This document shows the ingredients of a good assignment or project report for the M4CO course. Its \LaTeX{} source code exemplifies almost everything you need to know about \LaTeX{} in order to typeset a professional-looking assignment or project report (for the M4CO course). Use the source code of this document as a starting point for imitation, whether you use \LaTeX{} or not, and delete everything irrelevant. The \LaTeX{} source code of this document also contains useful comments. The usage of \LaTeX{} is \textit{optional}, but highly recommended, for reasons that will soon become clear to those who have never used it before. Any learning time of \LaTeX{} is \textit{outside} the time budget of this course, but will hugely pay off, if not in this course then in the next course(s) you take and when writing a dissertation or other scientific report.

Address \textit{each} task of \textit{each} problem, using the numbering and ordering in which the problems and tasks appear in the assignment statement:

- For each task requiring a model, follow the advice and structure of Section 1.D.
- For each task requiring an evaluation, follow the structure of Section 1.E.

Delete all unnecessary text and suitably replace all other text in this document.

1 The Minimal Magic Square Problem

All experiments were run under Linux Ubuntu 16.04 (64 bit) on an Intel Xeon E5520 of 2.27 GHz, with 4 processors of 4 cores each, with a 24 GB RAM and an 8 MB L2 cache (a ThinLinc computer of the IT department).

(If different hardware was used for different tasks, then justify this and replicate a paragraph like this one within each relevant sub-section.)

1.A Answer for the First Task

(Write your answer for the first task here.)

1.B Answer for the Second Task

\ldots

\footnote{Hint: Under Linux, do \texttt{more /proc/cpusinfo} to find this information. Under macOS, you find this information via “About This Mac” in the Apple menu.}
1.C Use Appropriate Titles for the Subsections

(Remember that this document is a guideline, so that you should change its structure and text when appropriate.)

1.D Model for the Minimal Magic Square Problem

A MiniZinc model must include comments that explain:

- the parameters that are not part of the possibly provided skeleton code;
- the decision variables;
- the problem constraints;
- the objective;
- the redundant decision variables, if any;
- the channelling constraints, if any;
- the implied constraints, if any; and
- the symmetry-breaking constraints, if any.

The quality of model comments is considered while grading. See Listing 1 for an example of sufficiently good model comments; you must include your model in your report in addition to submitting it as a separate file.

1.D.1 Description

(Provide, in the report, a more detailed explanation why each channelling, implied, and symmetry-breaking constraint is correct. Use any combination of in-lined code, mathematical notation, and plain English to explain your constraints.)

Channelling Constraints. Our model has no channelling constraints, because it has no redundant decision variables.

Implied Constraints. Since the sum of each row must be equal to the magic sum, and since all values from 1 to \( n^2 \) must occur exactly once in the magic square, it is implied that the sum of the rows is the sum of the entire magic square: \( n \times \text{magicSum} = \text{sum}(1..n^2) \). Since \( n \) is a parameter, this implied constraint actually fixes the decision variable \( \text{magicSum} \), namely to \( \text{sum}(1..n^2) \div n \), that is to \( (n\times n^2 + n) \div 2 \). We reckon this constraint is useful to solvers of all technologies and hence do not flag it using the \text{implied_constraint} predicate.

Symmetry-Breaking Constraints. A magic square has rotation and reflection symmetries:

- The rotation symmetries for 90, 180, and 270 degrees and the reflection symmetries for the horizontal axis, vertical axis, and up-right diagonal can be broken by requiring the top-left corner of the magic square to be smaller than the other corners.
- The reflection symmetry for the down-right diagonal can be broken by requiring the bottom-left corner to be smaller than the top-right corner.

