

Université de
Sherbrooke

Département d'informatique
IFT 313 – Introduction aux langages formels
Plan d'activité pédagogique
Été 2020

Enseignant

Djemel Ziou

Courriel : djemel.ziou@usherbrooke.ca

Local : D4-1024-2

Téléphone : +1 819 821-8000 x62859

Disponibilités : Deux heures par semaine

Responsable(s) : Direction du département

Site web du cours : <http://Aucun>

Horaire

Exposé magistral :	Lundi	13h30 à 15h20	salle D3-2040
	Jeudi	10h30 à 12h20	salle D3-2040
Exercices/laboratoires :	Jeudi	11h30 à 12h20	salle À venir

Description officielle de l'activité pédagogique¹

Cibles de formation :	S'initier aux fondements théoriques des langages de programmation, en particulier aux langages formels, à la théorie des automates ainsi qu'à l'analyse lexicale et syntaxique.
Contenu :	Langages réguliers et expressions régulières. Automates finis et analyseurs lexicaux. Langages et grammaires hors contexte. Arbre syntaxique et grammaire ambiguë. Automates à pile de mémoire, analyseurs syntaxiques descendants et analyseurs syntaxiques ascendants. Machines caractéristiques. Classes de grammaires hors contexte : LL, SLR, LALR et LR. Applications aux langages de programmation. Générateurs d'analyseurs lexicaux et syntaxiques.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure d'exercices par semaine 5 heures de travail personnel par semaine
Préalable	MAT 115
Particularités	Aucune

¹<https://www.usherbrooke.ca/admission/fiches-cours/ift313>

1 Présentation

Cette section présente les cibles de formation spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation du comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

On sait que les langages de programmation occupent une place prépondérante en informatique. Par leur structure et leurs particularités, ils exercent une influence certaine sur le style de programmation et par conséquent sur la conception des programmes. Par exemple, le langage *Prolog* est conçu pour faire de la programmation logique, le langage *Eiffel* de la programmation orientée objet, le langage *Scheme* de la programmation fonctionnelle, le langage *Occam* de la programmation parallèle et enfin le langage *C* de la programmation procédurale et impérative.

Les langages de programmation sont loin d'être les seuls langages utilisés en informatique. En effet, il arrive souvent qu'il faille exprimer des instructions ou des spécifications de manière formelle, par exemple pour décrire un système complexe (langages de spécification), faire exécuter une suite de tâches (langages de commandes), interroger une base de données (langages de quatrième génération), voire formater un texte ou utiliser un logiciel de calcul symbolique (langages spécialisés).

Tous ces langages ont plusieurs points en commun. D'abord, de la même manière que les langues naturelles peuvent être vues comme des ensembles de phrases constituées de mots, on peut définir ces langages comme des ensembles de suites de lexèmes qui obéissent à des *règles lexicales* (dans le cas des langages de programmation, les lexèmes sont les constantes numériques, les noms de variables, les mots-clés et les opérateurs comme + et =). Ensuite, on distingue les phrases valides des autres à l'aide d'un ensemble de *règles syntaxiques*, appelé *grammaire*.

Définir un langage de manière rigoureuse et formelle permet l'analyse automatique de programmes ou de textes écrits dans ce langage par un outil qui, selon le type de langage, est appelé un compilateur, un assembleur, un interpréteur, un préprocesseur ou un éditeur. Par exemple, un programme écrit en *Java* est soumis à un compilateur, qui vérifie s'il est conforme aux règles syntaxiques du langage *Java*; si c'est le cas, le compilateur pourra traduire le programme en *bytecode*, ce qui lui permettra d'être exécuté sur une machine virtuelle *Java*.

L'activité pédagogique intitulée *Introduction aux langages formels* présente les principaux outils formels de description de langages et de mise en oeuvre d'analyseurs lexicaux et syntaxiques. Ces outils sont basés sur la théorie des automates et des langages formels.

