

Programme de colle - Semaine 10 (20 novembre)

La démonstration des énoncés marqués d'une étoile est exigible

1 Union-Find et algorithme de Kruskal

- Structure Union-Find : implémentation naïve favorisant l'opération FIND avec un tableau dans lequel chaque case i contient le représentant de l'élément numéro i . Complexités des opérations.
- Structure Union-Find : implémentation à l'aide d'une forêt favorisant l'opération UNION.
 1. Optimisation 1 : en plaçant systématiquement l'arbre de plus petite hauteur comme fils lors de l'opération union. **Dans ce cas, la hauteur de chaque arbre ne dépasse pas $\log_2(m)$ avec m la taille de l'arbre considéré (*)**
 2. Optimisation 2 : en compressant les chemins lors de l'opération FIND. La complexité amortie obtenue est hors programme.
- Notion d'arbre couvrant d'un graphe non orienté. Algorithme générique de construction d'un arbre couvrant : utilisation de Union-Find dans ce cadre.
- Arbre couvrant de poids minimal d'un graphe non orienté pondéré. Existence. **Algorithme de Kruskal** : savoir décrire l'algorithme en pseudo-code avec la structure Union-Find, savoir l'appliquer sur un exemple, la preuve n'est pas au programme de colle.

2 Automates finis

Les étudiants doivent savoir appliquer les constructions et algorithmes sur les automates.

- Automate fini déterministe (afd) : définition et calcul d'un afd, mots reconnus, langages acceptés
- Accessibilité et co-accessibilité; afd émondé
- Afd complet; complétion d'un afd
- Automate complémentaire
- Automate produit
- Classe des langages reconnaissables : stabilité par complémentaire, intersection et union finie (concaténation et étoile pas encore vu)
- Automate fini non déterministe (afnd) : définition et calcul d'un afnd, mots reconnus, langage accepté
- Automate fini non déterministe avec transitions spontanées (ε -afnd) : définition, ε -clôture, calcul d'un ε -afnd, mots reconnus, langage accepté, élimination des transitions spontanées sans déterminer l'automate (fait en cours sur un automate de Thomson)

- Déterminisation d'un afnd ou d'un ε -afnd (automate des parties) : les classes de reconnaissabilité sont les mêmes pour tous les types d'automates finis.
- Langages non reconnaissables : **lemme de l'étoile (*)**, les exemples $\{a^n b^n \mid n \in \mathbb{N}\}$ et $\{u \mid |u|_a = |u|_b\}$ ont été vus en cours.

3 Théorème de Kleene

- Théorème de Kleene : la classe des langages réguliers est la même que la classe des langages reconnaissables.
- Passage d'une expression régulière à un automate :
 - Automate normalisé. Normalisation d'un automate à l'aide de transitions instantanées.
 - Construction d'automates normalisés de Thomson : aucune formalisation précise de l'algorithme n'est exigible mais les étudiants doivent savoir appliquer la méthode pour construire un automate de Thomson en appliquant exclusivement les règles de construction présentées en cours.
 - Langage local : définition, savoir déterminer si un langage est local (ou non), automate de Glushkov reconnaissant un langage local.
 - Algorithme de Berry-Séthi : connaître les étapes de l'algorithme et savoir l'appliquer.
- Passage d'un automate à une expression régulière :
 - Automate généralisé avec des expressions régulières sur les transitions ne dénotant pas le langage vide.
 - Algorithme par élimination successive des états en partant d'un automate normalisé.
- Rappels sur les conséquences du théorème de Kleene sur les stabilités des langages réguliers/reconnaissables.