

**The File Format.** See Figure 1 for an example.

- The file can start with comments lines that start with the character `c`.
- Right after the comments, you need the special line

`p cnf $nvar $ncls`

where `$nvar` is the total number of variables (5 here) in the file, and `$ncls` is the total number of clauses (3 here) in the file.

- Then write down the clauses. Recall that each clause represent a disjunction of literals. The variables are represented by integers starting from 1. Each clause is a sequence of non-zero integers. The 0 at the end denotes the end of the clause.
- A positive integer represents the positive occurrence of the corresponding variable (positive literal), and a negative number the negation of the variable (negative literal).

```
1 c
2 c Comments start with a "c"
3 c
4 p cnf 5 3
5 1 -5 4 0
6 -1 5 3 4 0
7 -3 -4 0
```

Figure 1: CNF Format

So the file in Figure 1 encodes the formula

$$(p_1 \vee \neg p_5 \vee p_4) \wedge (\neg p_1 \vee p_5 \vee p_3 \vee p_4) \wedge (\neg p_3 \vee \neg p_4).$$

**Encoding Sudoku.** There are 81 spots in the grid. Each spot can take 9 possible values, which you can encode with binary representation with 4 binary variables. Thus, for each spot, you need 4 propositional variables<sup>1</sup>. For instance, suppose you want to say  $x_{1,1}$  takes value 5. The corresponding encoding is

$$\neg v_3^{(1,1)} \wedge v_2^{(1,1)} \wedge \neg v_1^{(1,1)} \wedge v_0^{(1,1)}$$

Namely, each  $v_k^{(1,1)}$  variable holds a digit for the value on the spot (1,1), and right now encoding the number 5 (0101).

You first need to generate the condition that each spot has to take values from 1 to 9. Then the condition that the spots in all the rows, columns, and the 9 smaller squares must contain different values (these are constraints on the value variables). Make sure that everything is in CNF (De Morgan's laws may suffice). Then, for each particular problem, add clauses that fix the values on spots whose numbers are already given.

---

<sup>1</sup>Of course, you can also allow some redundancy and use 9 variables to hold the 9 possible values (so only one of them can take 1 for a spot).