

# THL (Théorie des langages)

## Programme du cours

David A. Madore

3 octobre 2018

### INF105

Git: ad186bc Wed Oct 3 11:13:07 2018 +0200

**1.** Présentation générale. Alphabet, mots, langages. Opérations et notions sur les mots : concaténation de deux mots, longueur d'un mot; préfixe, suffixe, facteur, sous-mot et mot miroir.

Opérations sur les langages : opérations booléennes (complémentaire, union, intersection), concaténation, étoile de Kleene. Définition des langages rationnels. Expressions rationnelles.

**2.** Automates finis : automates déterministes (DFA), automates déterministes à spécification incomplète, automates non déterministes (NFA), automates non déterministes à transitions spontanées ( $\epsilon$ -NFA).

Équivalence entre ces différentes sortes d'automates. Langages reconnaissables.

**3.** Stabilité des langages reconnaissables par complémentaire, union, intersection; stabilité par concaténation et étoile de Kleene. Les langages rationnels sont reconnaissables. Automate de Glushkov *ou* (au choix) automate de Thompson d'une expression rationnelle.

Automates à transitions étiquetées par des expressions rationnelles (informellement), équivalence avec les autres sortes d'automates, équivalence avec les expressions rationnelles (par élimination des états =algorithme de Kleene). Équivalence entre langages rationnels et reconnaissables.

**4.** Énoncé et démonstration du lemme de pompage pour les langages rationnels (=reconnaissables).

DFA minimal\* (=canonique). Algorithme de minimisation (=algorithme de Moore). (On insistera plus sur l'aspect algorithmique que sur la formulation théo-

---

\*. On conviendra d'utiliser la notion d'automate déterministe *complet*, et la première étape de minimisation sera de compléter l'automate en lui ajoutant éventuellement un puits.

rique du théorème de Myhill-Nerode.) Conséquence : on peut décider algorithmiquement si deux expressions rationnelles sont équivalentes.

**5.** TD sur les automates finis et langages rationnels.

**6.** TP sur les expressions régulières et automates finis.

**7.** Grammaires hors contexte<sup>†</sup>, langages algébriques (=définis par une grammaire hors contexte). Dérivations, dérivations gauches et droites. Arbre d'analyse (=de dérivation). Ambiguïté (grammaires inambiguës et ambiguës, exemple de langage intrinsèquement ambigu).

**8.** Stabilité des langages algébriques par réunion, concaténation et étoile de Kleene. Les langages rationnels sont algébriques : d'après ce qu'on vient de dire et directement en associant une grammaire à un DFA ou NFA.

L'intersection d'un langage algébrique et d'un langage rationnel est algébrique (admis sans démonstration).

Lemme de pompage pour les langages algébriques (admis sans démonstration, mais on peut esquisser l'idée).

L'appartenance d'un mot au langage défini par une grammaire hors contexte est algorithmiquement décidable (l'énoncer et éventuellement expliquer une approche possible<sup>‡</sup> si le temps le permet).

Selon le temps disponible : quelques notions sur l'analyse syntaxique en pratique : notion d'analyseurs descendants et ascendants (sans entrer dans les détails).

**9.** TD sur les grammaires hors contexte et langages algébriques.

**10.** Éléments de calculabilité<sup>§</sup> : algorithme, terminaison d'un algorithme, thèse de Church-Turing.

Ensembles/langages<sup>¶</sup> décidables (=calculables, =récursifs) et semi-décidables (=semi-calculables, =récursivement énumérables). Un ensemble est décidable ssi lui et son complémentaire sont semi-décidables. Un ensemble est semi-décidable ssi il est (vide ou) énuméré par une fonction calculable.

Notion de machine universelle. Indécidabilité du problème de l'arrêt.

**11.** TP sur les grammaires hors contexte avec JavaCC.

---

<sup>†</sup>. Je propose pour gagner du temps de ne faire que mentionner au passage le fait qu'il existe des grammaires plus générales, sans entrer dans les détails.

<sup>‡</sup>. Par ex., en montrant qu'on peut trouver une grammaire monotone équivalente; ou bien esquisser comment on peut mettre la grammaire sous forme normale de Chomsky et utiliser l'algorithme de programmation dynamique (CYK)?

<sup>§</sup>. Mieux vaut sans doute ne pas perdre de temps à introduire un modèle de calculabilité particulier (machine de Turing ou fonctions générales récursives, par exemple) : insister sur le fait que tout langage de programmation raisonnable, suffisamment idéalisé, est équivalent.

<sup>¶</sup>. Souligner qu'ici contrairement au reste du cours, on peut indifféremment considérer des ensembles d'entiers naturels ou de mots.