

UNIVERSITÉ DE
SHERBROOKE**Département d'informatique****IFT 580 – Compilation et interprétation des langages****Plan de cours
Hiver 2014**

Enseignant**Richard St-Denis**

Courriel :	Richard.St-Denis@USherbrooke.ca
Local :	D4-2005
Téléphone :	(819) 821-8000 poste 62847
Site :	http://info.usherbrooke.ca/rstdenis/
Disponibilité :	horaire de consultation affiché sur ma porte

Horaire

Cours magistral :	mardi	8 h 30 à 10 h 20	salle D4-2022
Présentation de l'étude de cas :	mercredi	15 h 30 à 16 h 20	salle D4-2022

Description officielle de l'activité pédagogique ¹

Objectifs	S'initier à l'écriture de compilateurs et d'interpréteurs de langages de programmation, en particulier à l'analyse sémantique ainsi qu'à la génération et à l'optimisation de code.
Contenu	Organisations générales de compilateurs et d'interpréteurs. Interpréteurs itératifs et récursifs. Grammaires attribuées. Évaluation d'attributs. Classes de grammaires attribuées. Analyse sémantique: gestion des symboles, vérification et inférence de types, allocation d'adresses, organisation de l'espace des données, traitement des énoncés de contrôle et des appels de fonctions. Code intermédiaire et machine virtuelle. Techniques de génération de code. Introduction à l'optimisation de code. Construction d'un compilateur à l'aide de générateurs de compilateurs.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 6 heures de travail personnel par semaine
Préalable	IFT 313 et IFT 339
Particularités	Cette activité pédagogique se présente sous la forme de cours magistraux.

1. www.usherbrooke.ca/fiches-cours/ift580

1 Présentation

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

1.1 Mise en contexte

La construction de compilateurs est une discipline de l'informatique. Elle possède sa propre terminologie, ses propres normes de conception, ainsi que ses propres théories et techniques mises au point depuis maintenant 45 ans². C'est pour cette raison que les compilateurs sont aujourd'hui d'une très grande efficacité et d'une haute fiabilité. Cela n'est pas très surprenant, car les compilateurs constituent un outil de travail essentiel pour les informaticiens.

Il existe aussi plusieurs outils qui s'apparentent aux compilateurs et qui requièrent les mêmes techniques de base pour leur mise en oeuvre. Mentionnons à titre d'exemples les assembleurs, les interpréteurs, les préprocesseurs et les éditeurs syntaxiques contextuels.

L'activité pédagogique intitulée *Compilation et interprétation des langages* est la deuxième activité portant sur l'étude formelle des langages de programmation. Elle complète l'activité pédagogique *Introduction aux langages formels* qui introduit des outils de description de langages et de mise en oeuvre d'analyseurs lexicaux et syntaxiques basés sur la théorie des automates et des langages formels. Elle intègre plusieurs aspects des activités pédagogiques suivantes: *Programmation système* (langage machine, langage d'assemblage et architecture des ordinateurs), *Structures de données* (schémas de découpage, gestion de mémoire et ramasse-miettes), *Algorithmes et structures de données* (tri topologique, programmation dynamique et coloriage de graphes) et *Intelligence artificielle* (unification).

Notons que, contrairement à l'activité pédagogique *Introduction aux langages formels*, celle dédiée à la compilation et à l'interprétation des langages n'est pas un cours théorique. L'importance est mise sur l'étude d'un petit compilateur (programme C d'environ 21 000 lignes généré à partir d'une spécification de haut niveau d'environ 6 000 lignes) qui sera modifié par les étudiants dans la deuxième moitié de la session.

1.2 Objectifs spécifiques

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

1. de formuler des règles sémantiques à l'aide de grammaires attribuées;
2. de comprendre différentes techniques d'évaluation d'attributs;
3. de comprendre les principaux traitements sémantiques relatifs aux langages impératifs et procéduraux;
4. de connaître et comprendre les principales techniques de génération de code;
5. de connaître et comprendre quelques techniques d'optimisation de code;
6. de connaître et comprendre deux formes de mise en oeuvre d'interpréteurs;
7. d'utiliser des systèmes d'écriture automatique de compilateurs.