These constraints are flagged using the \text{symmetry_breaking_constraint} predicate.
Listing 1: A MiniZinc model for the Minimal Magic Square problem

```mini
include "globals.mzn";

int: n; % the width (and height) of the magic square
set of int: N = 1..n; % the index range for the rows and columns

% Magic[r,c] = the value at row r and column c of the magic square:
array[N,N] of var 1..n*n: Magic;
var int: magicSum; % the sum of each row, column, and major diagonal

% All values in the magic square are different:
constraint all_different([Magic[r,c] | r, c in N]);

% The sum of each row is equal to the magic sum:
constraint forall(r in N)(sum([Magic[r,c] | c in N ] ) = magicSum);
% The sum of each column is equal to the magic sum:
constraint forall(c in N)(sum([Magic[r,c] | r in N ] ) = magicSum);
% The sum of each major diagonal is equal to the magic sum:
constraint sum([Magic[i,i] | i in N]) = magicSum;
constraint sum([Magic[n+1-i,i] | i in N]) = magicSum;

% Each row sums up to the magic sum, there are n rows, % and their sum is equal to the sum of the entire magic square, % hence the following implied constraint fixes magicSum % and is useful under all solving technologies:
constraint n * magicSum = sum(1..n*n); % implied constraint

% Break the three rotation symmetries and three of the four % reflection symmetries by requiring the top–left corner to % be smaller than the other corners:
constraint symmetry_breaking_constraint(
    Magic[1,1] < Magic[1,n] /\
    Magic[1,1] < Magic[n,1] /\
    Magic[1,1] < Magic[n,n]
);
% Break the down–right–diagonal reflection symmetry by requiring the % bottom–left corner to be smaller than the top–right corner:
constraint symmetry_breaking_constraint(Magic[n,1] < Magic[1,n]);

% Minimise the sum of the corners of the magic square:
solve minimize sum([Magic[i,j] | i,j in {1,n}]);

output
   ["Magic sum: ", show(magicSum), "\nMagic square: "] ++
   [show(Magic[r,c]) ++ if c=n then "\n" else " " endif | r,c in N]++
   ["CornerSum: ", show(sum([Magic[i,j] | i,j in {1,n}]))], "\n"]
   ++ ["Corners: ", show([Magic[i,j] | i,j in {1,n}])];
```
1.D.2 Implementation

The described model, with the prescribed comments, is uploaded as file \texttt{minMagicSquare.mzn}.

\textbf{Compilation and Running Instructions.} In order to compile and run \texttt{minMagicSquare.mzn} one must supply a value for the parameter \( n \) using the \texttt{-D} option in the command line. Assuming that \texttt{mzn-backend} is one of the backends provided for the course, the model can be compiled and run for \( n=5 \) by typing \texttt{mzn-backend minMagicSquare.mzn -D "n=5"} at the command line.

\textbf{Sample Test-Run Command.}

\begin{verbatim}
> mzn-gecode minMagicSquare.mzn -D "n=5"
Magic sum: 65
Magic square:
  4  9 18 24 10
 13 23  3 11 15
 19  1 25  6 14
 22 12  2  8 21
  7 20 17 16  5
CornerSum: 26
Corners: [4, 10, 7, 5]
---------
\% time elapsed: 124100 ms
---------
\end{verbatim}

1.E Evaluation of the Minimal Magic Square Model

We have chosen the backends for Gecode, Chuffed, Gurobi, OscaR.cbls, and Lingeling. (Choose, without justification, any backends you want, as long as you cover all the considered technologies.) Table 1 gives the results for various values of \( n \) on our Minimal Magic Square model. The time-out was 300 seconds.

