Université Badji Mokhtar d'Annaba

Faculté de Technologie

Département d'Informatique

Théorie des Langages

Cours et exercices

2024

Dr. Bilel HAMADACHE

TABLE DES MATIÈRES

7 Chapitre 7
Langages algébriques : Automates à pile et grammaires-type 2-

CHAPITRE 1

PRÉFACE

Ce manuel est un support de cours du module "Théorie des Langages," destiné aux étudiants en deuxième année informatique (LMD). Ce cours a pour objectif de fournir des concepts fondamentaux liés aux langages formels, aux grammaires, aux automates et aux expressions régulières. En complément de toutes ces notions, le manuel intègre des séries d'exercices et des travaux pratiques visant à renforcer la compréhension de ces concepts et à les mettre en pratique dans des contextes réels. Fruit de notre modeste expérience pédagogique aux côtés d'enseignants plus expérimentés tels que Dr. Benouhiba, ce cours reste encore largement perfectible et non-exempte d'erreurs. Nous encourageons activement les commentaires et suggestions dont le but est d'améliorer continuellement la qualité de ce support de cours afin de garantir une meilleure expérience d'apprentissage.


alphabet
déterminisme
symbole
Stabilité
hors-contexte
normalisation
complexité
ensemble
régulier
congruence
calculabilité
union
inclusion
Kleene
rejet
configuration
terminaux
minimisation
motif
Dépiler
exposant
formel
AEF
grammaire
intersection
thompson
propriété
ambiguë
induction
POSIX
règle
Chomsky
algébrique
syntaxe
mot
préfixe
facteur
accessibilité
suffixe
axiome
Langage
AP
appartenance
transition
état
pompe
concaténation
graphe
acceptation
chemin
déterminisation
palindrome
arbre
correspondance
simulation
grammaire propre
β-equivalent
miroir
entrelacement
décidabilité
dérivation
analyse
ε-fermeture
Produit
automate
Regex
complément
dérivées
récursivité
contextuel
ε-transitions
fermeture
remplacement
pile
dictionnaires
tête
Empiler

1.1 Introduction

Un langage est un moyen de communication entre différentes entités. C'est un concept souvent associé à l'intelligence humaine, en dépit de son universalité. Les animaux possèdent aussi leur propre langage, même les insectes, comme les abeilles et les fourmis, ont la capacité de communiquer de manières subtiles, s'informant mutuellement sur des endroits où trouver de la nourriture. Par ailleurs, notre langage ne fut pas non plus le premier langage vocal. Beaucoup d'animaux, y compris des espèces de singes, ont aussi des langages vocaux. Mais le trait le plus singulier du langage du Sapiens (l'homme moderne) réside dans la capacité de transmettre des informations non seulement sur des faits, sur des hommes mais aussi sur des choses qui n'existent pas, sur des choses qui ont eu lieu dans le passé ou qui pourraient avoir le lieu dans le futur. Cette faculté de parler de fictions, de coopérer socialement nous distingue par exemple des singes, et surtout des autres espèces humaines archaïque (comme le Néandertal).

L'apparition de nouvelles façons de penser et de communiquer à travers des messages chiffrés, des écritures, des ouvrages en différentes langues a caractérisé une vraie première révolution cognitive de notre espèce humaine. Cette révolution (la pierre angulaire) a précédé la révolution agricole, industrielle, scientifique, etc. Il y avait quelque chose de si particulier que possède la langue des Sapiens, qu'elle nous ait permis de conquérir le monde, de raconter des histoires, des mythes, de convaincre les gens d'y croire. La communication humaine continue à évoluer et passe par plusieurs canaux : e-mails, appels téléphoniques, échos dans la presse, médias sociaux, etc. Aujourd'hui, on ne communique pas seulement à travers des machines, mais on rend des machines de plus en plus intelligentes. Étant donné que notre langage est davantage axé sur notre pensée, les machines apprennent aussi à comprendre, raisonner et de générer du langage de manière similaire à celui généré par la pensée humaine.

