1. Introduction générale à la Théorie des Langage Formels et à la Compilation

Bernard ESPINASSE
Professeur à l'Université d'Aix-Marseille
1998

- **Introduction**
- **Présentation du cours**

Problématique

Les **microprocesseurs** disposent :

- d'un **jeu d'instructions très sommaire**,
- **peu convivial**
- et qui **varie d'un processeur à l'autre**

d'où la nécessité de :

fournir à l'utilisateur de systèmes informatiques :

- *des **formalismes universels**, **expressifs** et **concis**,*
- *de nombreux **langages de programmation***

La théorie des langages

- elle constitue l'un des **fondements de la science informatique**
- elle est **enseignée** dans tous les cursus d'informatique à l'étranger et en France, dans de nombreuses grandes écoles et universités
- la **recherche française** en ce domaine est particulièrement réputée dans le monde entier
  - «**French School**»
  - doit beaucoup à **M. P. Schützenberger** et **M. Nivat**
- elle est née d'une tentative de modélisation des **langues naturelles**
- elle connut une expansion considérable lorsqu'on s'aperçut de son adéquation à la description des **langages de programmation** :
  - **ses concepts** : langages algébriques, grammaire, dérivation, automate à pile,
  - sont à la **base** de tous les **algorithmes d'analyse syntaxique**, et partant des **compilateurs**

de la théorie des langages .... à la compilation

- la **théorie des langages** définit ces langages de programmation
- la **compilation** transforme les programmes écrits dans ces langages en code machine exécutables par les microprocesseurs

- **4 parties dans le cours :**
  - **1 : Théorie sur les mots**
  - **2 : Grammaires et langages formels**
  - **3 : Les automates**
  - **4 : Introduction à la compilation**

- **EP 1280 "Élaboration d'un langage de programmation"**
  - définition du langage
  - élaboration de l'analyseur syntaxique d'un compilateur associé à l'aide de Lex et Yacc (unix)

Partie 1 : Théorie des mots

Séance 1-2 : Théorie des mots

- présentation générale du cours
- cadre dans lequel se situe la théorie des langages (définitions).
- **phénomène de l'écriture** : emploi de symboles que l'on dispose les uns derrière les autres pour former des mots

-> structure de **monoïde libre** : cas des chaînes de caractères munies de la **concaténation**

-> théorie des mots.

- **calcul associatif** construit sur la **congruence** et l'**équivalence de Thue**
- **problème des mots** et sa résolution

Séance 3 : Les systèmes combinatoires (systèmes de réécriture)

- on introduit les systèmes combinatoires dans le monoïde libre :
  - notions de **production**
  - **schéma de production**
  - **types de production** : Semi-Thueiennes, normales et antinormales.
  - **definition** et **propriétés** des **systèmes combinatoires** (caractère monogénique, l'ambiguïté d'une démonstration, ...)

partie 2 : Grammaires et langages formels

Séance 4 : Introduction aux grammaires et langages formels

- notion de **langage**, en lien avec celle d'algorithme devant être effectués par des machines :
  - **syntaxe** et **sémantique** d'un langage
  - notion de **grammaire**
  - notion de **dérivation** dans une grammaire
- **langage de Backus** pour l'écriture des grammaires des langages de programmation.

Séance 5 : Grammaires et langages de Chomsky (C-grammaires et C-langages)

- Différentes **classes de grammaires** selon leur capacité descriptive
- **classification hiérarchique des grammaires formelles** selon leur capacité descriptive, cad la complexité des **langages** qu'elles engendrent
- **grammaires régulières** et **grammaires hors contexte**
- **grammaires ambigues**
- **opérations ensemblistes et symboliques** sur les langages
- **expressions régulières** : definition et propriétés

Partie 3 : Les automates

Séance 6-7 : Les automates à états finis

- **automates** = "machines" symboliques validant l'appartenance d'une chaîne donnée au langage qu'ils décrivent
- **automates à états finis** sont associés aux grammaires régulières
  - le **langage reconnu**
  - caractère **déterministe** ou **non déterministe**
  - **passage** d'un automate **non déterministe** à un automate **déterministe**
  - **passage d'une grammaire régulière** à **un automate à états fini**

Séance 8 : Les automates à pile

- associés aux grammaires hors contexte
  - **langage reconnu**
  - caractère **déterministe** ou **non déterministe**
  - **passage d'une grammaire hors contexte** à **un automate à pile**

Partie 4 : Introduction à la compilation

Séance 9 : Introduction à la compilation

- notions de traducteur, interpréteur et compilateur
- architecture et grandes fonctions des compilateurs :
  - **analyse lexicographique**
  - **analyse syntaxique**
  - **génération et optimisation du code**

Séance 10 : Analyse syntaxique d'un compilateur

- méthodes générales :
  - **méthode générale descendante**
  - **méthode générale montante**
  - autres méthodes