(Refer to and explain the table(s), draw some conclusions from the experiment, and analyse:) We observe that the chosen MIP backend wins overall, as it is the only one not to time out for the largest chosen instance, with \( n = 6 \). The chosen MIP and SAT backends scale best, as they are the only ones to establish feasibility for \( n = 6 \), even though the SAT objective value of 39 at time-out is far above the minimum of 26 proven by the MIP backend. On small instances, with \( n \in \{3, 4\} \), the chosen CP backend wins, narrowly defeating the chosen LCG backend followed by the MIP and SAT backends. The SAT backend is the only one where \( n = 4 \) is harder than the smaller \( n = 3 \). Starting from the medium instance, with \( n = 5 \), the MIP backend clearly wins, with only the CP backend also not timing out, the LCG backend finding but not proving the minimum, and the SAT backend missing the minimum by one unit. The chosen CBLS backend always times out, because local search can by construction not prove minima on problems, such as here, where the trivial lower bound (namely 10 = 1 + 2 + 3 + 4 here) on the objective value is not feasible; it finds the known minima for \( n \in \{3, 4\} \), but is far above the known minimum for \( n = 5 \) and cannot establish feasibility for \( n = 6 \) before timing out.

\textbf{Hint:} Use a script, such as the one explained in \url{http://user.it.uu.se/~gusbj192/courses/M4CO/cheatsheet.pdf}, that conducts the final experiments for you and directly generates a result table (see the \LaTeX{} source code of Table 1), which is automatically imported (rather than manually copied) into your report: each time you change the model, it suffices to re-run that script and re-compile your report, in your absence, without any tedious copying!
Table 1: Results for our Minimal Magic Square model. In the ‘time’ column, if the reported time is less than the time-out (300 seconds here), then the reported objective value in the ‘CornerSum’ column was proven optimal; else the time-out is indicated by ‘t/o’ and the reported objective value is either the best value found, but not proven optimal, before timing out, or ‘–’, indicating that no feasible solution was found before timing out. If the reported time is ‘–’, then that instance was not run on that backend, as the latter timed out on a smaller instance (this is normally not a correct assumption, as larger instances can be easier, but for the assignments of this course this assumption is fine).
1.F Answer for Another Task

1.G Alternative Model for the Minimal Magic Square Problem
1.G.1 Description
Channelling Constraints.
Implied Constraints.
Symmetry-Breaking Constraints.

1.G.2 Implementation

1.H Answer for Yet Another Task

Feedback
Please write a paragraph, which will not be graded, describing your experience with this assignment: which aspects were too difficult or too easy, and which aspects were interesting or boring? (Do not forget that optimisation is a highly experimental discipline, so conducting a lot of experiments and reporting on their results is inevitable; also recall the hint above.) This may help us improve the course for the next year.

References

Crash Report
For instances where no result is reported, due to a solver crashing or the MiniZinc toolchain failing to compile the model, please include an error message, if there is any. For example:

largecumulative.mzn For the larger instances, OscaR.cbls crashes with the following exception, which seems to be caused by the JVM not being allocated enough heap space.

```
> /usr/bin/oscar-cbls -u -t 3600 /tmp/tmp.fzn
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
at scala.collection.mutable.FlatHashTable$class.growTable(FlatHashTable.scala:217)
at scala.collection.mutable.FlatHashTable$class.addEntry(FlatHashTable.scala:159)
at scala.collection.mutable.HashSet.addEntry(HashSet.scala:40)
at scala.collection.mutable.FlatHashTable$class.addElem(FlatHashTable.scala:139)
at scala.collection.mutable.HashSet.addElem(HashSet.scala:40)
at scala.collection.mutable.HashSet.$plus$eq(HashSet.scala:59)
at oscar.cp.scheduling.constraints.EnergeticReasoning$$anonfun$computeIntervals$1$$anonfun$apply$mcVI$sp$2.apply(EnergeticReasoning.scala:127)
at oscar.cp.scheduling.constraints.EnergeticReasoning$$anonfun$computeIntervals$1$$anonfun$apply$mcVI$sp$2.apply(EnergeticReasoning.scala:126)
```
Checklist before Submitting

In order to protect yourself against an unnecessary loss of points, and in order to show both self-respect and respect for the human reader of your report, use the following checklist before submitting:

- Crosscheck your report against the assignment instructions.

