

Compte rendu du TP01 : 2-SAT

1) Les membres du binôme :

- AMROUCHE Yanis
- AGUENI Farouk

2) Les choix d'implémentation :

Notre programme pour la résolution du problème 2-SAT se compose de trois classes :

➤ Vertex :

Cette classe est faite pour générer les sommets du graphe. Elle contient un attribut 'name' qui représente le nom du sommet, un constructeur pour obtenir une instance d'un sommet et une méthode d'affichage 'toString' affichant le nom du sommet.

➤ Graph :

Cette classe est faite pour manipuler le graphe d'implication grâce à l'implémentation des différentes méthodes de parcours en profondeur et de l'algorithme de Kosaraju.

Elle contient une structure de donnée 'Map' avec les sommets comme clés, la liste des voisins comme valeur (liste d'adjacence). Ainsi que des méthodes pour l'ajout des arcs, ajout de sommets et pour la récupération des voisins d'un sommet.

On a rajouté une méthode de parcours en profondeur 'depthFirstTraversal' pour tout les sommets du graphe (même les sommets non accessibles), 'depthFirstSearch' pour les sommets accessibles uniquement et 'DFSExplore' qui récupère les sommets en prenant en compte leurs fin de traitement (dates de fin).

Au final, on a procédé à l'implémentation de l'algorithme de Kosaraju qui se compose de deux étapes, on a réalisé la première étape par la méthode 'firstStepOfKosarajuAlgorithm' qui fait d'abord un parcours en profondeur et lorsque un sommet est fini d'être traité par le parcours en profondeur on va insérer ce sommet dans une liste et à la fin de la première partie lorsque on a fini tout les parcours en profondeur on va inverser la liste et c'est le résultat de l'étape 01. Et pour la deuxième

étape réaliser par la méthode 'stronglyConnectedComponents ' on va prendre en compte le graphe transposé et pour l'obtenir il suffit d'inverser les arcs du graphe d'implication puis on va faire un parcours en profondeur sur le graphe transposé dans l'ordre de la liste obtenu a la première étape et chaque arbre est une composante fortement connexe, en plus de la vérification des satisfiabilités.

➤ Main :

Dans cette classe on a réussi a effectué la lecture des données a travers les fichiers et en manipulant une liste ou on a stocké les clauses (arcs) puis en créant le graphe d'implication et le graphe transposé en appliquant la formule d'implication 2-SAT grâce a la méthode 'OppositeValue'.

3) Les difficultés rencontrées :

Pour conclure, suite aux différent testes effectuer sur les différents fichiers (formule-2-sat.txt, formule-pos.txt, formule-neg.txt, formule-conflict.txt) on a constaté que les testes sont passées avec succès et pour cela on pas de difficultés a citées.