LF – Théorie des langages formels

Sylvain Brandel

2025 – 2026

sylvain.brandel@univ-lyon1.fr

Partie 1

INTRO, RAPPELS MATHS, ALPHABETS ET LANGAGES

Fonctionnement

- CM : 10 x 1h30 (10 → 6 séances)
  - Mardi 14h
  - Cela parait évident mais … Silence dans l'amphi. S'il vous plait.
- TD : 6 x 1h30 (6 → 8 séances)
  - Généralement lundi 8h
  - Début des TD lundi 8 septembre 2025
- TP : 4 x 1h30 (4 → 6 séances)
  - Généralement lundi 9h45 ou 11h30, après le TD de LC
  - Début des TP lundi 23 septembre 2025

- Fin des enseignements mardi 2 décembre 2025 (hors rattrapage)

- Lien fort avec LC

Planning global

	LUNDI				MARDI		
Date	8:00-09:30	9:45-11:15	11:30-13:00		14:00-15:30	15:45-17:15	17:30-19:00
01/09/25		LF CM1	LC CM1		LC CM2	LF CM2	
08/09/25	LC TD1	LF TD1			LC CM3	LF CM3	
15/09/25	LC TD2	LF TD2			LC CM4	LF CM4	
22/09/25	LC TD3	LF TD3			LF TP1		
29/09/25	LC TD4	LF TD4			LC TP1		
06/10/25	LC TD5	LF TP2			LC CM5	LF CM5	
13/10/25	LF TD5	LC TP2			LC CM6	LF CM6	
20/10/25	LC TD6	LF TP3			LC CM7	LF CM7	
27/10/25	Pas d'enseignements en licence						
03/11/25	LF TD6	LC TP3			LC CM8	LF CM8	
10/11/25	LC TD7	LF TD7			Férié		
17/11/25	LC TD8	LF TP4			LF CM9	LF CM10	
24/11/25	LC TD9	LC TP4					
01/12/25	TP noté (date à confirmer)				LC interro finale	LF interro finale	
08/12/25							
15/12/25							
22/12/25	Vacances						
29/12/25							
05/01/26							

TP

- En Coq
  - https://softwarefoundations.cis.upenn.edu
  - Liens étroits avec LC

Evaluation

- http://sylvain.brandel.pages.univ-lyon1.fr/langages/

- UE en CCI (Contrôle Continu Intégral)
  - 3 épreuves :
    - Une interrogation d'1h, en séance de TD le lundi matin (date communiquée) 30%
    - Un TP noté de 30min, lundi 1er décembre 2025 (à confirmer) 30%
    - Une interrogation finale d'1h, mardi 2 décembre 2025 (à confirmer) 40%
  - Une épreuve de seconde chance (E2C)
    - Ouvert à tous
    - La note remplacera la moins bonne des notes obtenues aux interrogations, si elle est supérieure (le TP noté ne peut pas être remplacé par l'E2C)

- ATTENTION : Depuis 2023 – absences aux contrôles
  - ~~ABJUS => note neutralisée ; ABINJ => zéro~~
  - Plus de distinction pédagogique entre ABJUS et ABINJ
    - Un absent n'a pas été évalué, l'absence justifiée ou non a le même impact sur la note
    - En cas d'absence, justifiée ou non : zéro à l'épreuve
    - La note d'E2C remplace une absence, justifiée ou non
    - En cas d'absences justifiées à 2 épreuves ou plus, *possibilité* d'épreuves de substitution

De votre côté

- Travail personnel conséquent
- Se préparer à l'avance
- Ne pas attendre que les réponses viennent toutes seules