- Remember that when submitting you implicitly certify that your report and all its uploaded attachments were produced solely by your team, except where explicitly stated otherwise and clearly referenced, that each teammate can individually explain any part starting from the moment of submitting your report, and that your report and attachments are not freely accessible on a public repository.


- Spellcheck all documents, including the comments in the MiniZinc source code.

- Proofread, if not grammar-check, your report at least once per teammate.

More LaTeX and Technical Writing Advice

Unnumbered itemisation (only to be used when the order of the items does not matter)²

- Unnumbered displayed formula:

\[ E = m \cdot c^2 \]

- Numbered displayed formula (which is cross-referenced somewhere):

\[ E = m \cdot c^2 \] (1)

²Use footnotes very sparingly, and note that footnote pointers are never preceded by a space and always glued immediately behind the punctuation, if there is any.
Figure 1: A binary search tree, a binary min-heap, and a binomial tree of rank 3

- Formula — the same as formula (1) — spanning more than one line:

\[ E = m \cdot c^2 \]

Numbered itemisation (only to be used when the order of the items does matter):

1. First do this.
2. Then do that.
3. If we are not finished, then go back to Step 2 else stop.


Use \textit{...} in mathematical mode for each multiple-letter identifier in order to avoid typesetting the identifier like the product of single-letter ones. For example, note the typographic difference between the identifier \textit{WL}, obtained through \texttt{\textit{WL}} , and the product \texttt{WL}, where there is a small space between the \texttt{W} and the \texttt{L}, obtained through \texttt{\$WL\$}.

Do not use programming-language-style lower-ASCII notation (such as ! for negation, && for conjunction, || for disjunction, and the equality sign = for assignment) in algorithms or formulas (but rather use \texttt{\texttt{\neg}} or \texttt{\texttt{not}}, \texttt{\texttt{\&\&}} or \texttt{\texttt{and}}, \texttt{\texttt{\|\|}} or \texttt{\texttt{or}}, and \texttt{\texttt{\leftarrow}} or \texttt{\texttt{:=}}, respectively), as this testifies to a very strong confusion of concepts.

Figures can be imported with \texttt{\texttt{\includegraphics}} or drawn inside the \LaTeX source code using the highly declarative notation of the \texttt{tikz} package: see Figure 1 for sample drawings. It is perfectly acceptable in this course to include scans or photos of drawings that were carefully done by hand.

Algorithms can be typeset as pseudo-code as exemplified in Algorithm 1: study its \LaTeX source code.

If you are not sure whether you will stick to your current choice of notation or terminology, then introduce a new (possibly parametric) command. For instance, upon

\texttt{\newcommand{\Cardinality}[1]{\left\lvert#1\right\rvert}}

the formula \texttt{\$\Cardinality{S}\$} typesets the cardinality of set \texttt{S} as \texttt{|S|} with autosized vertical bars and proper spacing, but upon changing the definition of that parametric command to
Algorithm 1: Silly algorithm

\begin{algorithm}
\textbf{function} \texttt{f}(n) \texttt{\;}
\textbf{if} \; n < 0 \; \textbf{then} \; // optional comment
\texttt{n ← −2 · n} \; // optional comment
\textbf{else} \; // \; n ≥ 0
\texttt{n ← 3 · n} \; \textbf{\;}
\textbf{while} \; n > 0 \; \textbf{do} \; // optional comment
\texttt{n ← n − 1} \; \textbf{\;}
\texttt{return} \; n
\end{algorithm}

and recomiling, the formula $\texttt{Cardinality}(S)$ typesets the cardinality of set $S$ as \#$S$. Similarly, upon

\begin{verbatim}
\texttt{Cardinality}{	exttt{S}}
\end{verbatim}

the text \texttt{MiniZinc} typesets into \textit{MiniZinc}, hyphenation being only possible in the middle, but upon changing the definition of that non-parametric command to

\begin{verbatim}
\texttt{MiniZinc}{\textsc{Mini-Zinc}}
\end{verbatim}

and recomiling, the text \texttt{MiniZinc} typesets into MINIZINC. You can thus obtain an arbitrary number of changes in the document with a constant-time change in its source code, rather than having to perform a linear-time find-and-replace operation within the source code, which is painstaking and error-prone. The imported file macros.tex has a lot of useful predefined commands about mathematics, CP, MiniZinc, modelling, and algorithms.

