# TD SSP

Denis Beauget

March 2020

## 1 Exercice 1

Montrons que si il existe un algorithme polynomial pour le problème d'optimisation alors on a un algorithme polynomial pour le problème de décision.

Soit AlgoDecision() un algorithme qui prend M un ensemble de mot et k un entier strictement supérieur à 0, qui renvoie un booléen.

Avons nous une superséquence S de M de longueur ¡= k telle que chaque mot de M est un sous$_{m}otdelasuperséquenceS$?

On en déduit un AlgoOptimisation() qui prend en entrée M et renvoie la taille de la plus petite superséquence S. AlgoOptimisation(M) {

unsigned int k = 1;
Tant que AlgoDecision(M,k) == 0 {
k ¡- k++:
}
fin tant que

renvoyer k;
}
finalement on en déduit que le problème d'optimisation est "au moins aussi difficile" que le problème de décision.

## 2 Exercice 2 )

L'algorithme Contient(s,m) vérifie si le mot m appartient à s en complexité polynomial On en déduit un algorithme qui étant donné une superchaine véirifie si elle satisfait le problème de décision.

Entrée : s superchaine et m un mot
Sortie : un booléen indiquant si m appartien à s.
Algo : {

Si $\|S\| \geq renvoyer0$
$PourtoutF(i)dansF\{$
$si!Contient(s, F)$
$renvoyer0$
$\}$
$renvoyer1\}$

# 3 Exercice 3 )

Prenons A,D,F et G les 4 sommets du graphe G on a bien une couverture minimale du graphe G (4 sommets couvrant tout le graphe). Donc Res = A,D,F,G

# 4 Exercice 4 )

Si on transforme une instance de vertex cover en instance de SSP alors on a montrer précedement que le problème SSP appartenait à NP, on a montrer qu'il existait un algorithme polynomial pour le problème SSP et si on se ramène à une réduction aux problème du Vertex Cover on en déduit qu'il existe un algorithme polynomial pour le problème du Vertex Cover également.

# 5 Exercice 5 )

V = A,B,C,D,E,F,G
F = (abab,baba),(abab,baba),(abab,baba),(abab,baba),(abab,baba),(abab,baba),
(abab,baba),(abab,baba),(abab,baba),(abab,baba)