI2021_PE3_mat1_mat2_mat3` where `mat1, mat2, mat3` are the matriculation numbers of the students who submit together. This folder must contain the following files:

- `authors.txt` listing the names and matriculation numbers of all students who submit together. Use one line per student and no spaces: Name;Matriculation number.
- The MiniZinc files containing your solutions.

Do not add any other folder or sub folder, this means place all files directly into `AI2021_PE3_mat1_mat2_mat3`. Do not place any file outside of `AI2021_PE3_mat1_mat2_mat3`.

Only one student of each group needs to do the submission! Remember that this sheet can be submitted in groups of up to three members (all members of the group must however be assigned to the same tutorial).

MiniZinc Cheat Sheet (for MiniZinc Version 2.4.3 included in VM)

How to define a Variable

In MiniZinc the syntax to define a boolean variable is:

`var bool:` *nameOfVariable*`;`

where *nameOfVariable* needs to be replaced with the name of the variable you are defining. There are further types of variables (integer `int`, floating point number `float`), which you will not need for the exercises on this exercise sheet. To define multiple variables, just list them.

Example: `var bool: A;`

How to define the type of Problem

In all the exercises on this sheet, the problems are satisfaction problems (ie. we check for the satisfiability of the problem). In these cases, you must add the line

`solve satisfy;`

after you have defined all your variables. Other types of problems (which do not occur on this exercise sheet) are `solve maximize` *functionToMaximize*`;` and `solve minimize` *functionToMinimize*`;`.

Example: `solve satisfy;`

How to define a Logical Formula (as a Constraint)

To define a constraint in MiniZinc, the syntax is

`constraint` *yourConstraint* `;`

where *yourConstraint* needs to be replaced by the constraint you wish to formulate. To formulate constraints, use the variables you have defined together with the logical operators defined below. Note that in MiniZinc a boolean variable is assigned value 1 if it is true and 0 if it is false. To define multiple constraints, just list them.

Example: `constraint not A;`

How to print Results

To ease the reading of the resulting complete assignment of each decision variable, use the following instruction, where X is the first and Y is the second decision variable:

```
output ["NameFirstVariable = \(X)\n",
   "NameSecondVariable = \(Y)\n"];
```

Note that you can add further decision variables to your output using the same syntax.

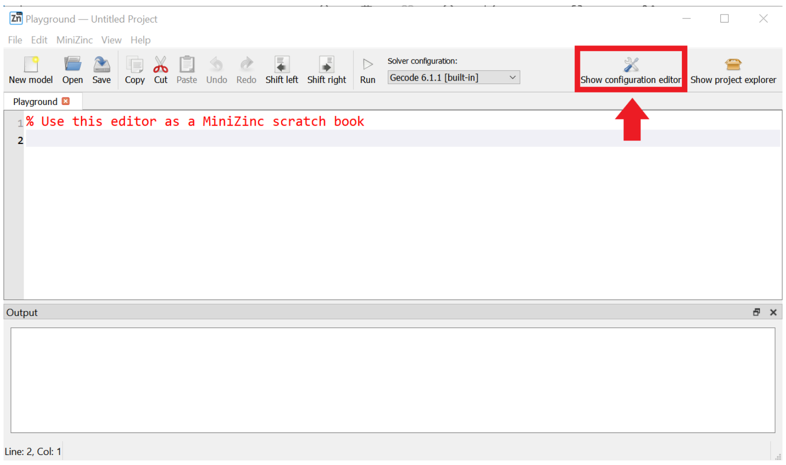
Example: `output [ "A = \(A)\n", "B = \(B)\n", "C = \(C)\n"];`

List of logical operators

Logical statement	Input in MiniZinc
$A \wedge B$	`A /\ B`
$A \vee B$	`A \/ B`
$A \Rightarrow B$	`A -> B`
$A \Leftrightarrow B$	`A <-> B`
$\neg A$	`not A`
A is TRUE	`A = 1`
A is FALSE	`A = 0`
A and B have the same truth value	`A = B`

How to display all possible solutions

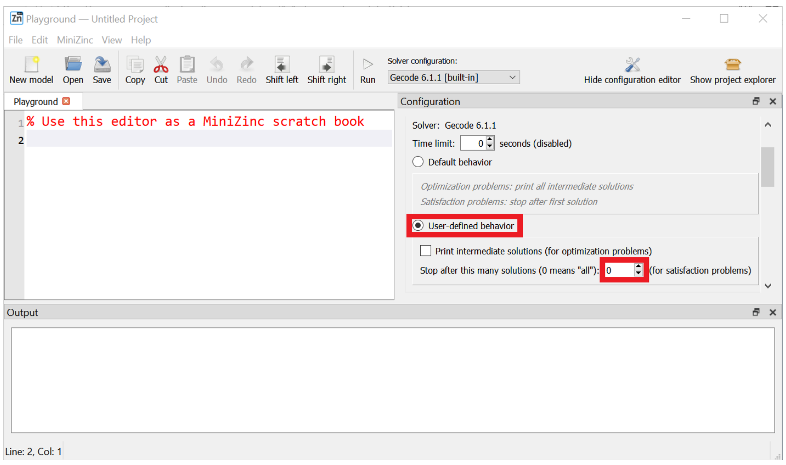
To display all possible solutions on MiniZinc, press the “Show configuration editor” button as seen in the screenshot below:
Once you have opened the configuration editor, scroll down and select “User-defined behaviour”. Then set the number of solutions for satisfaction problems to 0. This way, MiniZinc will display all possible solutions to satisfaction problems.


Playground — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run
Show configuration editor
Show project explorer
Playground
Output

Solver configuration: Gecode 6.1.1 [built-in]

```
% Use this editor as a MiniZinc scratch book
```

Line: 2, Col: 1


Playground — Untitled Project
File Edit MiniZinc View Help
New model Open Save Copy Cut Paste Undo Redo Shift left Shift right Run
Hide configuration editor
Show project explorer
Playground
Configuration
Output

Solver configuration: Gecode 6.1.1 [built-in]

```
% Use this editor as a MiniZinc scratch book
```

Solver: Gecode 6.1.1

Time limit: 0 seconds (disabled)

Default behavior

Optimization problems: print all intermediate solutions

Satisfaction problems: stop after first solution

[✓ User-defined behavior]

☐ Print intermediate solutions (for optimization problems)

Stop after this many solutions (0 means "all"): 0 (for satisfaction problems)

Line: 2, Col: 1

A	B	A AND B
0	0	0
1	0	0
0	1	0
1	1	1

A	B	A OR B
0	0	0
1	0	1
0	1	1
1	1	1

Table 1: As a reminder the truth-tables for AND and OR.