

1 Contenu général et objectifs du cours :

Est-il possible d'analyser un programme afin

- de déterminer s'il bouclera à l'infini ?
- de borner le temps que prendra son exécution s'il ne boucle pas ?
- de suspecter la piraterie puisqu'il aura le même comportement qu'un programme connu ?

La théorie sous-jacente à l'informatique offre des balises en ce qui a trait à ce type de questions. La théorie permet également parfois de dire qu'un problème calculatoire donné est impossible à résoudre en un temps raisonnable, *quel que soit* le programme utilisé.

La théorie ne fournit pas que des réponses négatives. Elle rend aussi des choses possibles. Par exemple, ce sont des travaux théoriques sur les langages formels qui ont permis des progrès répétés dans le domaine de la compilation des langages de programmation, ou celui de la vérification formelle de programmes ou encore celui de la cryptographie.

Voici les principaux objectifs du cours IFT2105 :

- faire saisir la portée de l'étude des langages formels et de leur classification
- introduire la complexité du calcul et aborder une question centrale de l'informatique théorique actuelle, soit celle du lien entre les classes de complexité P et NP
- introduire la calculabilité, dont l'intérêt est à la fois philosophique, mathématique et informatique
- offrir un tremplin pour une étude plus approfondie.

En plus d'acquérir les connaissances factuelles ci-dessus, l'étudiant(e)

- adoptera un mode de pensée rigoureux,
- développera une plus grande aisance face au formalisme mathématique,
- saura prendre un certain recul face à l'outil qu'est l'ordinateur.

2 Bibliographie

Le livre de Sipser et certains livres de référence sont placés en réserve à la bibliothèque de math-info (pavillon Aisenstadt).

2.1 Livre fortement recommandé

Michael Sipser, *Introduction to the Theory of Computation*, ISBN 0-534-95097-3, attention il s'agit de la 2^{ème} édition, 2006.

2.2 Autres références

- John E. Hopcroft, Rajeev Motwani et Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3^{ème} édition, Addison-Wesley, 2007. ISBN : 0-321-46225-4.
- Pierre Wolper, *Introduction à la calculabilité*, 2^{ème} édition, Dunod 2001, ISBN 2-10-004853-8.

3 Évaluation : devoirs (30%), intra (30%) et final (40%)

Les deux examens seront à livre *fermé* et l'examen final portera sur la matière de toute la session. Il y aura 6 ± 1 devoirs à caractère théorique, sans programmation, dont les numéros ne seront pas nécessairement tous corrigés. Si la note moyenne aux examens n'atteint pas le seuil de 40%, le cours est échoué.

4 Rappels de la Faculté des arts et des sciences

- Comprenez bien le *Règlement disciplinaire sur le plagiat et la fraude* disponible à http://www.direction.umontreal.ca/secgen/pdf/reglem/francais/sec.30/ens30_3.pdf

5 Plan approximatif de la session

Heures de cours	Matière	Chapitre de Sipser
2	Introduction Motivation, révision de techniques de preuve, terminologie	
3	Fonctions primitives récursives Fonction d'Ackermann, fonctions récursives Programmes répéter, programmes tantque	
8	Langages réguliers Automates finis Expressions régulières	1
3	Langages hors-contextes	2
4 + 4	Machines de Turing et hypothèse de Church-Turing Complexité du calcul	3
EXAMEN INTRA		
8	Complexité du calcul (suite) P, NP, NP-complétude Sujets connexes	7,8
7	Indécidabilité Problème d'arrêt	4,5
EXAMEN FINAL :		
TOTAL 39		