Programme de colle période 1 : décidabilité, graphes

1 Programme

1.1 Décidabilité

- Problème de décision, problème d'optimisation
- Représentation des instances, encodage
- Modèle de calcul, algorithme, machine universelle
- Indécidabilité du problème de l'arrêt
- Réduction Turing, réduction many-one
- Semi-décidabilité, co-problèmes (complémentaires)

1.2 Graphes

- Graphes orientés et non orientés, vocabulaire
- Parcours de graphe, parcours en largeur, en profondeur
- Arbres couvrants, union-find, algorithme de Kruskal
- Composantes fortement connexes, parcours en profondeur préfixe et postfixe, ordre topologique
- Algorithme de Kosaraju, application à 2SAT
- Algorithme A*
- Couplages, théorème de Berge, calcul d'un couplage de cardinal maximum dans un graphe biparti

2 Questions de cours

1. Montrer que le problème de l'arrêt est indécidable.
2. En admettant la question 1, montrer que le problème co`ARRÊT` n'est pas semi-décidable.
3. Montrer que `ARRÊT` se réduit many-one à `ARRÊT`$_\forall$ (le problème qui prend en entrée une fonction f et demande si $f(x)$ termine pour tout argument x). En déduire une propriété sur `ARRÊT`$_\forall$.
4. Soit $G = (S, A)$ un graphe non orienté. Montrer que G est un arbre si et seulement si G est connexe et $|A| = |S| - 1$ si et seulement si G est acyclique et $|A| = |S| - 1$. On pourra admettre des résultats intermédiaires.
5. Dans le langage de programmation de votre choix, écrire les fonction `unir` et `trouver` permettant d'implémenter la structure Union-find, avec compression des arbres et équilibrage des fusions.
6. Donner sans justifier un invariant de boucle de l'algorithme de Kruskal. En utilisant cet invariant, montrer la correction de l'algorithme de Kruskal.
7. Soit $G = (S, A)$ un graphe orienté. Montrer que G est sans cycle si et seulement si pour toute arête (s, t), s apparaît après t dans l'ordre d'un parcours en profondeur postfixe.
8. Énoncer et montrer le théorème de Berge.