



## CSE200 HW4 Winter 2019 - Instructions for Z3

- Download Z3. [<https://github.com/Z3Prover/z3>] You can either download the source and build or download a pre-built version - this is up to you. The pre-built version will allow you to use Python bindings. If you want to use a different language, you can build from source with options that provide bindings to a few more languages: C/C++, Java or OCaml. Try Z3 on some small examples to make sure it's working properly. Lots of examples are available on GitHub in different languages.

- Your usage of Z3 will roughly consist of the following steps:

0. Import Z3.

1. Create variables for your SAT instance, which depends on the details of your reduction.

- You'll need to create *Boolean* variables. This is done by writing `Bool()` where is a string. For example, in Python:

```
V = [Bool(str(i)) for i in range(0, n)]
```

creates a list of Boolean variables  $\{V_0, \dots, V_n\}$ .

2. Create a solver instance. In Python:

```
S = z3.solver()
```

3. Add constraints to the solver. In Python, this looks like:

```
S.add(<insert constraint here>)
```

to encode your reduction from SAT to Sudoku. For example (again in Python):

- If  $V$  is a Boolean variable

```
V = Bool("1")
S.add(Not(V))
```

constrains the resulting satisfying assignment to  $V = 0$ .

- If  $A, B, C$  are Boolean variables, we can encode the constraint  $A \vee B \vee C$  as

```
A = Bool("A"), B = Bool("B"), C = Bool("C")
S.add(Or[A, B, C])
```

4. For checking satisfiability and finding the satisfying assignment, use:

```
S.check() #checks if the constraint system is satisfiable  
S.model() #produces the satisfying assignment
```

- Tips:

- For building the constraints in bulk, you can write down  $Or(< list >)$  or  $And(< list >)$ .
- Make sure to add constraints corresponding to the board positions which are already set, given the puzzle. That is, if  $M(i, j) = k$  is given then make sure to add this as a constraint. The specifics of how to do this depend on your reduction.
- Using the satisfying assignment (if it exists), you can extract a solution to the puzzle, which will depend on exactly how the variables in the formula correspond to configurations of the puzzle.