

# Leçon 923 - Analyses lexicale et syntaxique. Applications.

7 juin 2019

## 1 Extraits du Rapport

### Rapport de jury 2018

*Cette leçon ne doit pas être confondue avec la 909, qui s'intéresse aux seuls langages rationnels, ni avec la 907, sur l'algorithmique du texte. Si les notions d'automates finis et de langages rationnels et de grammaires algébriques sont au cœur de cette leçon, l'accent doit être mis sur leur utilisation comme outils pour les analyses lexicale et syntaxique. Il s'agit donc d'insister sur la différence entre langages rationnels et algébriques, sans perdre de vue l'aspect applicatif : on pensera bien sûr à la compilation. Le programme permet également des développements pour cette leçon avec une ouverture sur des aspects élémentaires d'analyse sémantique.*

### Rapport de jury 2017

*[...] On pourra s'intéresser à la transition entre analyse lexicale et analyse syntaxique, et on pourra présenter les outils associés classiques, sur un exemple simple. Les notions d'ambiguïté et l'aspect algorithmique doivent être développés. La présentation d'un type particulier de grammaire algébrique pour laquelle on sait décrire un algorithme d'analyse syntaxique efficace sera ainsi appréciée. [...]*

## 2 Cœur de la leçon

- Grammaires et Langages algébriques. Existence de langages non algébriques (Lemme d'OGDEN). Propriétés de clôture des langages algébriques.
- Chaîne de compilation. Analyse lexicale. Analyse syntaxique (principes de l'analyse descendante et ascendante).

## 3 À savoir

- Formes normales. (quadratique, CHOMSKY, GREIBACH)
- Automate à pile et équivalence avec les langages algébriques.
- Analyse sémantique élémentaire (arbre de syntaxe abstraite, table des symboles, analyse de portée, typage, ...).
- Grammaires LL ou LR
- Ambiguïté.

## 4 Ouvertures possibles

- LALR ou SLR
- Problèmes de décision des langages algébriques. Décidabilité et complexité.
- Langages de pile.
- Hiérarchie de CHOMSKY.

## 5 Conseils au candidat

- Des DESSINS, de la chaîne de compilation, d'arbre de dérivation, d'automate, de règles de grammaires,...
- Mettre des exemples. Bien s'entraîner à faire tourner les algorithmes sur des exemples (notamment pour l'analyse lexical)
- Attention à ne pas trop se perdre sur certain théorique. On peut notamment se perdre et ne pas savoir répondre aux questions autour des parseurs LL, LR, LALR et SLR.
- En pratique, yacc est LALR, mais par exemple, le langage pascal est LL(1)

## 6 Questions classiques

- Qu'utilisent en pratique les programmeurs pour écrire des parseurs? Quelle type de grammaire est reconnu par ces outils?
- Peut-on comparer les parseurs LALR, LR, LL, SLR? Si des inclusions sont strictes, avez-vous des exemples?
- Quelles sont les applications de l'analyse lexicale et syntaxique? Avez-vous un exemple où la complexité est critique?
- Complexité d'une mise en forme normale?
- Pourquoi séparer l'analyse en deux étapes?
- Pourquoi l'ambiguïté d'une grammaire est problématique?

## 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.*
- [Aho] Compilers : Principles, Techniques, and Tools - AHO et. al - à la BU/LSV  
*La référence pour la compilation, surnommé le dragon. En réalité, seul deux/trois chapitres sont utiles.*

## 8 Dev

- ++ Algorithme de Cocke-Kasami-Younger - ([Hopcroft, Ullman], Ch7.4.4, p298) - 923  
*Uniquement esquissé dans le Carton. Maîtriser la mise en forme normale de Chomsky.*
- ++ Un exemple d'analyse d'un langage jouet - ([Aho], Ch4.4 ou Ch 4.9.1 p298) - 923  
*Le langage d'une calculatrice minimaliste est un classique. L'analyse d'une grammaire LL(1) est plus difficile. Ne pas hésiter à faire des dessins, avec des arbres d'interprétation possible.*
- Problèmes indécidables pour les grammaires algébriques. - ([Car], ) - 914,923  
*Ambiguïté, universalité. Insister sur la réduction, pas sur la notion de grammaire.*
- 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.*