- Lisez vos mails …

- Contactez-moi, par mail, précisez LF

LifLF – Théorie des langages formels

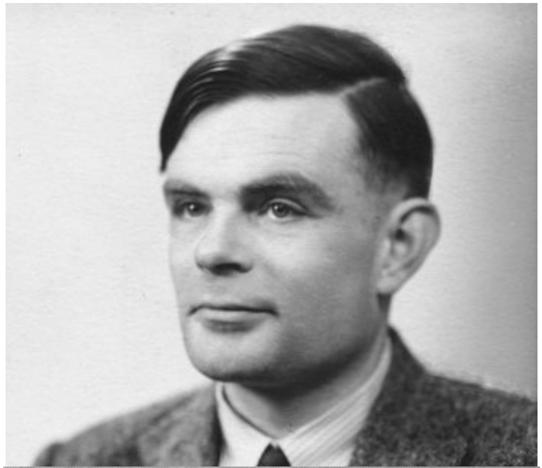
Sylvain Brandel

2025 – 2026

sylvain.brandel@univ-lyon1.fr

INTRODUCTION

Alan Turing
1912 – 1954



Kurt Gödel
1906 - 1979

Alonzo Church
1903 - 1995

Motivations

- Informatique fondamentale
- Historiquement
  - Théorie de l'incomplétude
  - Que peut-on calculer avec un algorithme ?
- Lien avec les langages de programmation
  - Ce cours prépare à deux cours de master
    - ~~Calculabilité et complexité~~ Modèles de calcul et complexité
    - Compilation
- Vous intéresser …
  - Si on sait qu'un problème est indécidable, inutile de chercher un algorithme pour le résoudre
  - Si on sait que la réponse est dans 10^{200} années, inutile de lancer le programme et d'attendre la réponse

Comment

- Définition d'objets et d'ensembles
  - Par décision → LifLF
  - Par construction → LifLC, et aussi LifLF
- En LF : fonction de reconnaissance
- En LC : preuves de correction

Programme

- Classifier des langages

Exemple d'école	Classe de langage	Reconnu par	Engendré par
$a^* b^*$	langages rationnels	automates à états finis	grammaire régulière
$\{a^n b^n \mid n \geq 0\}$	langages algébriques	automates à pile	grammaire algébrique
$\{a^n b^n c^n \mid n \geq 0\}$	langages récursifs	machine de Turing	grammaire (générale)

- La décidabilité et la complexité en découlent

Programme
Trame Lewis – Papadimitriou

- Notions mathématiques de base
  - Ensembles
  - Alphabets, langages
- Langages rationnels
  - Grammaires régulières
  - Automates finis
  - Expressions régulières
- Langages hors contextes / algébriques
  - Grammaires algébriques
  - Automates à pile

Programme

Trame Lewis – Papadimitriou

- Machines de Turing
  - Formalisme de base
  - Langages récursifs
  - Extensions
  - Machine de Turing Universelle
  - Grammaires
- Indécidabilité
  - Thèse de Church – Turing
  - Problèmes indécidables
- Complexité
  - Classes P, NP …
  - NP-complétude
  - Théorème de Cook

M1if09 (M1)

Programme
Prévisionnel

- CM1 : Notions mathématiques de base
- CM2 : Alphabets et langages

- **CM3** : Grammaires algébriques et langages algébriques
- CM4 : Automates à états finis déterministes ou non
- CM5 : Élimination du non déterminisme
- CM6 : Caractérisation des langages rationnels
- CM7 : Minimisation des états
- CM8 : Langages rationnels, expressions régulières rationalité
- CM9 : Automates à pile et algébricité

- CM10 : Analyse syntaxique

Littérature

Elements of the Theory of Computation
Harry R. Lewis, Christos H. Papadimitriou
éd. Prentice-Hall

Introduction à la calculabilité
Pierre Wolper
éd. Dunod

Introduction to the Theory of Computation
Michael Sipser, MIT
éd. Thomson Course Technology

Introduction to Theory of Computation
Anil Maheshwari, Michiel Smid, School of Computer Science, Carleton University
free textbook

Gödel Escher Bach, les Brins d'une Guirlande Eternelle
Douglas Hofstadter
éd. Dunod

Logicomix
Apóstolos K. Doxiàdis, Christos Papadimitriou, Alecos Papadatos, Annie Di Donna
éd. Vuibert

Rappels mathématiques
Ensembles

- Définition d'un ensemble
  - Par extension : $\Sigma = \{a, b, c\}$
  - Par intension : $P = \{x \in Z \mid \exists\, y \in Z \,;\, x = 2y\}$

- Opérations ensemblistes
  - Appartient : $x \in E$, $x \notin E$
  - Ensemble vide : $\forall\, x \in E_1$, $x \notin \varnothing$
  - Inclusion : $E_1 \subset E_2$, $E_1 \not\subset E_2$