1.2 Cibles de formation spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable :

1. de comprendre et d'expliquer les principales notions liées à la définition formelle de langages de programmation ;
2. de comprendre et d'expliquer les principales méthodes de spécification et d'analyse lexicale et syntaxique ;
3. d'appliquer les notions vues en cours à des exemples concrets ;
4. d'utiliser des outils d'écriture automatique d'analyseurs lexicaux et syntaxiques.

1.3 Contenu détaillé

Thème	Contenu	Nbr. d'heures	Objectifs	Travaux	Lectures
1	Introduction : Présentation du domaine des automates et des langages formels ; présentation du plan de cours.	2			
2	Langage régulier : Alphabet, chaîne et langage ; opérations sur les chaînes et les langages ; façons de définir formellement un langage ; ensemble régulier et expressions régulières ; exemples concrets à partir de langages de programmation.	6	1, 3	✓	Chap. 2 de [7]
3	Automate fini déterministe : Rappel de la notion d'automate fini déterministe, brève introduction à la notion de non déterministe, exemples d'automates finis ; algorithmes de traduction directe d'une expression régulière en un automate fini déterministe.	6	1	✓	Sec. 5.1 à 5.3 de [7]
4	Construction d'analyseurs lexicaux : introduction à un outil de construction automatique d'analyseurs lexicaux.	2	4	✓	[2] ou [6]
5	Grammaire hors-contexte : Notion de grammaire hors contexte, exemples concrets de langages hors contexte ; notions de dérivation, d'arbre de dérivation et de grammaire ambiguë ; transformation d'une grammaire ambiguë en une grammaire non ambiguë ; formes de règles de production et formes de grammaire hors-contexte ; grammaires régulières et grammaires linéaires.	8	1, 3	✓	Chap. 3 (sauf sec. 3.4) et survol du chap. 4 de [7]
6	Du lexical au syntaxique : Frontière entre l'analyse lexicale et syntaxique, lemme de l'étoile.	1	2	✓	Sec. 6.6 de [7]
7	Introduction à l'analyse syntaxique : Graphe d'une grammaire ; analyse descendante et analyse ascendante à l'aide d'une recherche en largeur.	3	2	✓	Chap. 18 de [7]
8	Automate à pile : Notion d'automate à pile, types d'automate à pile et leur équivalence.	4	1	✓	Sec. 7.1 à 7.3 de [7]
9	Analyse syntaxique descendante : Notion de grammaire LL(\$k\$) ; calcul des ensembles <i>First</i> , <i>Follow</i> et calcul des <i>lookaheads</i> ; construction d'analyseurs syntaxiques descendants, descente réursive ; transformation d'une grammaire non LL(\$k\$) en une grammaire LL(\$k\$) ; introduction à un outil de construction automatique d'analyseurs syntaxiques descendants.	8	2, 4	✓	Chap. 19 de [7], [1] ou [4]
10	Analyse syntaxique ascendante : Notion de grammaire LR(\$k\$) et notions d'items ; construction d'analyseurs syntaxiques ascendants ; grammaires LR(0), SLR(1), LR(1), LALR(1) ; transformation d'une grammaire non LR(\$k\$) en une grammaire LR(\$k\$) ; introduction à un outil de construction automatique d'analyseurs syntaxiques ascendants.	8	2, 4	✓	Chap. 20 de [7], [3] ou [5]

1. Le cours doit comprendre au moins quatre travaux pratiques couvrant tous les sujets marqués «✓» dans le tableau.
2. Les lectures indiquées ne sont là qu'à titre indicatif. L'enseignant est libre de choisir un autre document de référence.

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Une semaine comporte quatre heures de présence en classe réparties dans une proportion de trois heures de cours magistral et d'une heure d'exercices. Les exercices sont faits par les étudiantes ou les étudiants, autant que possible. Toutes les notions théoriques présentées dans les cours magistraux sont illustrées à l'aide d'études de cas puisées à partir de langages de programmation comme *C* ou *C++*, *Python* et pourquoi pas *COBOL* et *Fortran*. L'approche pédagogique sera adaptée en fonction de l'évolution de la maladie du COVID-19. Je tiens à préciser que le calendrier de la session donné ci-dessous est approximatif.

