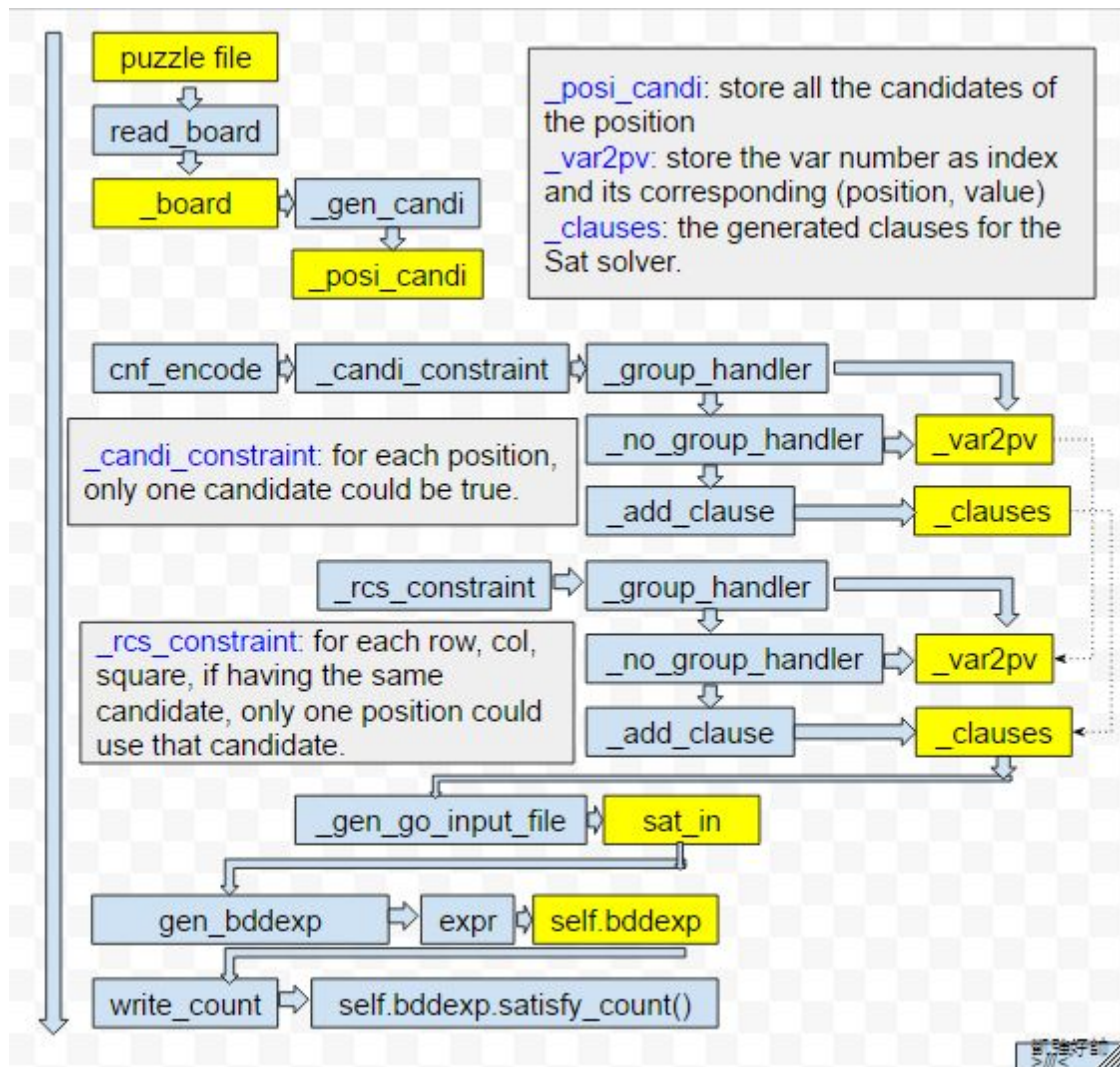


Homework 2 Report

0456008 翁慶年

1. Architecture of My Implementation



The blocks in yellow stand for variables.

The blocks in light blue stand for functions.

