

Programme de colle - Semaine 9 (colle du 12 novembre)

La démonstration des énoncés marqués d'une étoile est exigible

1 Langages réguliers

- Mots : alphabet, mot sur un alphabet, concaténation, puissance d'un mot, structure de monoïde.
- Langages : opérations ensemblistes sur les langages (union, intersection, différence, complémentaire), concaténation de deux langages, étoile de Kleene d'un langage, **l'union d'une famille finie de langages réguliers est un langage régulier (*)**
- Langages réguliers (aussi appelés langages rationnels): définition inductive de la classe des langages réguliers, **tout langage fini est régulier (*)**
- Expressions régulières : définition inductive, langage dénoté par une expression régulière (notation $\mu(e)$)
- **Un langage est régulier si et seulement si il existe une expression régulière qui le dénote (*)** [NB : interroger sur un seul sens uniquement]
- Expressions régulières étendues : aucune connaissance spécifique n'est exigible mais les étudiants doivent savoir trouver une expression régulière équivalente pour chacune des opérations introduites.

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, notion de calcul dans un afd,
- Langage reconnu (ou accepté) par automate : calcul réussi, mots reconnus (ou acceptés). Classe des langages reconnaissables.
- Accessibilité et co-accessibilité; afd émondé
- Afd complet; complétion d'un afd
- **Automate complémentaire : Si L est reconnaissable alors \bar{L} l'est (*)**
- Automate fini non déterministe (afnd) : définition et calcul d'un afnd, mots reconnus, langage accepté
- Déterminisation d'un afnd
- Automate fini non déterministe avec transitions instantanées (ϵ -afnd) : définition, ϵ -clôture, calculs d'un ϵ -afnd, mots reconnus, langage accepté
- Déterminisation d'un ϵ -afnd (automate des parties)
- Lemme de l'étoile. **Le langage $L = \{a^n b^n, n \in \mathbb{N}\}$ n'est pas reconnaissable (*)**.

NB : le lien entre langages réguliers et langages reconnaissables n'a pas encore été étudié

3 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.

4 Théorème de Kleene (Le début)

- La classe des langages réguliers et la classe des langages reconnaissables sont les mêmes.
- Algorithme de Berry-Séthi : passage d'une expression régulière à un automate. Dans ce contexte : langage local (notations utilisées : $P(L)$ pour les premières lettres, $D(L)$ pour les dernières et $T(L)$ pour les paires de lettres valides), automate de Glushkov pour reconnaître un langage local, procédure de Berry-Séthi.