2.2 Calendrier

Semaine	Date	Thème	Lectures
1	2020-05-04	1 et 2	notes de cours et chapitre 2 de
2	2020-05-11	2	notes de cours et chapitre 2 de
3	2020-05-18	3	notes de cours et sections 5.1 à 5.3 de
4	2020-05-25	3 et 4	notes de cours et sections 5.1 à 5.3 de
5	2020-06-01	5 et 6	notes de cours et chapitre 3 de
6	2020-06-08	5	notes de cours et chapitres 3 et 4, section 6.6 de
7	2020-06-15	1, 2, 3, 4 et 5	notes de cours, révision des chapitres et sections lus de
9	2020-06-29		examen périodique
10	2020-07-06	8	notes de cours et sections 7.1 à 7.3 de
11	2020-07-13	7 et 9	notes de cours et chapitres 18 et 19 de
12	2020-07-20	9	
13	2020-07-27	10	note de cours et chapitre 20 de
14	2020-08-03	10	note de cours et chapitre 20 de
15	2020-08-10	Révision	

2.3 Évaluation

Devoirs	30 %
Examen intra	30 %
Examen final	40 %

Dans le cas de confinement causé par le COVID-19 et si les règles le permettent, le plan de cours sera adapté. Par exemple, je pourrais donner des tests courts au lieu des deux examens périodique et final.

Les directives, la date de remise et le barème relatifs aux devoirs seront connus à la remise de l'énoncé de chaque devoir aux étudiantes et aux étudiants.

2.3.1 Qualité de la langue et de la présentation

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages² l'enseignante ou l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

²https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/Etudiants_actuels/Informations_academiques_et_reglements/2017-10-27_Reglement_facultaire_-_evaluation_des_apprentissages.pdf

2.3.2 Plagiat

Le plagiat consiste à utiliser des résultats obtenus par d'autres personnes afin de les faire passer pour sien et dans le dessein de tromper l'enseignante ou l'enseignant. Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 9.4.1 du Règlement des études³. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 9.4.1 du Règlement des études de l'Université de Sherbrooke. L'étudiante ou l'étudiant peut s'exposer à de graves sanctions qui peuvent être soit l'attribution de la note E ou de la note zéro (0) pour un travail, un examen ou une activité évaluée, soit de reprendre un travail, un examen ou une activité pédagogique. Tout travail suspecté de plagiat sera transmis au Secrétaire de la Faculté des sciences. Ceci n'indique pas que vous n'avez pas le droit de coopérer entre deux équipes, tant que la rédaction finale des documents et la création du programme restent le fait de votre équipe. En cas de doute de plagiat, l'enseignante ou l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'elle ou qu'il considère comme étant plagié. En cas d'incertitude, ne pas hésiter à demander conseil et assistance à l'enseignante ou l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

Les dates de remise des travaux seront indiquées sur les énoncés.

2.4.1 Directives particulières

Des devoirs sont données aux étudiantes et aux étudiants pour qu'ils assimilent les concepts présentés dans le cours. Les devoirs peuvent être faits individuellement ou par équipe de deux personnes. Les devoirs comportent entre autres l'écriture d'analyseurs lexicaux et d'analyseurs syntaxiques nécessitant de la programmation C. Aucun devoir ne peut être remis par courriel. Les devoirs non remis reçoivent automatiquement la note zéro. La correction des devoirs et des examens est entre autres basée sur le fait que chacune de vos réponses soit :

- claire, c'est-à-dire lisible et compréhensible pour le correcteur ;
- précise, c'est-à-dire exacte ou sans erreur ;
- complète, c'est-à-dire que toutes les étapes de résolution du problème sont présentes ;
- concise, c'est-à-dire que la méthode de résolution est la plus courte possible.

Les mêmes critères seront utilisés aussi pour la correction des examens.

