Computational complexity – Homework

Marc CHEVALIER

October 26, 2014

1 NP-Hardness

1.1 Halting problem

Let be φ an instance of SAT problem. We denote by n the number of variables.

Let be M a TURING machine which tests in a cycle all the 2^n possible assignations of the previous formula : when M has tested all assignations, it starts again. This machine halts if and only if φ is satisfiable. This reduction is polynomial, therefore $SAT \leqslant_p$ HALT, ie. HALT is NP-hard since SAT is NP-hard.

HALT is not NP-complete otherwise it was decidable by a TURING-machine, but HALT is unsatisfiable.

1.2 $TQBF$

All instance of SAT problem is an instance of $TQBF$. Without transformation, we have a polynomial reduction, ie. $SAT \leqslant_p TQBF$ so $TQBF$ is NP-hard.

This problem is known for being PSPACE-complete. It's not NP-complete.

1.3 $NAE - 3 - SAT$

Let be φ an instance of $NAE - 3 - SAT$.

$$\varphi = \bigwedge_{i=1}^{n} (x_{i,1} \vee x_{i,2} \vee x_{i,3})$$

We will describe the "not all equal" condition is term of formula.

$$\bigwedge_{i=1}^{n} \neg(x_{i,1} \wedge x_{i,2} \wedge x_{i,3}) \wedge \neg(\neg x_{i,1} \wedge \neg x_{i,2} \wedge \neg x_{i,3})$$

using DE MORGAN's law:

$$\psi := \bigwedge_{i=1}^{n} (\neg x_{i,1} \vee \neg x_{i,2} \vee \neg x_{i,3}) \wedge (x_{i,1} \vee x_{i,2} \vee x_{i,3})$$

$|\psi| \sim 2|\varphi| \Rightarrow |\psi| = \mathcal{O}(|\varphi|)$.

Let $\varpi := \varphi \wedge \psi$. ϖ is an instance of SAT and the reduction is polynomial. If there is an solution to the SAT problem ξ, then φ is satisfied and, thanks to ψ, the "not all equal" condition is true.

Reciprocally, if there is a solution of the $NAE-3-SAT$ problem φ, then this assignation makes ξ true.

Consequently, $SAT \leqslant_p NAE-3-SAT$ and $NAE-3-SAT$ is NP-hard.

Moreover, $NAE-3-SAT$ is clearly NP: a valid assignation is a sufficient witness. We can check in polynomial time if this assignation makes the formula true and if the "not all equal" condition is satisfied. Then $NAE-3-SAT$ is NP-complete.

1.4 MAXCUT

Let F be an instance of $NAE-3-SAT$

$$F = \bigwedge_{i=1}^{m} C_i$$

We produce a graph $G=(V,E)$ which has a vertex for each literal of F. There is a edge between two vertices if there is a clause which contains this two literals. So each clause is described by a triangle. Moreover, we add $|F|_{x_i}$ (the number of occurrences of x_i in F) edges between x_i and $\neg x_i$. The size of the cut we search is at least $5m$.

If we have an assignment of the $NAE-3-SAT$, we take the vertices which are true in S and the other in $\overline{S}$. So, we have $2m$ from the triangles due to the clauses and $3m$ from the edges between all pair $(x_i, \neg x_i)$.

Reciprocally, if we have a cut of size $\geqslant 5m$.

If we have no pair $(x_i, \neg x_i)$ on the same size, we have a valid assignment.

If there is a such pair, we can move one of them on the opposite side without decreasing the size of the cut. Let n_i the number of edges between x_i and $\neg x_i$. We note a the number of edges which x_i is an extremity and which the other is in the opposite side. We note b the number of edges between b and a vertex of the opposite size. We know that $a+b \leqslant 2n_i$. If we move x_i in the opposite size, the cut gains $n_i - a$ edges. If $\neg x_i$ go to the opposite side, it gains $n_i - b$. $\max(n_i - a, n_i - b) \geqslant 0$, so we can move one of these vertices to the opposite side without decreasing the size of the cut. We redo this transformation until we reach a cut of the first case (at most m times).

We proved that $NAE-3-SAT \leqslant_p$ MAXCUT.

Moreover, MAXCUT is in NP. Indeed, a witness is the list of the vertices of S (or $\overline{S}$). The size is actually polynomial with respect of the size of G and we can check the solution in a polynomial time : we check easily that the cut has a size $\geqslant k$ in a quadratic time.

So, MAXCUT $\in$ NP.

