Mandatory Excercise 2

Computationally hard problems

September 14, 2015

Troels Thomsen `152165`

Solution discussed with Rasmus Haarslev `152175`

1 Excercise 2.1

1.1 A

If we assume that C is of conjunctive normal form (a conjunction of disjunctions), we can define $A_o(C)$ in the following manner.

```
A_o(C):
    G = graph(C, ∅)
    for c in C:
        for x in c:
            for cr in {c ∉ C}:
                for xr in cr:
                    X = {x, xr}
                    if x = xr:
                        X = {x}
                    if A_d(X, {c, cr}):
                        edge(x,xr,G)
    X = clique_o(G, |C|) or ∅
    for i in |X|:
        if X_i is negation:
            X_i = false
        else:
            X_i = true
    return X
```

Where

- `graph(`V`, `E`)` creates a graph with vertices V and edges E.
- `edge(`v_1`, `v_2`, `G`)` creates an edge between vertex v_1 and v_2 in graph G.
- `clique`$_o$`(`G`, `k`)` finds a clique in graph G of size k.

We construct a graph containing each element x in each clause c in the set of clauses C as vertices. We then compare each vertex with each other vertex, and if they satisfy A_d we create an edge between them. We find the clique of size $|C|$ and assign truth values to the variables X represented by the vertices, such that they evaluate to true. This set of truth assignments X is a solution to $A_d(X, C) = true$

1.2 B

In order to show that our algorithm is correct, we must show that a clique formed from our graph will in fact represent a set of variables X whose truth assignments are a solution to $A_d(X, C) = true$.

This is true because we draw edges based on A_d and we therefore know that the two vertices connected by an edge do not have conflicting truth assignments. Thus if we find a clique with a vertex from each clause, whom all have interconnected edges, the set of vertices in this clique will represent a set of non-conflicting truth assignments. Since we have $|C|$ clauses, the clique must be of size $|C|$.

We are guaranteed to draw edges creating such a clique if it exists, since we try each vertex with each other vertex in the graph.

1.3 C

The algorithm consists primarily of four nested loops. The outer most loop is repeated $|C|$ times, the second loop is repeated $|X|$ times, the third is repeated $|C|$ times and the inner most is repeated $|X|$ times. Thus we have approximately $|C| \times |X| \times |C| \times |X| = (|C| \times |X|)^2$. We know that $|C| + |X|$ is in fact the total length of our input n, and since we know that *clique* has $O(n^)$ running time, the total running time is $O(n^2) + O(n^2) = O(2n^2) = O(n^2)$.