Advanced algorithms

Exercise sheet #4 (Solutions) – Parametric complexity (continued)

October 19, 2022

Exercice 1 (3-Coloring and treewidth)**.** The 3-Coloring problem is the following:

Input: A graph G;
Question: Does the vertices of G admit a coloring with 3 colors such that no two adjacent vertices have the same color?

Show that this problem is FPT with respect to the treewidth of the input. In order to do that, give an algorithm that takes as input a graph G together with a *nice* tree decomposition of width k and m nodes and that computes a 3-coloring (or ensures that there is no such coloring) in time $O(m \cdot k \cdot 3^k)$.

Solution — Note that given any tree decomposition of G it is easy to get a decomposition in nice *form with same width. Such a decomposition is way easier to handle.*

We consider a nice *tree decomposition $\mathcal{T}$ of G that we root in one arbitrary node R. Then, there are four types of nodes:*

- *Addition (Only one child)*
- *Deletion (Only one child)*
- *Duplication (Two children)*
- *Leaves (0 child)*

Each node t of $\mathcal{T}$ is associated with a subset V_t of the vertices of G, that we call a bag. Moreover, we consider the subgraph G_t of G induced by the union of all V_s, where s is in the subtree rooted at t (We only keep those vertices and the edges that join them).

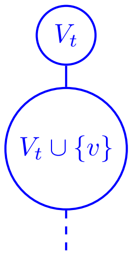
We will use a bottom-up approach. The idea is to solve a subproblem corresponding to each subtree, beginning by the problem induced by the leaves:

Given a node t associated with its bag $V_t = \{x_1, \dots, x_i\}$, the subproblem will be to find the set C_t of partial 3-colorings of the vertices in the bag V_t that can be extended to a 3-coloring of the subgraph G_t. More formally, the answer to this subproblem can be represented as a boolean table $B(t)$ of 3^i entries corresponding to all maps $V_t \mapsto \{1, 2, 3\}$. For any such map c, the entries corresponding to c will be TRUE if c extends to a 3-coloring of G_t and FALSE otherwise.

Therefore, the graph G has a 3-coloring iif $B(R)$ is not entirely FALSE.

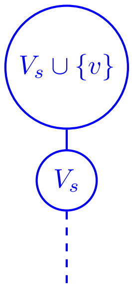
Let's distinguish the four cases:

1. Leaf. *Then V_t is a singleton $\{x\}$, and for any coloring of x, $B(t,x)$ is TRUE. The computation complexity is $O(1)$.*
2. Deletion node.



Then t has a single child s and $V_s = V_t \cup \{v\}$ for some vertex v not in V_t. (So that v is deleted when we go up the tree.) We observe that $G_t = G_s$ and it follows easily that $C_t = \{\phi|_{V_t} \mid \phi \in C_s\}$. When we deal with t we already delt with s, therefore the computation complexity is $O(\#C_s) = O(3^k)$, where k is the treewidth.

3. Addition node.

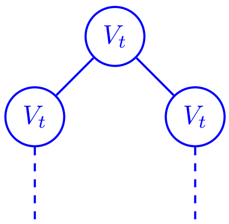


Then t has a single child s and $V_t = V_s \cup \{v\}$ for some vertex v not in V_s. (So that v is added when we go up the tree.) Basic properties of tree decompositions ensure that v is not G_s and that any vertex in G_s that is adjacent to v is in V_t.
When we deal with t, we already know the partial coloring of V_s that extend to a coloring of G_s. To decide whether such a coloring $\phi_{|V_s}$ can be extended in a coloring of G_t, we only need to exclude the colors given by $\phi_{|V_s}$ to neighbors of v in V_t.
So we have

$$C_t = \left\{\phi \colon \phi_{|V_s} \in C_s \text{ and } \phi(v) \neq \phi(w) \text{ if } (v,w) \in E\right\}.$$

The computation complexity is $O(k \cdot 3^k)$.

4. Duplication node.



Then t has two children r and s and $V_t = V_r = V_s$. However, the graphs G_r and G_s are in general dferent and more precisely the intersection of those two graphs are reduced to vertices in V_t. Two 3-colorings of G_r and G_s extend to a coloring of G_t iff they agree on V_t. It follows easily that $C_t = C_r \cap C_s$, and $B(t) = B(r) \wedge B(s)$. The computation complexity is $O(3^k)$.

The total complexity is $O(m \cdot k \cdot 3^k)$ where m is the size of $\mathcal{T}$.

Exercice 2 (Quadratic kernel for MaxSat)**.** The problem MaxSat is the following:

Input: A SAT formula given in CNF with m clauses; an integer k;
Question: Is there an assignment of the variables that satisfies at least k clauses?

(a) Show that the problem always has a solution if $k \leqslant \frac{m}{2}$.

Solution — Let p (resp. q) be the number of clauses satisfied by the assignment of all variables to TRUE (resp. FALSE). It is clear that $p + q \geqslant m$, so $p \geqslant k$ or $q \geqslant k$.