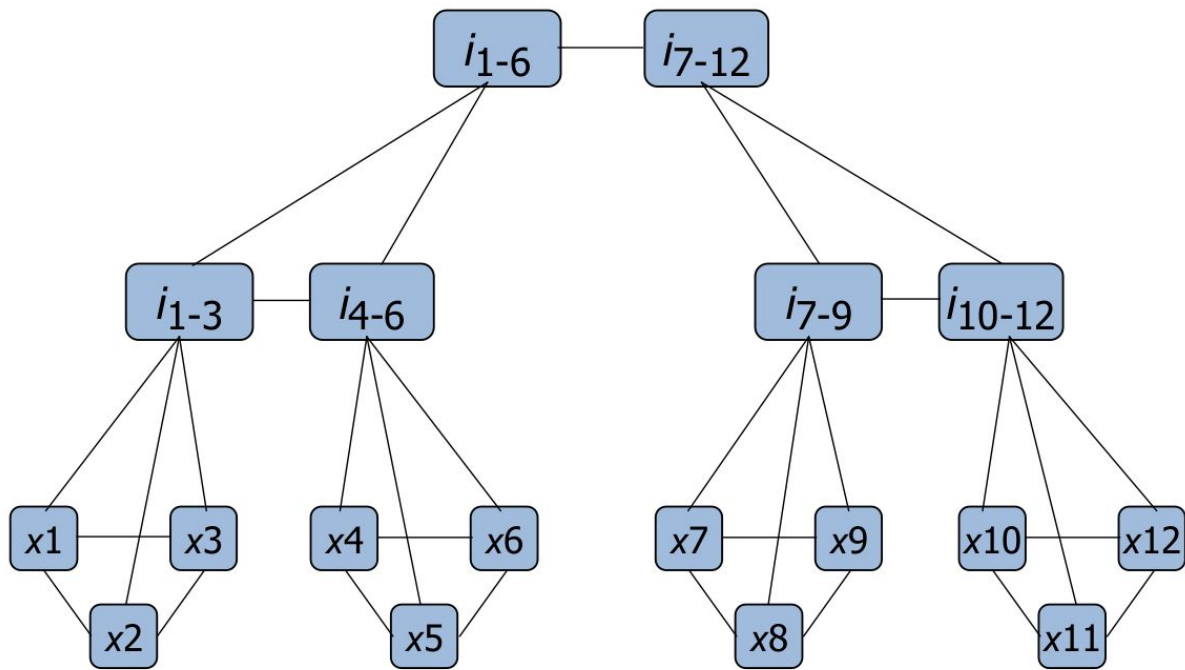
The blocks in grey stand for some notes about the functions or variables.

The flow of processing is from top to down, from left to right.

2. CNF Encoding Method

I use the method mentioned in the course slide "SAT Encoding for Sudoku Puzzles".

I also implemented Hierarchical Grouping to reduce the number of clauses.



3. How to

`python3 cnf_bdd.py [Input Puzzle] [output filename]`

In addition to [output filename], `sat_in`, `var2pv`, `posi_candi` would also be generated. The first one(`sat_in`) is converted to boolean expression in string format and fed to `expr` function, then use `satisfy_count()` to get the count of solutions. The others are just used for debugging.

4. Some Notes

actually I use the same encoding method as in HW1 and get good performance.

```
linux1 [/u/gcs/104/0456008/CT/HW2] -cnwong- % time python3 cnf_bdd.py 4.txt 4
0.129u 0.009s 0:00.15 80.0% 0+0k 0+32io 0pf+0w
linux1 [/u/gcs/104/0456008/CT/HW2] -cnwong- % time python3 cnf_bdd.py 9.txt 9
0.549u 0.023s 0:00.58 96.5% 0+0k 0+56io 0pf+0w
linux1 [/u/gcs/104/0456008/CT/HW2] -cnwong- % time python3 cnf_bdd.py 16.txt 16
2.126u 0.016s 0:02.15 99.0% 0+0k 0+160io 0pf+0w
```

I modified the code of HW1 and call the compiled exe file in python to get the encoded result of the input sudoku puzzle, then convert the CNF into format accepted by `expr` function, and use `satisfy_count` to get the count of solutions. If this homework should all be implemented in python, please tell me and I would spend another time revising the code ><