

# Leçon 909 – Langages rationnels et automates finis. Exemples et applications.

17 juillet 2019

## 1 Extraits du Rapport

### Rapport de jury 2018

*Pour cette leçon très classique, il importe de ne pas oublier de donner exemples et applications, ainsi que le demande l'intitulé. Une approche algorithmique doit être privilégiée dans la présentation des résultats classiques (déterminisation, théorème de Kleene, etc.) qui pourra utilement être illustrée par des exemples. Le jury pourra naturellement poser des questions telles que : connaissez-vous un algorithme pour décider de l'égalité des langages reconnus par deux automates ? quelle est sa complexité ? Des applications dans le domaine de l'analyse lexicale et de la compilation entrent naturellement dans le cadre de cette leçon.*

## 2 Cœur de la leçon

- Définitions des automates finis et de la notion de langages reconnaissables. Exemples de langages reconnaissables.
- Définitions des expressions rationnelles et langages rationnels.
- Théorème de KLEENE, connaître/mentionner les constructions pour la preuve (parmi THOMPSON, GLUSHKOV, ANTIMIROV, MCNAUGHTON-YAMADA, BRZOSOWSKY-MCCLUSKEY). Attention toutefois, Glushkov qui paraît simple n'est pas facilement trouvable dans les livres.

## 3 À savoir

- Existence de langages non reconnaissables. Lemme de pompage. Exemples.
- Applications, par exemple : la compilation, la recherche de motif, le model checking (et autres).
- Accessibilité, Complétude, déterminisation, minimisation (NERODE).
- Propriétés de clôture des langages reconnaissables, et construction effective sur les automates.
- Les complexités des algorithmes.
- Analyse lexicale et chaîne de compilation.

## 4 Ouvertures possibles

- Des questions de complexité.
- Lemme d'ARDEN.
- Divers algorithmes de minimisation (dur à faire en développement)
- Reconnaissance par monoïdes.
- Lien avec les autres modèles de calcul (Boustrophédon, Machine de TURING. qui n'écrivent pas sur leur entrée).
- Lien avec la logique (MSO).

## 5 Conseils au candidat

- Si l'analyse lexicale et la compilation peuvent être un exemple d'application, ils ne doivent pas prendre une place prépondérante. Les automates à piles ne sont notamment pas au programme. On ne rentrera pas dans de nombreuses définitions pour les grammaires.
- Cette leçon est basique et fondamentale, il faut très bien couvrir toutes les bases, et éviter de partir trop loin théoriquement.
- Réfléchir à la complexité.
- Éviter MCNAUGHTON-YAMADA, qui est infaisable sur un exemple.
- Les expressions rationnelles sont des termes, pas des mots.

## 6 Questions classiques

- Comment tester le vide d'un automate ? Comment tester l'acceptation d'un mot ? L'inclusion de langages ? L'égalité de langages ? Obtenir le complémentaire d'un langage ? Complexités ? (attention aux inputs, regexp vs.  $\mathcal{A}$  déterministe vs.  $\mathcal{A}$  non déterministe)
- Dur : Tester l'universalité d'une expression rationnelle ?
- Automate dont le déterminisé est en  $2^{|Q|}$  ?
- Quelle classe de complexité est reconnue par les automates ?
- Comment prouver la minimalité d'un automate ?
- Prouver que tel langage  $X$  n'est pas régulier.
- Coût d'élimination des  $\epsilon$ -transitions ?
- Avantages/inconvénients des constructions du théorème de KLEENE ?

## 7 Références

- [Car] Langages formels, calculabilité et complexité - CARTON - à la BU/LSV  
*Très bonne référence couvrant beaucoup de bases. Se méfier de certaines preuves faites un peu rapidement.*
- [Bea] Éléments d'algorithmique - D. BEAUQUIER, J. BERSTEL, Ph. CHRÉTIENNE - à la BU/LSV  
*Bonne référence pour l'algo, pleins de dessins et de preuves. Un peu vieillissant et devenu rare.*
- [Sak] Éléments de théories des Automates - SAKAROVITCH - à la BU/LSV  
*Style austère, mais très complet, et bonne source de développements. Beaucoup de hors programme.*

## 8 Dev

- ++ Théorème de KLEENE - ([Car], Thm 1.59 p.36) - 907,909,923  
*Tout faire est ambitieux, mais cela passe si on prend les constructions les plus basiques (THOMPSON et MCNAUGHTON-YAMADA). Si on fait par contre ANTIMIROV, cela peut suffire.*
- + Décidabilité de l'arithmétique de PRESBURGER - ([Car], Thm 3.63 p.164) - 909,914,924  
*Idée générale simple, mais attention aux détails. Réfléchir au codage, à sa sémantique, et à la complexité globale de la construction.*
- + Automate des occurrences - ([Cor], p.886) - 907,909,927  
*Tiens bien en 15 min, mais attention à bien maîtriser les petits calculs. Extension possible vers KMP.*
- ++ Minimisation de NÉRODE - ([Car], Sec 1.7 p.49) - 909
- ++ Lemme de l'étoile et variantes - ([Car], Sec 1.9 p.61) - 909
- + Automate d'Aho-Corasick - ([Cro2], [Bea]) - 907,909,921  
*Généralise KMP à plusieurs motifs. [Bea] donne bien mieux les intuitions que [Cro2]*