

## 1º Atividade Lógica para Computação

Quanto à equipe: No máximo 3 pessoas.

Prazo para entrega: 15/11

Atividade:

Implementar o algoritmo DPLL visto em sala e utilizá-lo para resolver qualquer problema de sudoku.

DICAS (não quer dizer que você vá fazê-las, mas elas ajudam):

1- A entrada do algoritmo DPLL pode ser feita em arquivo passado para o programa usando “<”. Exemplo: /a.out < sudoku.txt

Onde sudoku.txt tem um problema de sudoku na FNC.

2- O DPLL pode usar inteiros para representar literais positivos e negativos mapeados da entrada. Cada cláusula pode corresponder a uma linha do arquivo. Por exemplo:

$x_1 \quad x_2 \quad \neg x_3$

$x_3 \quad \neg x_2 \quad x_3$

Representa a fórmula  $(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$ . Mapeando para inteiros temos:

1 2 -3

3 -2 3

3- Para fazer o mapeamento da entrada do sudoku para inteiros você pode usar HashMap (Java), map (C++), ou A = {} (python).

Exemplo simples:

```
map<string,int> mapping;
```

```
int counter;
```

```
string formula;
```

```
cin >> formula;
```

```
if ( mapping.find(formula) == mapping.end() ) //nao tem o elemento mapeado
```

```
    mapping[formula] = counter++;
```

4- As restrições da fórmula do sudoku devem ser construídas conforme discutido em sala.