2. M. Jackson, Loyal opposition: specializing in software engineering, *IEEE Software*, 16 (6), 1999.

1.3 Contenu détaillé

Le tableau suivant contient la matière présentée dans ce cours. L'étude de cette matière est accompagnée d'au moins quatre devoirs couvrant tous les sujets, dont au moins trois travaux pratiques utilisant des outils de construction automatique d'analyseurs sémantiques et de compilateurs.

Thème	Contenu	Nb d'heures	Objectifs	Références
1	Introduction: présentation du domaine des compilateurs; présentation du plan de cours; quelques rappels sur l'analyse lexicale et syntaxique.	3		
2	Grammaire attribuée: grammaire S-attribuée, L-attribuée, sans cycle, fortement sans cycle, ordonnée. Évaluation d'attributs: graphe de dépendance, tri topologique, évaluation dirigée par la syntaxe, parcours d'arbres et de graphes. Classification des grammaires attribuées.	8	1,2	chap. 5 de [1] chap. 3 de [13]
3	Analyse sémantique et génération de code intermédiaire: gestion des symboles; vérification et inférence de types; allocation d'adresses; organisation de l'espace de données; traitement des énoncés de contrôle; traitement des appels de fonctions; graphe orienté acyclique pour expressions; code des instructions; machine virtuelle.	6	1,2,3	chap. 6 de [1]
4	Technique de génération de code: blocs de base, optimisation des blocs de base; allocation et affectation de registres; optimisation à la lucarne; génération de code par programmation dynamique.	6	4	chap. 8 de [1]
5	Optimisation de code: sources principales d'optimisation; introduction à l'analyse de flot de données.	6	5	chap. 9 de [1]
6	Interpréteurs: interpréteurs itératifs et interpréteurs récursifs.	2	6	chap 4 de [13]
7	Construction d'un compilateur: boîte à outils de construction de compilateurs.	9	7	note de cours [2]

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 trois heures de présence en classe réparties dans une proportion de deux heures de présentation de notions dites théoriques et de une heure consacrée à une étude de cas, celui d'un compilateur d'un sous-ensemble du langage *java*.

2.2 Calendrier

Semaine	Thèmes	Lecture	Cas d'études	Devoir
1	1	notes de cours et chapitre 1 de [1]	boîte à outils [3,10,12]	remise de tous les devoirs
2	2,7	sections 5.1 à 5.3 de [1]	analyse lexical [4-6]	
3	2,7	sections 5.4 et 5.5 de [1]	analyse syntaxique [9]	
4	2,7	section 3.1 de [13]	analyse sémantique [7,8]	
5	2,7	section 3.2 de [13]	analyse sémantique [7,8]	
6	3,7	sections 6.1 à 6.9 de [1]	machine virtuelle <i>java</i> [2]	
7	3,7	sections 6.1 à 6.9 de [1]	génération de code [11]	
8	examen périodique			
9	4,7	sections 8.1 à 8.9 de [1]	étude de cas [2]	
10	4,7	sections 8.10 à 8.12 de [1]	étude de cas [2]	
11	5,7	section 9.1 de [1]	étude de cas [2]	
12	5,7	section 9.2 de [1]	étude de cas [2]	
13	6	section 4.1 de [13]	étude de cas [2]	
14-15	examen final			

2.3 Évaluation

Devoirs	30%
Examen périodique	30%
Examen final	40%
Qualité du français	0%

Toute documentation est permise aux examens. Toutefois, les appareils électroniques (baladeur numérique, téléphone cellulaire, calculatrice, portable, etc.) sont interdits pendant les examens.

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages

(<http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-d-evaluation/>),

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.

Toute situation de plagiat sera traitée en conformité, entre autres, avec l'article 8.1.2 du *Règlement des études* de l'Université de Sherbrooke disponible à l'adresse

<http://www.usherbrooke.ca/programmes/references/reglement>

Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 8.1.2.

2.4 Échéancier des travaux

Les directives, les dates de remise et les barèmes relatifs aux devoirs seront précisés dans la série d'exercices remise au début de la session.

