Solution to Assignment 4

Thomas Feng *

April 16, 2003

1. *Solution:*
CNF: $(\bar{x}+y+\bar{z})\cdot(x+z)\cdot(\bar{y}+z)\cdot(\bar{x}+y+z)\cdot(x+\bar{y})$
Equivalent clique problem:

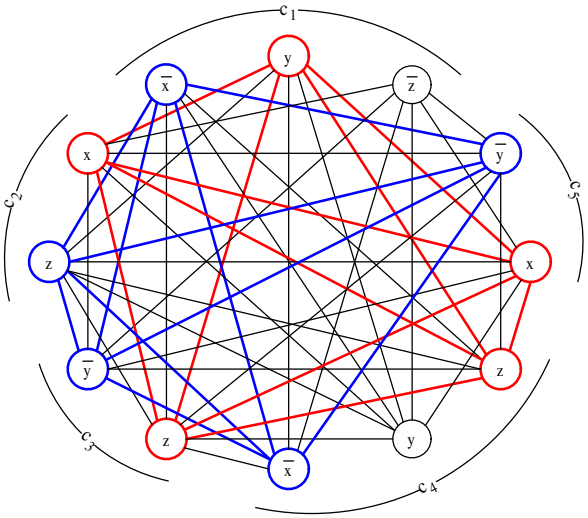
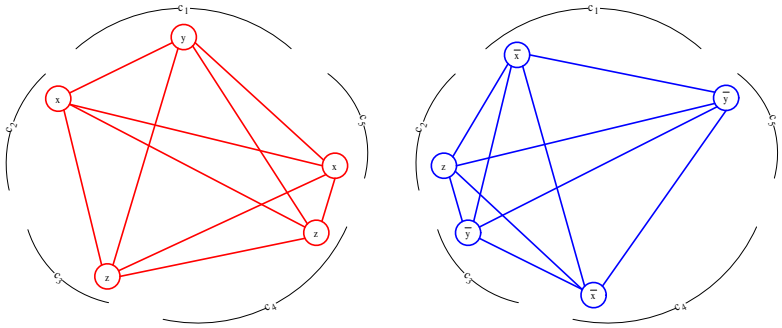


Figure 1: Equivalent clique

The red nodes and blue nodes represent two cliques, corresponding to two truth assignments. They are also shown in Figure 2.



a. $x=T$, $y=T$, $z=T$ b. $x=F$, $y=F$, $z=T$

Figure 2: Two cliques taken from Figure 1

*Email: thomas@email.com.cn
Homepage: http://moncs.cs.mcgill.ca/people/tfeng/

The two cliques correspond to these truth assignments of the CNF:

$$(F+T+F)\cdot(T+T)\cdot(F+T)\cdot(F+T+T)\cdot(T+F)=T \quad (1)$$
$$(T+F+F)\cdot(F+T)\cdot(T+T)\cdot(T+F+T)\cdot(F+T)=T \quad (2)$$

2. *Proof:*
Reduce the Subset Sum problem to Knapsack problem.

- Knapsack problem

  Instance: *Non-negative weights a_1, a_2, $\cdots$, a_n, b, and profits c_1, c_2, $\cdots$, c_n, k.*

  Question: *Is there a subset of weights with total weight at most b, such that the corresponding profit is at least k?*

- Subset Sum problem

  Instance: *Non-negative integer numbers s_1, s_2, $\cdots$, s_n and t.*

  Question: *Is there a subset of these numbers with a total sum t?*

The first step is to prove Knapsack is in the NP class. (This is very important, but some students forget.)

Given an input set, it is very easy to check if the total weight is at most b and if the corresponding profit is at least k. It takes only linear time to add the weights and profits of all the goods to find the *true/false* result.

The second is to prove a certain problem, which is already known to be NP-Complete, can be reduced to Knapsack problem in polynomial time. We can choose any of the NP-Complete problem we have learned. Because we already know all the problems in the NP class can be reduced to the chosen problem, say Subset Sum, we know all these problems can also be reduced to Knapsack problem.

