	M2-RIT Recherche Opérationnelle **Graphes et Réseaux**	Année 2014 - 2015 Semestre 1

TD 4: Fixed-parameter algorithms

1 Complexity theory

The outline of complexity theory in the course was a rather rough sketch. We give here some more indications.

The standard NP-complete problem is **SAT**, the problem of finding a satifying assignment for an arbitrary set of clauses. NP-hardness of **SAT** can be proved (Cook's theorem, 1971) by reducing any non-deterministic polynomial time computation to a set of clauses in **SAT**.

The problem **3-SAT** is the problem of finding a satifying assignment for a set of clauses, where each clause contains exactly 3 literals.

Exercice 1 We show that **3-SAT** is indeed NP-complete, as claimed in the course, by showing that $\mathbf{3-SAT} \leq^p \mathbf{SAT}$ (which is trivial) and $\mathbf{SAT} \leq^p \mathbf{3-SAT}$, by the following procedure:

Given a set $\mathcal{S}$ of clauses $\{\mathcal{C}_1 \dots \mathcal{C}_k\}$, where each $\mathcal{C}_\rangle$ has the form $\{\ell_{i,1} \dots \ell_{i,m}\}$.

1. Eliminate from $\mathcal{S}$ all clauses $\mathcal{C}_\rangle$ which are of the form $\{\}$ or $\{\{\}\}$ (how?).
2. Eliminate from $\mathcal{S}$ all one-element clauses $\{\ell_{i,1}\}$ (how?).
3. Replace in $\mathcal{S}$ all two-element clauses $\{\ell_{i,1}, \ell_{i,2}\}$ by two clauses $\{\ell_{i,1}, \ell_{i,2}, h\}$ and $\{\ell_{i,1}, \ell_{i,2}, \neg h\}$, where h is a new variable (why does this preserve satisfiability?)
4. Replace in $\mathcal{S}$ all m-element clauses (with $m > 3$) recursively according to the following schema:

   $\{\ell_{i,1}, \ell_{i,2}, \ell_{i,3} \dots \ell_{i,m}\}$ is replaced by the two clauses $\{\ell_{i,1}, \ell_{i,2}, h\}$ and $\{\neg h, \ell_{i,3} \dots \ell_{i,m}\}$, where h is a new variable.

   This gives $m - 2$ clauses of size 3 (preservation of satisfiability?)

Exercice 2 Show that any formula ϕ (not necessarily in clausal form) can be transformed to an equivalent clausal form with a polynomial time transformation.

1. Show that the classical procedure of applying distributivity laws such as $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ is not polynomial.
2. Define a translation based on the idea of translating a formula of the form $a \vee (b \wedge c)$ to $((t \leftrightarrow (b \wedge c)) \wedge (a \vee t))$, where t is a new variable. Define the details of this transformation, show that it preserves satisfiability and can be carried out in polynomial time.
3. Adapt this transformation so as to produce clauses of size 3, thus directly providing a reduction from the satisfiability of arbitrary formulas to **3-SAT**.

Exercice 3 Apply the algorithm for finding a valuation for **2-SAT** to the formulas:

1. $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$
2. $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$

Exercice 4 The **2-SAT** satsfiability checking procedure constructs a completely symmetric graph. Would it be possible to simplify the construction of the graph by only inserting an arc $(\neg \ell_i, \ell_j)$ (and not $(\neg \ell_j, \ell_i)$) for each clause $\{\ell_i, \ell_j\}$?

Exercice 5 Show that the **2-SAT** checking algorithm (also known as the Aspvall-Plass-Tarjan algorithm [1]) has the following properties.

Given a set of clauses $\mathcal{S}$ and the graph G constructed from $\mathcal{S}$.

- if there is a variable x with a cycle $x \to \cdots \to \neg x$ in G, then $\mathcal{S}$ is not satisfiable.
- if there is no variable x with a cycle $x \to \cdots \to \neg x$ in G, then the algorithm constructs a satisfying assignment of $\mathcal{S}$.

Exercice 6 Take the clause set $\mathcal{S} = \{\{u_1, \neg u_3, \neg u_4\}, \{\neg u_1, u_2, \neg u_4\}\}$ and its translation to a graph seen in the course.

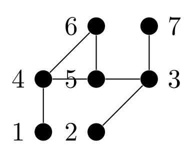
1. Take the vertex cover $\{\neg u_1, \neg u_2, u_3, \neg u_4, a_1^1, a_1^2, a_2^2, a_2^3\}$. Show that it corresponds to a satisfying truth assignment of the variables of the clause.
2. Given the satisfying assingment $(u_1 \mapsto \top, u_2 \mapsto \top)$, construct a vertex cover of size $k = |Var| + 2|\mathcal{S}| = 8$ corresponding to this assignment.

Exercice 7 To prove the correctness of the reduction of **3-SAT** to **VC**, show that

1. Every vertex cover of size $k = |Var| + 2|\mathcal{S}|$ of the graph gives rise to a truth assignment of the variables of the clause set.
2. Every truth assignment of the clause set yields a vertex cover of the graph.

2 FP Complexity

Exercice 8 Apply the p-**VC** algorithm to the following graph:



to show that the graph

- has no vertex cover of size 2
- has a vertex cover of size 3

[1] http://www.math.ucsd.edu/~sbuss/CourseWeb/Math268_2007WS/2SAT.pdf