Use commands on positioning (such as \texttt{hspace}, \texttt{vspace}, and \texttt{noindent}) and appearance (such as \texttt{small} for reducing the font size, and \texttt{textit} for italics) very sparingly, and ideally only in (parametric) commands, as the very idea of mark-up languages such as \LaTeX{} is to let the class designer (usually a trained professional typesetter) decide on where things appear and how they look. For instance, \texttt{\textbf{emph}} (for emphasis) compiles (outside italicised environments, such as \texttt{theorem}) into \textit{italics} under the \texttt{article} class used for this document, but it may compile into \textbf{boldface} under some other class.

If you do not (need to) worry about how things look, then you can fully focus on what you are trying to express!

Note that no absolute numbers are used in the \LaTeX{} source code for any of the references inside this document. For ease of maintenance, \texttt{\label} is used for giving a label to something that is automatically numbered (such as an algorithm, equation, figure, footnote, item, line, section, subsection, or table), and \texttt{\ref} is used for referring to a label. An item in the bibliography file is referred to by \texttt{\cite} instead. Upon changing the text, it suffices to recompile, once or twice, and possibly to run BibTeX again, in order to update all references consistently.

Prefer \texttt{Section~\ref{sec:minMagicSquare}} over \texttt{Section \ref{sec:minMagicSquare}}, using the non-breaking space (which is typeset as ~) instead of the normal space, as this always gives “Section I” and avoids that a cross-reference is spread across a line break, as in “Section I”, which is considered poor typesetting.