    $E_1 \subset E_2$ si $\forall\, x : (x \in E_1 \Rightarrow x \in E_2)$
  - Ensemble des parties de E : $P(E) = \{E_1 \mid E_1 \subset E\}$
  - Intersection : $E_1 \cap E_2 = \{x \mid x \in E_1 \text{ et } x \in E_2\}$
  - Union : $E_1 \cup E_2 = \{x \mid x \in E_1 \text{ ou } x \in E_2\}$
  - Complémentarité : $C_E^{E1} = \{x \in E \mid x \notin E_1\}$

    (ou $\neg E_1$ lorsque E est sous entendu)
  - Différence : $E \setminus E_1 = \{x \in E \mid x \notin E_1\}$
  - Produit cartésien : $E_1 \times E_2 = \{(x_1,x_2) \mid x_1 \in E_1 \text{ et } x_2 \in E_2\}$

Rappels mathématiques
Relations

- Relations binaires : $R \in P(E_1 \times E_2)$, R est un ensemble de couples
- Relations n-aire : $R \in P(E_1 \times E_2 \times \ldots E_n)$

- Relations binaires
  - R réflexive $\Leftrightarrow \forall\, x : x\,R\,x$
  - R symétrique $\Leftrightarrow \forall\, x, y : x\,R\,y \Rightarrow y\,R\,x$
  - R antisymétrique $\Leftrightarrow \forall\, x, y : x\,R\,y \text{ et } y\,R\,x \Rightarrow x = y$
    $\Leftrightarrow \forall\, x, y : x\,R\,y \text{ et } x \neq y \Rightarrow (y, x) \notin R$
  - R transitive $\Leftrightarrow \forall\, x, y, z : x\,R\,y \text{ et } y\,R\,z \Rightarrow x\,R\,z$

  - Une relation réflexive, symétrique et transitive c'est … ?
  - Une relation réflexive, antisymétrique et transitive c'est … ?

Rappels mathématiques

Stabilité en clôture

- E : ensemble, R : relation n-aire sur E ($R \subset E^n$), E_1 : une partie de E
- E_1 stable par R ou close par R ssi

  $\forall\, x_1 \in E_1, x_2 \in E_1, \ldots, x_{n-1} \in E_1$ et $(x_1, x_2, \ldots, x_n) \in R$

  $\Rightarrow x_n \in E_1$
- Plus simple à voir pour R binaire
- Si E_1 non stable par R, il existe un plus petit sous ensemble F de E tel que $E_1 \subset F$ et F stable par R

  $\Rightarrow$ F : clôture de E_1 par R
- Soit b une relation sur D, c'est-à-dire $b \subset D^2$
  - Fermeture transitive de b : la plus petite relation binaire T telle que $b \subset T$ et T transitive

Rappels mathématiques

Fonctions – applications – bijections – cardinal

- Fonction f de E_1 vers E_2 : relation de E_1 vers E_2 telle que

  $\forall$ x $\in$ E_1, il existe au plus un élément y $\in$ E_2 tel que x f y

  - y : image de x par f : y = f(x)
  - sous-ensemble de E_1 des éléments ayant des images par f : domaine de f

- Composition de fonctions : $\circ$

  $f \circ g\,(x) = f(g(x))$ $\quad$ $E_1 \rightarrow (g) \rightarrow E_2 \rightarrow (f) \rightarrow E_3$

Rappels mathématiques
Fonctions – applications – bijections – cardinal

- Application f de E_1 vers E_2 : fonction telle que dom f = E_1

- Une application f est injective

  si $\forall\ x_1, x_2 \in E_1, f(x_1) = f(x_2) \Rightarrow x_1 = x_2$

- Une application f est surjective

  si $\forall\ y \in E_2$, il existe au moins un élément x de E_1

  tel que $f(x) = y$

- Une bijection est une application injective et surjective

  (Ou $f(E_1) = E_2$.)