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁴, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à condition que leur usage soit explicitement permise dans le plan de cours.

Dans ce cours le règlement 4.2.3 s'applique à moins d'avoir obtenu personnellement l'autorisation de la personne enseignante. Cette permission peut être retirée en tout temps, si l'appareil n'est pas utilisé uniquement à des fins d'apprentissage.

Tel qu'indiqué dans le règlement universitaire des études, section 4.2.3⁵, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission de la personne enseignante.

Note : Je ne réponds à aucun courriel.

Ces règles pourraient changer en fonction de l'évolution de la maladie COVID-19.

3 Matériel nécessaire pour l'activité pédagogique

Aucun

³<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

⁴https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Intranet/Informations_academiques/Sciences_Reglement_complementaire_2017-05-09.pdf

⁵<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

4 Références

- [1] J. GROSCH : The Parser Generator Ell. Rapport technique Document No. 8, CoCoLab - Datenverarbeitung, 1998. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [2] J. GROSCH : Rex - A Scanner Generator. Rapport technique Document No. 5, CoCoLab - Datenverarbeitung, 2000. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [3] J. GROSCH : Lark - An LALR(2) Parser Generator with Backtracking. Rapport technique Document No. 32, CoCoLab - Datenverarbeitung, 2002. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [4] JAVACC : A Parser/Scanner Generator for Java. <http://javacc.dev.java.net/>.
- [5] JAVACUP : LALR Parser Generator for Java. <http://www.cs.princeton.edu/appel/modern/java/CUP/>.
- [6] JFLEX : The Fast Scanner Generator for Java. <http://jflex.de/>.
- [7] T. A. SUDKAMP : *Languages and Machines : An Introduction to the Theory of Computer Science*. Addison-Wesley, 2005. Manuel disponible à la Coop de l'Université de Sherbrooke et en réserve à la bibliothèque.



L'intégrité intellectuelle passe, notamment, par la reconnaissance des sources utilisées. À l'Université de Sherbrooke, on y veille!

Extrait du Règlement des études (Règlement 2575-009)

9.4.1 DÉLITS RELATIFS AUX ÉTUDES

Un délit relatif aux études désigne tout acte trompeur ou toute tentative de commettre un tel acte, quant au rendement scolaire ou une exigence relative à une activité pédagogique, à un programme ou à un parcours libre.

Sont notamment considérés comme un délit relatif aux études les faits suivants :

- a) commettre un plagiat, soit faire passer ou tenter de faire passer pour sien, dans une production évaluée, le travail d'une autre personne ou des passages ou des idées tirés de l'œuvre d'autrui (ce qui inclut notamment le fait de ne pas indiquer la source d'une production, d'un passage ou d'une idée tirée de l'œuvre d'autrui);
 - b) commettre un autoplagiat, soit soumettre, sans autorisation préalable, une même production, en tout ou en partie, à plus d'une activité pédagogique ou dans une même activité pédagogique (notamment en cas de reprise);
 - c) usurper l'identité d'une autre personne ou procéder à une substitution de personne lors d'une production évaluée ou de toute autre prestation obligatoire;
 - d) fournir ou obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour une production faisant l'objet d'une évaluation;
 - e) obtenir par vol ou toute autre manœuvre frauduleuse, posséder ou utiliser du matériel de toute forme (incluant le numérique) non autorisé avant ou pendant une production faisant l'objet d'une évaluation;
 - f) copier, contrefaire ou falsifier un document pour l'évaluation d'une activité pédagogique;
- [...]

Par plagiat, on entend notamment :

- Copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets;
- reproduire des présentations, des dessins, des photographies, des graphiques, des données... sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire;
- utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources;
- résumer ou paraphraser l'idée d'un auteur sans en indiquer la source;
- traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets ;
- utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord);
- acheter un travail sur le Web ou ailleurs et le faire passer pour sien;
- utiliser sans autorisation le même travail pour deux activités différentes (autoplagiat).

Autrement dit : mentionnez vos sources