Directives particulières : Une série d'exercices pratiques est donnée au début de la session aux étudiantes et aux étudiants pour qu'ils assimilent les concepts présentés dans le cours. Un nombre de points a été attribué à chacun. Il s'agit d'accumuler le maximum de points jusqu'à concurrence de 30 points. Il est possible d'accumuler 10 points supplémentaires pour certains exercices plus difficiles. Ces points seront cumulés avec ceux de l'examen final.

Les devoirs peuvent être faits individuellement ou par équipe de deux personnes. Les devoirs comportent entre autres l'écriture d'analyseurs sémantiques et la modification d'un compilateur. **Aucun devoir ne peut être remis par courrier électronique. 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.

3 Matériel nécessaire pour le cours

L'étude de cas est décrit dans un rapport technique [2]. Le manuel de Aho et al. est fortement conseillé [1]. Le manuel de Grune [13] est complémentaire. Les manuels techniques [3–12] contiennent une description détaillée de la boîte à outils *cocktail*. Tout le matériel présenté en classe se trouve dans le répertoire

P:\Public\Cours\IFT580

4 Références

- [1] A. V. Aho, M. S. Lam, R. Sethi, and L. D. Ullman. *Compilers : Principles, Techniques, and Tools, second edition*. Addison-Wesley, 2007. La première édition de ce manuel disponible à la bibliothèque.
- [2] D. Côté and R. St-Denis. *Compilateur du langage java--*. Technical report, Université de Sherbrooke, 2006. Disponible dans le répertoire P:\Public\Cours\IFT580\java--.
- [3] J. Grosch. *Specification of a minilax-interpreter*. Technical Report Document No. 22, CoCoLab — Datenverarbeitung, 1997. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [4] J. Grosch. *Preprocessors*. Technical Report Document No. 24, CoCoLab — Datenverarbeitung, 2000. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [5] J. Grosch. *Rex — A scanner generator*. Technical Report Document No. 5, CoCoLab — Datenverarbeitung, 2000. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [6] J. Grosch. *Selected examples of scanner specifications*. Technical Report Document No. 7, CoCoLab — Datenverarbeitung, 2000. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [7] J. Grosch. *Ag — An attribute evaluator generator*. Technical Report Document No. 16, CoCoLab — Datenverarbeitung, 2002. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [8] J. Grosch. *Ast — A generator for abstract syntax trees*. Technical Report Document No. 15, CoCoLab — Datenverarbeitung, 2002. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [9] J. Grosch. *Lark — An lalr(2) parser generator with backtracking*. Technical Report Document No. 32, CoCoLab — Datenverarbeitung, 2002. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [10] J. Grosch. *Toolbox introduction*. Technical Report Document No. 25, CoCoLab - Datenverarbeitung, 2002. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [11] J. Grosch. *Puma — A generator for the transformation of attributed trees*. Technical Report Document No. 26, CoCoLab — Datenverarbeitung, 2004. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.

- [12] J. Grosch and H. Emmelmann. A tool box for compiler construction. Technical Report Document No. 20, CoCoLab — Datenverarbeitung, 1990. Disponible dans le répertoire `opt/cocktail-0210/doc.pdf`.
- [13] D. Grune, H. E. Bal, C. J. H. Jacobs, and K. G. Langendoen. *Modern Compiler Design*. Wiley, 2000. Manuel disponible 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

8.1.2 Relativement aux activités pédagogiques

L'expression délit désigne d'abord tout acte ou toute manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique.

Sans restreindre la portée générale de ce qui précède, est considéré comme un délit :

- a) la substitution de personnes ou l'usurpation d'identité lors d'une activité évaluée ou obligatoire;
- b) le plagiat, soit le fait, dans une activité évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui;
- c) l'obtention par vol ou par toute autre manœuvre frauduleuse de document ou de matériel, la possession ou l'utilisation de tout matériel non autorisé avant ou pendant un examen ou un travail faisant l'objet d'une évaluation;
- d) le fait de fournir ou d'obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour un examen ou un travail faisant l'objet d'une évaluation;
- e) le fait de soumettre, sans autorisation préalable, une même production comme travail à une deuxième activité pédagogique;
- f) la falsification d'un document aux fins d'obtenir une évaluation supérieure dans une activité ou pour l'admission à un programme.

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 (autoplégat)

Autrement dit : mentionnez vos sources.
