Mandatory Excercise 5

Computationally hard problems

October 11, 2015

Troels Thomsen 152165

Solution discussed with Rasmus Haarslev 152175

1 Excercise 5.4

1.0.1 NP-complete problem P_c

We are going to use 3-SAT as our NP-complete problem P_c.

1.0.2 Proof that 3-SAT $\leq$ 3-SAT-WITH-MAJORITY

We define our transformation algorithm T as follows:

1. For any clause c, take the first literal z.
2. Define two independent literals y_1 and y_2.
3. Find all instances z' of z in X including negations of z.
4. For each instance z' create a new clause $c' = z'y_1y_2$, and give y_1 and y_2 the same negation as z'.
5. repeat for all z in c.
6. repeat for all c in X.

This gives us the following pseudo-code, where $neg(a, b)$ returns the negation of a projected on b, and X' is the transformed set of clauses from input X, containing all the c' clauses.

```
for c[i] in X:
    for z[j] in c:
        declare y[j]1, y[j]2
        for z'[k] siblings of z[j] in X:
            c'[k] = z'[k] or neg(z'[k], y[j]1) or neg(z'[k], y[j]2)
            add c'[k] to X'
return X'
```

Show that if the answer to X is YES then so is the answer to $T(X)$. For any literal z we will generate two new literals y_i and y_j and a number of clauses corresponding to the number of times z literal occurs in the whole set - including times it appears with a negation.

$$c' = z \wedge y_i \wedge y_j \tag{1}$$

$$c' = \overline{z} \wedge \overline{y_i} \wedge \overline{y_j} \tag{2}$$

$$\vdots \tag{3}$$

Since we also negate the y's if the instance of z is negated, we end up with a set of clauses where half of them will always contain only true values of z has a truth assignment and vice vesa.

Show that if the answer to $T(X)$ is YES then so is the answer to X. In order to show that ...