

Please submit your solution by sending e-mail to mrg@cs.ubc.ca. Your submission should include the following two files as attachments:

hw1.txt or hw1.pdf – plain, ASCII text or PDF.

hw1.py source code for your solutions.

1. Logic Puzzles (10 points)

I found a source of on-line logic puzzles at <http://www.logic-puzzles.org/init.php>. Use their puzzle generator to generate a puzzle with a 4x5 grid, and use z3 to solve it.

- Include the original statement with your solution.
- You are welcome to use the functions I’ve provided in [puzzle_utils.py](#).
- If you’d like to solve a different logic puzzle, feel free to do so. It doesn’t have to be a grid puzzle. The answer shouldn’t be obvious. For example, here’s a list of “The 10 Hardest logic puzzles ever”: <http://www.conceptispuzzles.com/index.aspx?uri=info/article/424>, and the “Ponder this” puzzles from IBM are challenging (and might land you a job ☺): <https://www.research.ibm.com/haifa/ponderthis/index.shtml>. Sudoku puzzles don’t count, because there’s a Sudoku solver in the Z3 documentation. Sudoku variants such as “Calcudoku” are fine. Other sources are fine.

2. Symbolic Software Execution (20 points)

I have hand-translated the binary search code from Program 3.5 of “Satisfiability Modulo Theories: Introduction and Applications” to a representation in Z3. See the python code in [q2.py](#). When I run the code:

```
python q2.py
```

I get the result:

```
my_prove(And(mid >= 0, mid < high)): refuted, here’s a counter-example
  [high = 1646799189, mid = 3247833024, low = 553899562]
  ...
```

Which shows that the first assertion failed.

- Fix the error in the original code
- Update the Z3 version, show that it now passes.
- Do the other tests pass? If not, diagnose the errors and fix them.

3. Positive Definite Matrices (20 points)

Machine-learning involves lots of optimization problems, especially convex optimization. Such problems often involve positive-definite matrices.

Let A be an $n \times n$ matrix. A is *positive definite* iff for every $n \times 1$ column vector x where $x \neq 0$, $x^T A x > 0$ where $x = 0$ means all elements of x are 0; $x \neq 0$ means at least one element of x is not 0; and x^T is the transpose of x (i.e. the corresponding row vector).

Consider the following three properties for matrix A :

- A is *symmetric* if for all $0 \leq i, j < n$, $A[i, j] = A[j, i]$ where I’m writing $A[i, j]$ to denote the element in row i and column j , and indices range from 0 to $n - 1$ (to match the conventions of python).
- A is *element-wise positive* if for all $0 \leq i, j < n$, $A[i, j] > 0$.
- A is *diagonally dominant* if for all $0 \leq i < n$,

$$A[i, i] > \sum_{j \neq i} |A[i, j]|$$

For each pair of conditions below, use Z3 to show if the pair of conditions is sufficient to guarantee that A is positive-definite, or give a counter-example, when A is a 3×3 matrix.

- (a) A is symmetric and element-wise positive.
- (b) A is symmetric and diagonally dominant.
- (c) A is element-wise positive and diagonally dominant.

I've provided some functions in [z3mat.py](#) to use Z3 with the `numpy` package. To give an example, I've provided [q3x.py](#) which uses Z3 to prove $(x^T A^T)^T = Ax$, again for the case where A is a 3×3 matrix. I also use Z3 to provide a counter-example to the putative equality $x^T A = (Ax)^T$.

4. Papers and Projects (10 points)

- (a) Are there any topics that you'd like to see covered that aren't on the reading list?
- (b) Are there any topics that you'd like to see covered in more depth than what you see on the reading list?
- (c) Do you have an areas of interest for a term project?
 - i. If "yes", please sketch your idea. This should be much less detailed than your eventual project proposal.
 - ii. If "no", let me know if you'd like to meet to brainstorm on ideas.

5. Bugs?

I've tested all the links and run simple examples with all the code that I've posted. Of course, there may be bugs lurking in the code or html. If you find an error, please let me know. The first person to report an error gets an extra credit bonus. If you find something frustrating to use and aren't sure if it's a bug or just that you're confused, send me e-mail. I'll be happy to think about it and fix the bug (and give you a bug bounty) or send a clarification. Thanks.

Here are some more references that you may find useful:

- Z3 documentation. I've put copies of the Z3py user's guide at the following two links: [Z3py Guide](#), [Z3py Advanced Examples](#). Most Z3 functions have docstrings. Using the python `dir` function on Z3 objects and then printing the docstrings for various methods can be helpful.
- The [Python 2.7 Tutorial](#). Note: Z3py is written for Python 2.7 and doesn't work with Python 3.
- The [numpy user's guide](#).