(b) Show an instance can always be reduced in polynomial time to another where no clause contains more than k literals.

Solution — Assume that the input is $(\Phi \wedge L, k)$, where L is a clause with $\geqslant k$ literals. Then this instance is equivalent to $(\Phi, k-1)$. One implication is trivial. For the other, consider an assignment of the variable that satisfies $k-1$ clauses in Φ. We can pick $k-1$ variables whose assignment is enough to satisfy these $k-1$ clauses. Then we pick a variable in L, different from the previous ones, and assign it in order to satisfy L.

(c) Show that every instance of MaxSat has a kernel of size $O(k^2)$.

Solution — By the previous question, we may assume that no clause contains more than k literals. By the first question, we may assume that the total number of clauses is $\leqslant 2k$. So we obtain a kernel of size $O(k^2)$.

Exercice 3 (Paths and colorings)**.** We consider the problem PartialHamiltonian:

Input: $G = (V, E)$ a graph; k an integer;
Output: Is there a path in G visiting exactly $\geqslant k$ vertices and no vertex more than once?

We will study in particular the case where $k = O(\log n)$, where n is the number of vertices of G.

(a) Give a naive algorithm. Is it polynomial in n when $k = O(\log n)$?

Solution — By recursive exploration we can test the n neighbors of each vertex, and the next $n-1$ neighbors, and so on and so forth. This can be done in time $n(n-1)\dots(n-k-1) \sim n^k$. This is not polynomial, even with $k = O(\log n)$.

(b) Let $C : V \to \{1, \dots, k\}$ be a coloring of G with k colors (without constraints on the colors of adjacent vertices). A path is totally multicolor if every vertex has a different color.

Show how to decide the existence a totally multicolor path of length k in time $O(2^k \cdot n^2)$. *Hint: dynamic programming.*

Solution — For a subset $S \subseteq \{1, \dots, k\}$ and a vertex v, let $T_{S,v}$ be the number of totally multicolor paths with colors in S, of length $\#S$ and ending at v. Then $T_{\{c\},v} = 1$ if $C(v) = c$ and 0 otherwise. And by induction on $\#S$, we compute further that $T_{S,v} = 0$ if $C(v) \notin S$ and

$$T_{S,v} = \sum_{\substack{w \text{ neighbor of } v \\ \text{s.t. } C(w) \in S}} T_{S \setminus \{C(v)\}, w}$$

otherwise. This gives a $O(2^k \cdot n^2)$ algorithm: $O(2^k \cdot n)$ values to compute and each application of the recursive formula costs $O(n)$. (We just want to check that some $T_{\{1,\dots,k\},v}$ is > 0 so we don't need to actually compute the big integers).

(c) Give a probabilistic algorithm to solve PartialHamiltonian. Is the complexity polynomial when $k = O(\log n)$?

Solution — We pick a k-coloring at random and decide the existence of a totally multicolor path of length k. If there is one, then PartialHamiltonian has a solution and we return TRUE*. We repeat m times this procedure, with m to be fixed later, and return* FALSE *if none of the interations is successful.*

- *If* PartialHamiltonian *has no solution, then this procedure always returns* FALSE.
- *On the contrary, if there exists a solution P, then after the choice of a random k-coloring, P will be totally multicolor with probability at least $\frac{k!}{k^k} \sim \sqrt{2\pi k}e^{-k}$.*

  If we repeat m times this procedure, the probability

  So we choose $m = \lfloor (\pi k)^{\frac{1}{2}} e^k \rfloor$. (Observe that $1-(1-p)^{\lfloor \frac{1}{p} \rfloor} \geqslant \frac{1}{2}$ for any $p \in (0,1)$.) Then our algorithm returns always FALSE *if the instance has no solution and return* TRUE *with probability $\geqslant \frac{1}{2}$ if the instance has a solution.*

Exercice 4 (HamiltonianCycle and treewidth)**.** The HamiltonianCycle problem is the following:

Input: A graph G;
Question: Is there a cycle in G that visits all the vertices exactly once ?

Show that HamiltonianCycle is FPT with respect to the treewidth.

Solution — Use the same approach as for the 3-coloring. Be careful in choosing the subproblem associated to a given node t of the nice tree decomposition so that you can glue together partial solutions.

One good choice is the following: To each triplet (M, U, L) where U and L are disjoint subsets of V_t and M is a perfect matching of $V_t \setminus (U \bigsqcup L)$, decide if there exists a set of disjoint paths in G_t such that the endpoints of the paths are given by M, the vertices of U are internal to the paths and the vertices in L are free. The table of a node can then be indexed by the perfect matching with $2j$ elements (where $0 \leq j \leq \lfloor k/2 \lfloor$). For each of these matchings, we need to find the state of at most $k-2j$ nodes (free nodes or nodes belonging to a path). This yields a table of at most $O(4^k 2^k)$ entries.

Try to describe the 4 cases.