Fondamentalement, un langage repose sur la notion de codage. Lorsque deux entités A et B qui n'ont pas de mémoire commune souhaitent communiquer pour échanger des informations, ils ont besoin de coder ces informations (selon des règles R connues par A et B) et les transmettre via un support (Fumé dans l'air, signaux sonores, fil électrique, onde électromagnétique, substance chimique, etc). Prenons un simple scénario ou l'entité A souhaite envoyer une information à l'entité B :

1. L'entité A code son message selon les règles R.
2. Le message codé est envoyé via un support vers B
3. L'entité B reçoit le message et le décode en appliquant les règles R et récupère ainsi l'information

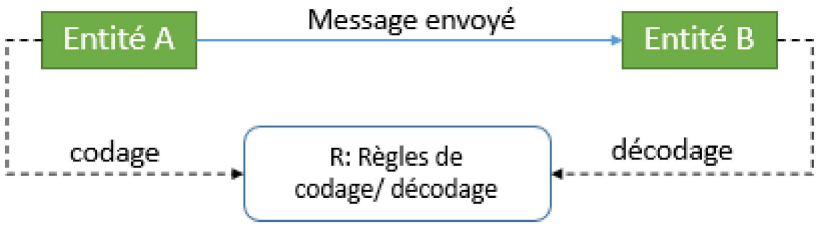


FIGURE 1.1 – Communiquer selon un langage basé sur le codage/décodage

Le bon fonctionnement de ce processus nécessite l'utilisation de règles communes de codage et de décodage partagées par les entités A et B. Si par exemple l'entité A s'exprime uniquement en arabe et envoie son message à B qui ne comprend que l'anglais. dans ce scénario, il devient impossible de communiquer et comprendre l'information échangée.

1.2 Langage formel et théorie des langages

Il est important tout d'abord de faire la différence entre un langage formel et un langage naturel, celui utilisé par les humains. Un langage formel est intrinsèquement défini comme étant dépourvu de sémantique et se caractérise uniquement par sa syntaxe. Cette distinction le sépare considérablement des langages naturels, car il ne repose que sur une structure grammaticale sans tenir compte du sens des mots.

Toutefois, le langage humain possède une caractéristique notable, celle de l'ambiguïté. C'est à dire que la même structure syntaxique peut exprimer des significations (sémantique) différentes et dans des contextes

différents. L'étude de cette sémantique appartient généralement au domaine de l'intelligence artificielle.

Pourquoi étudier la théorie des langages ?

La théorie des langages étant un domaine de l'informatique théorique fondamentale s'intéresse plutôt aux langages formels. C'est également un domaine dont les principaux concepts et outils trouvent aussi des origines dans les mathématiques et la linguistique. Nous étudions à travers ce cours, la théorie des langages formels en informatique partant du principe que l'ordinateur a considérablement amplifié l'importance des langages formels :

1. L'automatisation des processus de calcul avec des algorithmes écrits dans des langages de programmation formels. Un compilateur est nécessaire pour transformer les commandes d'un langage de programmation donné en un langage exécutable par une machine.
   Mais ce processus passe par plusieurs étapes : Une analyse lexicale qui consiste à identifier des séquences de caractères formant des mots-clés, des noms de variables ou des nombres réels à titre d'exemple. Une autre étape majeure consiste à analyser syntaxiquement la structure interne des séquences de symboles et d'opérateurs, etc., pour détecter les erreurs de syntaxe et garantir que les programmes (séquences finies) soient bien formés. Par exemple un compilateur d'expressions arithmétiques doit pouvoir analyser une séquence $x+y\times z$ et traduire correctement le calcul requis : $x+(y\times z)$.

   Donc avant d'étudier les aspects de la compilation, la théorie des langages va répondre aux défis qui consistent à comprendre comment définir la syntaxe des programmes bien formés, identifier les séquences et la structuration interne des programmes respectant cette syntaxe afin de permettre de déterminer la séquence d'instructions à exécuter.
2. Le développement de la théorie de la calculabilité et des automates qui étudient la puissance et les limites des langages formels
3. Traitement du langage naturel (NLP) : Les langues naturelles ou encore celles parlées par les humains sont des systèmes d'écriture, des mots ou encore des suites linéaire définis sur des signes (Par exemple en français, ce sont des symboles alphabétiques). Les phrases dans une langue sont aussi considérées comme des séquences d'éléments venant d'un inventaire fini (dictionnaire de mots).
   Mais sachant que toutes les combinaisons ne sont pas grammaticalement correctes, le traitement informatique des langues naturelles nécessite la distinction entre ce qui relève de la langue et ce qui n'en relève pas. Ce genre de problématique peut être formulé en théorie des langages. De l'autre coté, la théorie des langages a bénéficié de nombreuses avancées grâce aux travaux des linguistes formels.
   Ce domaine de recherche nécessite aussi la compréhension de la structure des énoncés (identifier le sujet , le verbe, etc.), afin de comprendre le sens, générer du texte en langage naturel, et éventuellement le traduire dans d'autres langues, etc. Ce domaine a connu aussi des avancées significatives grâce à l'IA, en particulier avec l'émergence des LLMs (Large Language Models).
4. En bioinformatique : La biologie moléculaire et la génétique offrent des exemples naturels (Un chromosome formé de deux brins d'ADN) modélisables par des séquences linéaires de symboles (ensemble fini de symboles) : A, T, C, G qui représentent quatre bases différentes (des nucléotides). Ces séquences de nucléotides fournissent une information génétique importante. La bio-informatique se concentre sur la recherche de séquences spécifiques de nucléotides dans les chromosomes et la détection de similitudes, de différences entre des fragments d'ADN, permettant des applications telles que l'évaluation des proximités évolutives et la localisation de gènes similaires. La théorie des langages nous offre des outils pour modéliser ce type de calculs qui s'étend également aux protéines, dont la structure primaire peut être également modélisée par une séquence linéaire des acides aminés (en un nombre fini : 20 lettre possibles)

