GUIDE IFT-2002

Professeur Pascal Tesson : pascal.tesson@ift.ulaval.ca

La matière du cours se divise en trois grands chapitres:

1 - Automates finis et langages réguliers

* Automates finis déterministes et langages réguliers

* Automates finis non-déterministes

* Langages non-réguliers incluant le lemme de pompage pour les langages réguliers

* Propriétés de clôture des langages réguliers (union, concaténation, fermeture sous étoile, intersection, complément)

* Expressions régulières

* Exercices typiques dans ce chapitre (liste non exhaustive) :

construire un automate déterministe ou non-déterministe pour un langage donné (possiblement spécifié par une expression régulière)

identifier le langage accepté par un automate donné

transformer un automate non-déterministe en un équivalent déterministe

démontrer qu'un langage n'est pas régulier

transformer un automate fini en expression régulière équivalente

2 - Automates à piles et grammaires non-contextuelles

* Automates à pile non-déterministes et déterministes

* Grammaires non-contextuelles, dérivations, arbres de dérivation

* Ambiguïté dans les grammaires non-contextuelles

* Transformation d'une grammaire non-contextuelle en automate à pile équivalent

* Langages contextuels et lemme de pompage pour les langages non-contextuels

* Exercices typiques dans ce chapitre (liste non exhaustive) :

Construire un automate à pile pour un langage donné

identifier le langage accepté par un automate à pile donné

Construire une grammaire non-contextuelle pour un langage donné

identifier le langage généré par une grammaire donnée

Montrer qu'une grammaire donnée est ambiguë

Montrer qu'un langage n'est pas non-contextuel

3 - Machines de Turing et décidabilité

* Machines de Turing (déterministes ou pas, avec k rubans et autres variantes)

* Concept de Turing-acceptabilité versus décidabilité

* Indécidabilité

* Exercices typiques dans ce chapitre (liste non exhaustive) :

Construire une machine de Turing pour un langage donné

identifier le langage accepté par une machine de Turing donnée

Démontrer qu'un langage donné est Turing-acceptable

Démontrer qu'un langage donné est décidable

Démontrer qu'un langage donné est indécidable

Si vous comptez faire l'examen d'IFT-2002, demandez à avoir accès au site du cours. Cela vous permettra de consulter dans la rubrique "Contenu et activités" les diapositives présentées en cours, les anciennes notes de cours (utiles pour les chapitres 1 et 2 seulement) et des bribes de nouvelles notes de cours (incluant en particulier, et c'est ce qui importe, une banque d'exercices dont une partie sont corrigés). Si vous êtes à l'aise en anglais (ou si vous ne lisez pas le français!), je recommande fortement le livre de Michael Sipser intitulé Introduction to the Theory of Computation. La matière d'IFT-2002 correspond aux 5 premiers chapitres du livre à quelques exceptions près : toute la fin du chapitre 2 sur les grammaires déterministes et les grammaires LR(k) (inclus seulement dans la troisième édition du livre!) et, dans le chapitre 5, la partie REDUCTIONS VIA COMPUTATION HISTORIES de 5.1 et le sous-chapitre 5.2.