The rules of English for how many spaces to use before and after various symbols are given in Table 3. Beware that they may be very different from the rules in your native language.\footnote{Feel free to report to the head teacher any other features that you would have liked to see discussed and exemplified in this template document.}
<table>
<thead>
<tr>
<th>Topic</th>
<th>\LaTeX code</th>
<th>Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greek letter</td>
<td>$\Theta, \Omega, \epsilon$</td>
<td>$\Theta, \Omega, \epsilon$</td>
</tr>
<tr>
<td>multiplication</td>
<td>$m \cdot n$</td>
<td>$m \cdot n$</td>
</tr>
<tr>
<td>division</td>
<td>$\frac{m}{n}$, $m \div n$</td>
<td>$\frac{m}{n}$, $m \div n$</td>
</tr>
<tr>
<td>rounding down</td>
<td>$\lfloor n \rfloor$</td>
<td>$\lfloor n \rfloor$</td>
</tr>
<tr>
<td>rounding up</td>
<td>$\lceil n \rceil$</td>
<td>$\lceil n \rceil$</td>
</tr>
<tr>
<td>binary modulus</td>
<td>$m \mod n$</td>
<td>$m \mod n$</td>
</tr>
<tr>
<td>unary modulus</td>
<td>$m = n \mod \ell$</td>
<td>$m = n \mod \ell$</td>
</tr>
<tr>
<td>root</td>
<td>$\sqrt{n}$, $\sqrt[n]{n}$</td>
<td>$\sqrt{n}$, $\sqrt[n]{n}$</td>
</tr>
<tr>
<td>exponentiation, superscript</td>
<td>$n^{i}$</td>
<td>$n^{i}$</td>
</tr>
<tr>
<td>subscript</td>
<td>$n_{i}$</td>
<td>$n_{i}$</td>
</tr>
<tr>
<td>overline</td>
<td>$\overline{n}$</td>
<td>$\overline{n}$</td>
</tr>
<tr>
<td>base 2 logarithm</td>
<td>$\log_{2} n$</td>
<td>$\log_{2} n$</td>
</tr>
<tr>
<td>base b logarithm</td>
<td>$\log_{b} n$</td>
<td>$\log_{b} n$</td>
</tr>
<tr>
<td>binomial</td>
<td>$\binom{n}{k}$</td>
<td>$\binom{n}{k}$</td>
</tr>
<tr>
<td>sum</td>
<td>$\sum_{i=1}^{n} i$</td>
<td>$\sum_{i=1}^{n} i$</td>
</tr>
<tr>
<td>numeric comparison</td>
<td>$\leq, \neq, &gt;, \geq$</td>
<td>$\leq, \neq, &gt;, \geq$</td>
</tr>
<tr>
<td>non-numeric comparison</td>
<td>$\prec, \nprec, \preceq, \succeq$</td>
<td>$\prec, \nprec, \preceq, \succeq$</td>
</tr>
<tr>
<td>extremum</td>
<td>$\min, \max, +\infty, \bot, \top$</td>
<td>$\min, \max, +\infty, \bot, \top$</td>
</tr>
<tr>
<td>function</td>
<td>$f: A \to B, \circ, \mapsto$</td>
<td>$f: A \to B, \circ, \mapsto$</td>
</tr>
<tr>
<td>sequence, tuple</td>
<td>$\langle a, b, c \rangle$</td>
<td>$\langle a, b, c \rangle$</td>
</tr>
<tr>
<td>set</td>
<td>${a, b, c}, \emptyset, \mathbb{N}$</td>
<td>${a, b, c}, \emptyset, \mathbb{N}$</td>
</tr>
<tr>
<td>set membership</td>
<td>$\in, \not\in$</td>
<td>$\in, \not\in$</td>
</tr>
<tr>
<td>set comprehension</td>
<td>${i \mid 1 \leq i \leq n}$</td>
<td>${i \mid 1 \leq i \leq n}$</td>
</tr>
<tr>
<td>set operation</td>
<td>$\cup, \cap, \setminus, \times$</td>
<td>$\cup, \cap, \setminus, \times$</td>
</tr>
<tr>
<td>logical quantifier</td>
<td>$\forall, \exists, \not\exists$</td>
<td>$\forall, \exists, \not\exists$</td>
</tr>
<tr>
<td>logic connective</td>
<td>$\land, \lor, \neg, \Rightarrow$</td>
<td>$\land, \lor, \neg, \Rightarrow$</td>
</tr>
<tr>
<td>logical</td>
<td>$\models, \equiv, \vdash$</td>
<td>$\models, \equiv, \vdash$</td>
</tr>
<tr>
<td>miscellaneous</td>
<td>$&amp;, #, \approx, \sim, \ell$</td>
<td>$&amp;, #, \approx, \sim, \ell$</td>
</tr>
<tr>
<td>dots</td>
<td>$\ldots, \cdots, \vdots, \ddots$</td>
<td>$\ldots, \cdots, \vdots, \ddots$</td>
</tr>
<tr>
<td>dots (context-sensitive)</td>
<td>$\dots, \ldots, n$</td>
<td>$1, \ldots, n$</td>
</tr>
<tr>
<td>parentheses (autosizing)</td>
<td>$\left(m^{n-k}\right)$</td>
<td>$(m^{n-k})$</td>
</tr>
<tr>
<td>identifier of &gt; 1 character</td>
<td>$\text{mathit{identifier}}$</td>
<td>$\text{mathit{identifier}}$</td>
</tr>
</tbody>
</table>

Table 2: The typesetting of elementary mathematics. Note very carefully when italics are used by \LaTeX and when not, as well as all the horizontal and vertical spacing performed by \LaTeX.

<table>
<thead>
<tr>
<th>number of spaces after</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of spaces before</td>
<td>0</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(</td>
</tr>
</tbody>
</table>

Table 3: Spacing rules of English