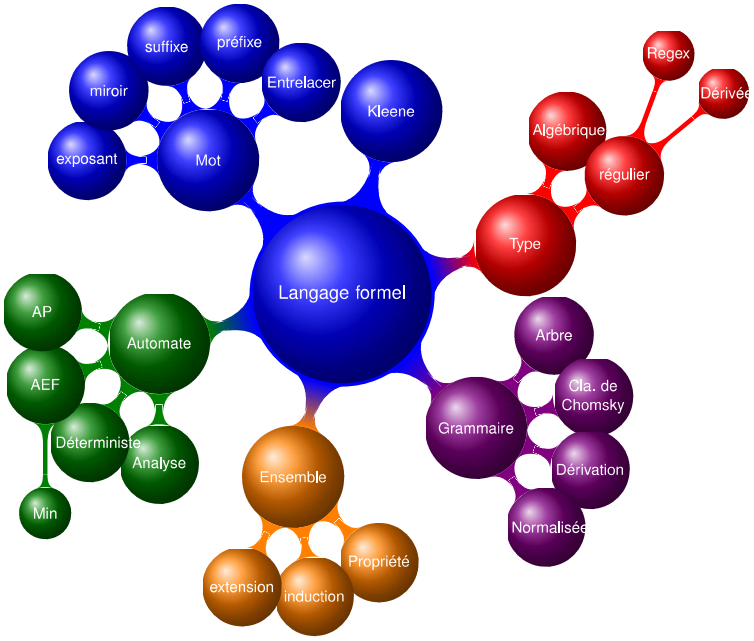
Pour résumer, les langages formels sont construits selon des modèles mathématiques stricts (un cadre formel : théorie des ensembles, théorie des graphes, etc.) compréhensibles par la machine. Mais, comment peut-on formaliser et étudier ces langages ?

1.3 Objectifs

- Ce cours va nous permettre d'aborder des concepts nécessaires à l'étude d'un langage formel. Nous partons de certaines notions simples comme : alphabet, symbole, mot, langage jusqu'à arriver à la théorie des grammaires génératives et des automates de reconnaissance.

- Pour définir et étudier un langage formel, ce cours va s'appuyer sur une question fondamentale **"Est-ce qu'un mot w appartient à un langage L ?"**. La réponse à cette question nous servira à donner diverses définitions aux langages. Globalement, nous allons définir et étudier un langage formel selon trois points de vue :

FIGURE 1.2 – Certains principaux concepts abordés dans ce cours (Mindmap)



1. Point de vue ensembliste de L : Notre objectif ici est d'étudier les langages comme des ensembles et raisonner dessus. La théorie des ensembles sera d'ailleurs un prérequis important pour entamer ce cours.
   - Le but et de se familiariser avec les opérations sur les mots (préfixes, suffixes, miroir, entrelacement, etc.), les étendre sur des langages avec d'autres opérations : Ensemblistes, Fermeture de Kleene, Concaténation, etc.
2. Définition grammaticale : L'objectif ici est de montrer comment générer (dériver) les éléments d'un langage L via une grammaire.
   - La théorie des grammaires génératives, la normalisation des grammaires aident à développer des analyseurs de programmes notamment les compilateurs.
   - Etudier des grammaires, nous permettra également d'étudier la classification de Chomsky et reconnaître ainsi différents types de langages formels.
   - Ce cours permettra d'étudier et manipuler certains types de langages, essentiellement des langages réguliers et des langages algébriques
   - En étudiant des lagnages réguliers, ce cours permettra d'utiliser des outils puissants comme les expressions régulièrer pour récupérer des informations, résoudre des problèmes de recherches scientifiques pratiques en sécurité informatique, en bioinformatique, etc.
3. Définition par automate : Selon ce point de vue notre objectif est de fournir une réponse à la question d'appartenance d'un mot à un langage par oui ou non après l'avoir analysé avec une machine abstraite (automate/algorithme).
   - Ce cours permettra d'étudier des automates comme les automates à états finis (AEF), des automates à pile. etc.
   - Le but est aussi l'assimilation du fonctionnement de quelques types d'automates pour pouvoir comprendre la notion de décidabilité, et les bases de la théorie de la complexité algorithmique.
   - Des cas pratiques nous montrerons comment analyser et évaluer la complexité (le temps nécessaire pour analyser un mot).