2 Reductions

3 Difference of NP problems

Proposition 1. EXACTINDSET *is in DP.*

Proof. Let A be the set of all pairs (G,k) such that G has an independent set of size at least k, and let B be the set of all pairs (G,k) such G has a independent set of size at least $k+1$. Then EXACTINDSET $= A \setminus B$ and A is in NP and B is in NP. Hence by definition of DP, EXACTINDSET is in DP. □

Proposition 2. $\forall L \in DP$, *L is polynomial-time reducible to* EXACTINDSET.

Proof.

Lemma 3. INDSET $\geqslant_p 3 - SAT$

Proof. Suppose we have an instance F of $3 - SAT$ problem where $F = \bigwedge_{i=1}^{m} C_i$ where C_i is the disjunction of 3 variables. We note $x_1, \ldots, x_n$ the variables. We create the graph G as follows:

- For each variable in each clause, create a vertex, which we will label with the name of the variable. Therefore there may be multiple vertexs with the label x_i or $\neg x_i$, if these variables appear in multiple clauses.
- For each clause, add an edge between the three vertices corresponding the variables from that clause.
- For all i, add an edge between every pair of vertexs with one is labelled with x_i and the other labelled with x_i.

There is a independent set of size m in G if and only if F is satisfiable. ☐

We note that this reduction from $3 - SAT$ to INDSET took an instance φ of $3 - SAT$ consisting of m clauses each of three literals and produced a graph G_φ with $3m$ vertexs such that if φ is satisfiable then the largest independent set in G has m vertexs, and if G is unsatisfiable then the largest independent set of G has at most $m - 1$ vertices.

Now suppose that A is in DP. We want to show that $A \leqslant_p$ EXACTINDSET. By definition of DP, $A = L_1 \setminus L_2$ for $(L_1, L_2) \in NP^2$. Since $3 - SAT$ is NP-complete, there are polytime functions f_1 , f_2 such that for $i = 1$, 2 and for all $x \in \{0,1\}^\star$ we have $x \in L_i \Leftrightarrow f_i(x) \in 3SAT$. Hence for each fixed $x \in \{0,1\}^\star$, setting $\varphi_i = f_i(x)$, we have $x \in L_i \Leftrightarrow \varphi_i$ is satisfiable Thus from the above reduction to INDSET, there is a polytime function which takes x to a pair of graphs G_1, G_2 such that if m_i is the number of clauses in φ_i, then for $i = 1$, 2, no independent set in G_i has more than m_i vertices, and $x \in L_i \Leftrightarrow$ the largest independent set in G_i has size m_i.

Now we use the notation$G \sqcup H$ for the disjoint union of graphs G and H. That is, the vertices in $G \sqcup H$ are the disjoint union of those in G and H, and similarly for the edges. Now let $G_1' = G_1 \sqcup G_1$ Then a maximum independent set in G_1' is the union of maximum independent sets in the two copies of G_1. Thus $x \in L_1 \Rightarrow$ maximum independent set of G_1' is $2m_1$ and $x \in \overline{L_1} \Rightarrow$ maximum independent set of G_1' is $\leqslant 2m_1 - 2$. Now define G_2' so that its vertices are those of G_2 together with $m_2 - 1$ new vertices, and the edges consist of those of G_2 together with an edge from each of the new vertices to every vertex of G_2. Then we have designed $G'2$ so that no independent set can contain both vertices of G_2 and new vertices, so $x \in L_2 \Rightarrow$ maximum independent set of G_2' is m_2, $x \in \overline{L_2} \Rightarrow$ maximum independent set of G_2' is $m_2 - 1$. Now let $G_3 = G_1' \sqcup G_2'$ and let $k = 2m_1 + m_2 - 1$. To finish the proof that $A \leqslant_p DP$, it suffices to show that

Lemma 4.

$$x \in A \Leftrightarrow (G_3, k) \in DP$$

($\Rightarrow$): Suppose $x \in A$. Then by (2)$x \in L_1 \cap L_2$, so by (3) and (6) we conclude the maximum independent set of $G_3 = G_1' \sqcup G_2'$ is $2m_1 + m_2 - 1 = k$.

($\Leftarrow$): Suppose $x \notin A$. There are three cases:

- $x \in L_1 \cap L_2 \Rightarrow maxindset(G_3) = 2m_1 + m_2 > k$.
- $x \in L_1 \cap L_2 \Rightarrow maxindset(G_3) \leqslant 2m_1 + m_2 - 2 < k$
- $x \in L_1 \cap L_2 \Rightarrow maxindset(G_2) = 2m_1 + m_2 - 3 < k$

☐

Theorem 5. EXACTINDSET *is DP-complete.*

Proof. EXACTINDSET is in DP (proposition 1) and it is DP-hard (proposition 2). ☐

4 Classes with exponential resources

5 Downward self-reducibility

6 Space hierarchy theorem

7 Polynomial hierarchy