

LifLF – Théorie des langages formels

Sylvain Brandel

2016 – 2017

sylvain.brandel@univ-lyon1.fr

Fonctionnement

- Supports fournis en avance
 - Supports de cours
 - Ce qui est projeté en CM
 - Sujets de TD
- CM
 - Compléments du support fourni
 - Evitez de découvrir le cours en cours ...
- TD
 - Plus une discussion autour des solutions que vous aurez cherchées avant de venir
 - Evitez de découvrir les exercices de TD en TD ...
- Bref, travail de votre part **AVANT** et **PENDANT**

Projet

- Diverses choses autour des automates à états finis
- Va s'appuyer sur JFLAP (cf. Google)
 - Plateforme de test d'un cours
 - Duke University, Trinity, Caroline du Nord, Etats-Unis
 - Pas tout récent
- A faire en binôme, ou en solo
- Séances de TP pour discuter de ce qui coince
- Langage : C / C++
 - CM supplémentaire le 4 octobre à 15H45 (R. Thion) commun avec LifLC
 - Utilise les conteneurs STL (vector, set ...)

Evaluation

- <http://liris.cnrs.fr/sylvain.brandel/wiki/doku.php?id=ens:lifLF>
- Le plan réussite en licence n'est plus
- Contrôles de CM / TD
 - 1 seul contrôle intermédiaire mardi 18 octobre 2016
 - 30% (poids à confirmer)
- Projet
 - A rendre en deux parties
 - Total : 35% (poids à confirmer)
- Contrôle continu final
 - En janvier, anonyme et tout
 - 35% (poids à confirmer)

De votre côté

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

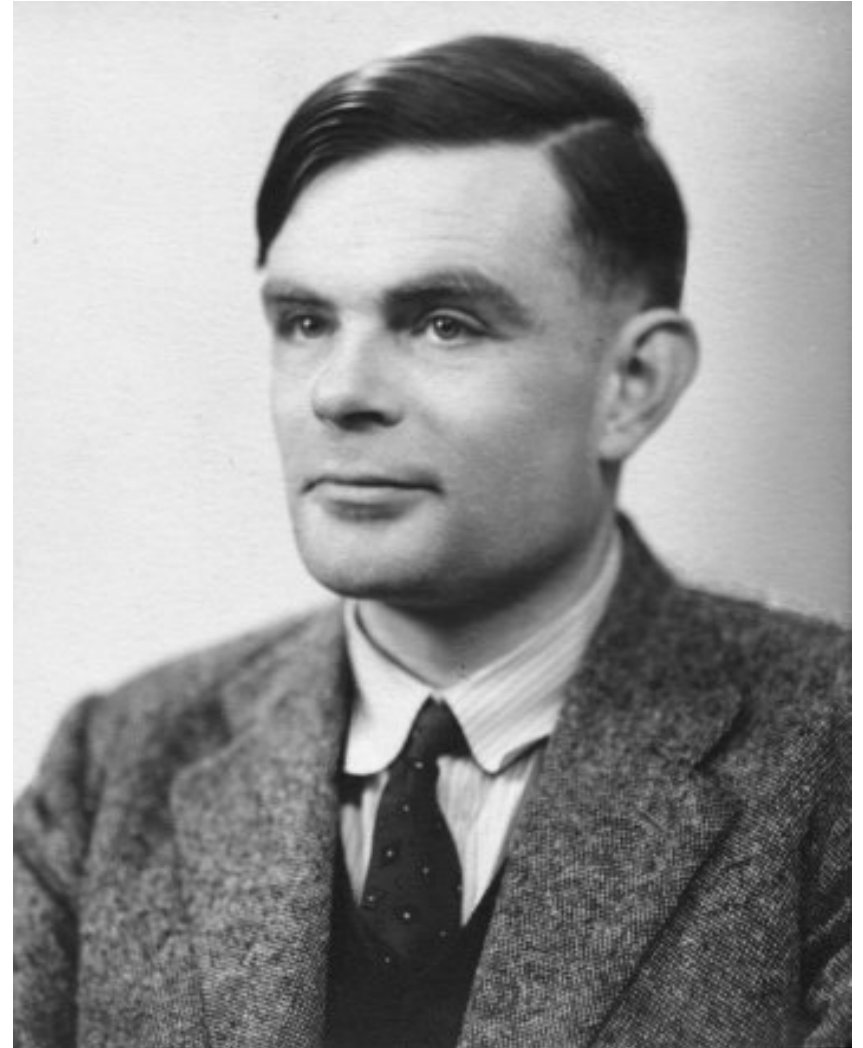
LifLF – Théorie des langages formels

Sylvain Brandel

2016 – 2017

sylvain.brandel@univ-lyon1.fr

INTRODUCTION



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
 - Compilation
- Vous intéresser ...
 - Si on sait qu'un problème est indécidable, inutile de chercher un algorithme pour le résoudre

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 rékursifs	machine de Turing	grammaire (générale)

- La décidabilité et la complexité en découlent

Programme

- Notions mathématiques de base
 - Ensembles
 - Alphabets, langages, expressions régulières
- Automates à états finis
 - Déterministes ou non
 - Liens avec les expressions rationnelles
 - Rationalité
 - Minimisation
- Langages algébriques
 - Grammaires algébriques
 - Automates à pile
 - Algébricité

Programme (suite)

- Machines de Turing
 - Formalisme de base
 - Langages rékursifs
 - 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 cours par cours

- CM1 et CM2 : Notions mathématiques de base
- CM3 : Automates à états finis déterministes ou non
- CM4 : Non déterminisme, élimination du non déterminisme
- CM5 : Caractérisation des langages rationnels
- CM6 : Rationalité
- CM7 : Minimisation
- CM8 : Grammaires algébriques et langages algébriques
- CM9 : Automates à pile
- CM10 : Algébricité

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