  - Ce cours abordera des notions comme le déterminisme pour comprendre le comportement de ces automates

- Ce cours offre aux étudiants une variété d'outils de modélisation. Chacun permet de répondre à la question posée de manière complémentaire par rapport au précédent. C'est le cas par exemple des langages de programmation usuels, ou l'algorithme/l'automate d'analyse (le compilateur) est construit à partir de la grammaire du langage. L'algorithme construit sera par la suite modifié pour faire davantage d'analyses.
- À l'issue de ce cours, l'étudiant pourra acquérir des notions fondamentales utiles pour la suite de son cursus et notamment en matière de compilation.

1.4 Organisation

Ce support de cours est organisé en sept chapitres, accompagnés de cinq séries d'exercices de travaux dirigés (TD) et quatre séries d'exercices pratiques (TP) permettant d'approfondir les différentes notions du cours et de les mettre en pratique.

1. Chapitre introductif.
2. Chapitre 02 : Dans ce chapitre, nous définissons les notions fondamentales de la théorie des langages, en se focalisant sur la définition ensembliste d'un langage formel et ses trois variantes (par extension, par compréhension, par induction). Nous étudions certaines opérations sur les mots, des opérations standards ensemblistes sur les langages et d'autres opérations spécifiques. Ce chapitre se termine par une série de TD et une série de TP
3. Chapitre 03 : Dans ce chapitre, nous étudions les grammaire formelles, leur formalisme, dérivations et arbres syntaxiques, puis nous aborderons la classification de Chomsky des grammaires et des langages. Ce chapitre se termine par une série de TD et une série de TP.
4. Chapitre 04 : Ce chapitre est consacré aux machines abstraites (automates) qui reconnaissent les types de langages vus dans le chapitre précédent. Nous étudions ici les composants essentiels qui constituent un automate de manière générale, ses configurations clés. Ces automates nous serviront aussi pour étudier brièvement la décidabilité d'un langage (langage décidable ou récursif) et sa complexité qui lui est intrinsèquement liée.
5. Chapitre 05 : Ce chapitre concerne les automates reconnaissant les langages réguliers, en l'occurrence les automates à états finis. Nous aborderons leur représentation formelle, leur mode de fonctionnement, leur déterminisme, ainsi qu'un ensemble d'algorithmes d'optimisation. Nous aborderons également les opérations courantes effectuées sur ces automates et discuterons de leur simulation selon diverses structures et stratégies (en Python). Ce chapitre se termine par une série de TD et une série de TP.
6. Chapitre 06 : Ce chapitre est consacré à la forme particulière des langages réguliers (rationnels) qui sera étudiée sous la lumière des expressions régulièrers. Nous parlerons de l'utilité pratique des expressions régulières, des notions comme l'ambiguïté. Nous verrons comment s'en servir pour effectuer des correspondances, des recherches de motifs et des remplacements. Nous explorerons par ailleurs divers algorithmes et méthodes permettant d'établir des équivalences entre les trois principales représentations des langages réguliers : expressions régulières, automates à états finis et grammaires régulières. Nous discuterons également de la régularité des langages réguliers et de leurs propriétés (fermeture). Ce chapitre se termine également par une série de TD et une série de TP.
7. Chapitre 07 : Ce dernier chapitre concerne les langages dits algébriques. Nous allons étudier la représentation des machines abstraites (automates à pile) qui reconnaissent ce type de langage, leur fonctionnement et déterminisme. La deuxième partie de ce chapitre sera consacrée aux grammaires qui génèrent ce type de langage. Nous aborderons des notions liées aux arbres de dérivation, ambiguïté, ainsi que la simplification et la normalisation de ces grammaires. Ce chapitre se termine par une série de TD.