

Programme de colle 9 (13 novembre)

La démonstration des énoncés marqués d'une étoile est exigible

1 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 instantanées (ε -afnd) : définition, ε -clôture, calcul d'un ε -afnd, mots reconnus, langage accepté
- 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.

2 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.

3 Théorème de Kleene

- Langages locaux. Définition et calcul de $P(L)$, $D(L)$ et $T(L)$ pour un langage L quelconque. Définition de langage local. Exemples.
- Automate de Glushkov : construction de l'automate qui reconnaît un langage L local à partir de ses caractéristiques.