Rappels mathématiques

Fonctions – applications – bijections – cardinal

- Deux ensembles sont équipotents ou ont même cardinal
  ssi il existe une bijection de l'un vers l'autre

- Un ensemble est fini s'il est équipotent à {1, 2, … n}
  pour tout entier n
- Un ensemble infini est un ensemble non fini

- On dit qu'un ensemble est infini dénombrable s'il est équipotent à $\mathbb{N}$
- S'il n'existe pas de bijection entre X et une partie de $\mathbb{N}$, alors on dit que X est infini non dénombrable

- Proposition *Il existe des ensembles infinis non dénombrables*

Alphabets et langages
Mot

- Alphabet : *ensemble* fini, non vide, de symboles

  Généralement noté Σ

- Mot ou chaine (ang. *string*) sur un alphabet Σ :

  suite finie d'éléments de Σ

- On note Σ^* l'ensemble de tous les mots (y compris le mot vide) définis sur Σ

Alphabets et langages
Mot

- Longueur : nombre de symboles d'un mot

- Deux mots u et v sont égaux ssi
  - Ils ont même longueur
  - $\forall\ i \in \{1, \ldots, |u|\} : u_i = v_i$

Alphabets et langages
Mot

- Concaténation de 2 mots u et v de Σ^* : mot noté uv et défini par :
  - $u = u_1u_2\ldots u_n$, $v = v_1v_2\ldots v_n$ → $w = u_1\ldots u_nv_1\ldots v_n$
    - $\forall\ i \in \{1, |u|\}$ $(uv)_i = u_i$
    - $\forall\ i \in \{|u|+1, \ldots\ |u|+|v|\}$ $(uv)_i = v_{i-|u|}$

- Propositions
  - la concaténation est régulière à droite et à gauche
    - $wu = wv \Rightarrow u = v$
    - $uw = vw \Rightarrow u = v$
  - $|uv| = |u| + |v|$

Alphabets et langages
Mot

- Facteur gauche de w : mot u tel que uv = w
- Facteur droit de w : mot v tel que uv = w
- Facteur de w : mot u tel que il existe v et v' tels que vuv' = w

- Miroir (Reverse)
  - Fonction miroir $\_^R : \Sigma^* \rightarrow \Sigma^*$ définie par récurrence :
    - w tq $|w| = 0$: $w^R = \varepsilon^R = \varepsilon$
    - w tq $|w| > 0$: $\exists\, a \in \Sigma$ tq $w = au$
      et $w^R = (au)^R = u^Ra$

- Propriété
  - $\forall\, u, v \in \Sigma^* : (uv)^R = v^Ru^R$

Alphabets et langages

Langage

- Langage sur Σ : *ensemble* de mots sur Σ

- Remarques de cardinalité
  - Σ fini
  - Σ^* infini dénombrable (rappel : dont on peut énumérer les éléments)
  - P (Σ^*) est infini non dénombrable

- Opérations sur les langages
  - $\cup$, $\cap$, $\neg$ (complément), $\backslash \Rightarrow$ comme d'habitude

    (complément : $\neg A = \Sigma^* \setminus A$)
  - Concaténation : $L_1 \subset \Sigma^*$, $L_2 \subset \Sigma^*$
    - $L = L_1.L_2$ ou L_1L_2

      est défini par $L = \{w \mid \exists\ w_1 \in L_1$ et $\exists\ w_2 \in L_2 : w = w_1w_2\}$
  - Clôture de Kleene (Kleene star) ou étoile de L
    - $L^* = \{w \in \Sigma^* \mid \exists\ k \in N, \exists\ w_1, w_2, \ldots, w_k \in L : w = w_1w_2\ldots w_k\}$

Représentation finie des langages

- Langages définis par
  - Éléments de base :
    - L'ensemble $\varnothing$
    - Le mot vide ε
    - Les singletons sur Σ
  - Opérations :
    - La concaténation de langages
    - La réunion de deux langages
    - La fermeture de Kleene

- De tels langages sont appelés langages rationnels

Représentation finie des langages

- Description et manipulation des langages rationnels
  - Génération — *Grammaires régulières*
    → Un moyen de générer un langage
  - Reconnaissance — *Automates finis*
    → Un moyen de décider l'appartenance
  - Caractérisation algébrique — *Expressions régulières*
    → Notations et équations — *Systèmes*

Représentation des langages

- Description et manipulation des langages

  – Génération *Grammaires*

    → Un moyen de générer un langage

  – Reconnaissance *Automates*

    → Un moyen de décider l'appartenance

  – Caractérisation algébrique

    → Notations et équations