Lemme de Courcelle

NATTAF Margaux

Théorème. *Toutes propriété de graphe exprimable en MSOL (logique monadique du second ordre) peut être testée en temps $f(k) \cdot n$ pour un graphe avec une largeur arborescente k.*

La réciproque n'est pas vraie en général. Par exemple, on ne peut pas exprimer le problème CYCLE HAMILTONIEN en MSOL, mais il existe un algorithme paramétré pour le résoudre.

Limites du théorème de Courcelle

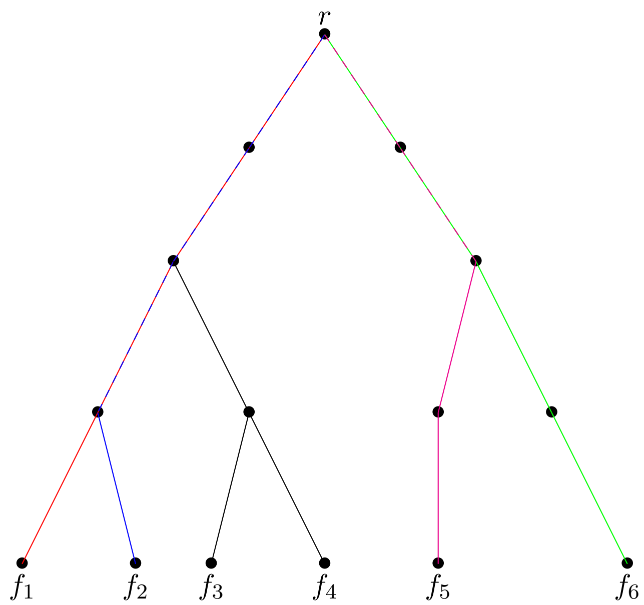
La première limite de ce théorème est que, malgré le fait qu'il prouve l'existence d'un algorithme FPT pour un problème NP-complet, il ne donne pas, en général, l'algorithme en lui-même. De plus, le fonction $f(k)$ dépend de la formule ϕ et la hauteur de la tour d'exponentielle dépend des alternances de $\forall$ et $\exists$. En conclusion, le théorème de Courcelle est un outil de preuve d'existence d'un algorithme FPT.

k-LONGEST PATH

Existence d'un algorithme FPT

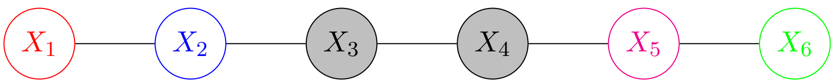
Proposition. *k-LONGEST PATH est FPT.*

Démonstration. Sachant que l'existence d'un chemin de longueur k est exprimable en MSOL, montrons que k-LONGEST PATH est FPT. Le but est de montrer que si la $tw(G)$ est trop grande alors, soit l'instance est négative (G contient une obstruction), soit l'instance peut être réduite (on diminue $tw(G)$).
Soit T un arbre de parcours en profondeur de G.



On a alors deux cas :

- Soit il existe un chemin de longueur au moins k entre deux feuilles de T et on a une solution en temps polynomial.
- Soit la hauteur de T est inférieure à k et alors $tw(G) < k$. En effet, si la hauteur de T est inférieure à k, alors on construit la treewidth de G comme suit : Soit r la racine de T et $\{f_1, ..., f_l\}$ ses feuilles. Pour chaque feuille f_i, on range dans un paquet X_i tous les sommets sur le chemin de r à f_i et on relie X_i à X_{i+1}, $\forall i \in \{1, ...n-1\}$ (sauf X_1 et X_l).



Pour montrer que c'est bien une treewidth de G, montrons les trois propriétés suivantes :

- la couverture des sommets
- la couverture des arêtes
- la consistance

Couverture des sommets Soit x un sommet de G, comme T est un arbre couvrant, $x \in V(T)$. De plus, tous les sommets de T appartient à un X_i.

Couverture des arêtes Si il existe une arête (u, v) non couverte par la treewidth, alors il existe $i \neq j$ avec $u \in X_i$ et $v \in X_j$. Supposons, sans perte de généralité que $i < j$. Alors, soit b_i (respectivement b_j) la branche correspondant à la feuille f_i (respectivement f_j), $u \in b_i$ et $v \in b_j$. Or, comme T est un arbre de parcours en profondeur, et $v \in N(u)$, v aurait du être explorer avant de remonter.

Consistance Soit $v \in G$ tel que $\exists i \neq j$avec $v \in X_i \cap X_j$. Alors, par construction, il appartient à $b_i \cap b_j$. Alors, il appartient à $\bigcap_{k=i}^{j} b_k$ et donc $v \in \bigcap_{k=i}^{j} X_k$.

De plus, comme chaque branche est le longueur au plus k, on a $tw(G) < k$.

□

Programmation dynamique

Pour la programmation dynamique, supposons, sans perte de généralité, que la décomposition arborescente de G est simple. Si ce n'est pas le cas, comme il existe un algorithme polynomiale pour transformer toute décomposition en une décomposition simple, nous appliquons cet algorithme sur le graphe pour le rendre simple.

Nous avons alors 4 types de nœuds à traiter :
- les nœuds feuille
- les nœuds ajout
- les nœuds suppression
- les nœuds fusion

nœud feuille Les nœuds feuille ne contenant qu'un seul sommet, un plus long chemin contient juste le sommet contenu dans le nœud.

nœud ajout Supposons que X_i est le père de X_j et $X_i = X_j \cup \{v\}$. Comme T est une décomposition arborescente, on sait que les seules arêtes de G incidentes à un sommet de X_i sont contenues dans X_i. Donc, si l'ajout du sommet v augmente la taille d'un plus long chemin, c'est qu'à l'étape précédente, un sommet de X_j était la fin d'un plus long chemin.

nœud suppression Supposons que X_i est le père de X_j et $X_i = X_j \setminus \{v\}$. Comme le nœud X_j contient tous les sommets de X_i et le sommet v, un plus long chemin contenant un sous ensemble de sommets de X_j a déjà été calculé à l'étape précédente. Donc pour chaque sous ensemble $S \in X_i$, la taille d'un plus long chemin est le max entre la taille du plus long chemin de $S \cup \{v\}$ et la taille du plus long chemin de S à l'étape précédente.

nœud fusion Supposons que X_i est le père de X_{j_1} et de X_{j_2} avec $X_i = X_{j_1} = X_{j_2}$. Observons d'abord que si $v \in V_{j_1}$ et $w \in V_{j_2}$, alors $\{v, w\} \notin X_i \Rightarrow (v, w) \notin E$.
En d'autres termes, pour $v \in V_{j_1}$ et $w \in V_{j_2}$, $(v, w) \in E$ si v ou w appartient à X_i. Or, si $v \in X_i$ (respectivement $w \in X_i$) alors $\{v, w\} \in V_{j_2}$ (respectivement $\{v, w\} \in V_{j_1}$).