

## Programme de colle - Semaine 11 (colle du 26 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, 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 ( $\epsilon$ -afnd) : définition,  $\epsilon$ -clôture, calculs d'un  $\epsilon$ -afnd, mots reconnus, langage accepté
- Déterminisation d'un  $\epsilon$ -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*

### 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. **Algorithm 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

- 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.
- Algorithme de Thomson : passage d'une expression régulière à un automate. Notion d'automate **normalisé**. Automates de Thomson : construction par induction d'un automate normalisé qui reconnaît le langage dénoté par une expression régulière.
- Passage d'un automate à une expression régulière :
  - Automate généralisé avec des expressions régulières sur les transitions.
  - Algorithme par élimination successive des états en partant d'un automate normalisé.
- Rappels sur les conséquences du théorème de Kleene sur les stabilités (propriétés de fermeture) des langages réguliers=reconnaissables.

### 4 Apprentissage supervisé (début)

- Principe de l'apprentissage supervisé
- Arbres binaires de décision : définition et écriture de la fonction de classification utilisant un ABD.
- Éléments de théorie de l'information : définition de l'entropie, cas du pile ou face traité en cours, calcul de l'entropie dans le cas d'un ensemble d'exemples d'apprentissage, calcul du gain d'information pour un attribut binaire uniquement.
- Algorithme ID3 (\*): savoir écrire l'algorithme en pseudo-code et l'appliquer sur un exemple. Seul le cas d'apprentissage d'arbres **binaires** de décision est au programme (attributs binaires)