It is very easy to reduce an instance of Subset Sum problem to an instance of Knapsack problem. We just create such a Knapsack problem that

$$\begin{cases} a_i = c_i = s_i \\ b = k = t \end{cases}$$

The *Yes/No* answer to the new problem corresponds to the same answer to the original problem. Now prove:

The following deduction implies the new problem is equivalent to the original problem

$$\begin{cases} \sum_{i\in S} a_i \le b \Longleftrightarrow \sum_{i\in S} s_i \le t \\ \sum_{i\in S} c_i \ge k \Longleftrightarrow \sum_{i\in S} s_i \ge t \end{cases} \Longleftrightarrow \sum_{i\in S} s_i = t$$

Suppose we have a *Yes* answer to the new problem, it means we can find such a subset $S \subseteq [1, 2, \cdots, n]$ that satisfies the left part of the deduction. Then this subset S is also a solution to the right part. So we must also have a *Yes* answer to the original problem.

Conversely, suppose we have a *No* answer, it means there is no subset S that satisties the left part. So, of course, the answer to the original problem must also be *No*. ■

3. *Solution:*
We cannot draw this conclusion. To disprove it, we simply find a counter-example. I.e., the Independent Set problem is to decide if there is an independent set of size r in a graph. it is NP-Complete for both general graphs and also bipartite graphs. We can prove it in the following way:

- It is obvious that, given a set of vertices, we can decide if they are independent and if the size is r in polynomial time. So this problem is in NP class.

- We can reduce a clique problem to an independent set problem. Suppose we have a clique problem for general graph. We already know it is NP-Complete.

  We can create an independent set problem for G', which is the complement of G in the clique problem. Then deciding if G has a clique of size r is equivalent to deciding if G' has an independent set of size r, so the clique problem can always be reduced to the independent set problem in polynomial time. So we proved the independent set problem is NP-Complete for general graphs.

  To prove it is also NP-Complete for bipartite graphs, we consider a general graph G whose complement is a bipartite graph. G itself is not a bipartite graph (unless it has less than 5 nodes). The clique problem for G is NP-Complete. Because the clique problem for G can be reduced to the independent set problem for G', the independent set problem for G', which is a bipartite graph, is thus NP-Complete.

Now that we have a counter-example: independent set problem is NP-Complete for general graphs, but for bipartite graphs, it is also NP-Complete.

Hint: *Sometimes the structure of bipartite graphs simplifies the problem, so it becomes polynomial even if it is NP-Complete for general graphs. But if the structure has nothing to do with the nature of the problem, it is still NP-Complete. This consideration is only a key to find the counter-example, but not the answer to this question. You must give a counter-example, or disprove the claim formally (this is not required).*

4. *Solution:*
We can reduce Subset Sum problem to a new problem in non-polynomial time. The new problem is polynomially solvable.

- Subset Sum problem (NP-Complete)

  Instance: *Non-negative integer numbers s_1, s_2, $\cdots$, s_n and t.*

  Question: *Is there a subset of these numbers with a total sum t?*

- Integer Searching (Linear)

  Instance: *Non-negative integer numbers a_1, a_2, $\cdots$, a_n and r.*

  Question: *Is there an integer which equals to r?*

We can use a non-polynomial reduction to reduce a Subset Sum problem to the Integer Searching: enumerate all the subsets in the Subset Sum problem and compute the sum of each of these sets. The sums of these sets are the elements in a set which is used for integer searching. Also let r be t. This reduction is exponential, because for a set of n elements, the number of subsets is 2^n. However, the resulting integer searching problem is just linear.

Hint: *To answer this question easily, one must know that the input set of the new problem can be exponentially larger than the original problem. The reduction to generate this large input set is thus non-polynomial. The new problem itself may be very trival. It can be polynomial or even linear to the transformed input set (but of course exponential to the original input set.*