

Théorie et construction des compilateurs

Groupe 20

Mardi, de 13h30 à 16h30 PK-6605 (cours)

Mercredi, de 9h30 à 12h30 PK-S1535 (atelier)

Responsable(s) du cours

Nom du coordonnateur : GAGNON, Étienne M.**Nom de l'enseignant :** GAGNON, Étienne M.**Local :** PK-4730**Téléphone :** (514) 987-3000 #8215**Courriel :** gagnon.etienne_m@uqam.ca**Site Web :** info.uqam.ca/~egagnon

Description du cours

Familiariser les étudiants avec les principes et techniques de base de la compilation et avec certains outils de traitement des langages. Grammaires et langages: expressions régulières, grammaires non contextuelles, grammaires attribuées et schémas de traduction. Méthodes d'analyse lexicale et syntaxique (descendante vs ascendante). Outils pour le traitement des langages (lex/yacc, antlr). Vérifications contextuelles: table des symboles et règles de portée, vérification des types. Environnement d'exécution: organisation et gestion de la mémoire, traitement des accès non locaux, passage des paramètres. Introduction à la génération et à l'optimisation de code. Travaux en laboratoire.

Préalables académiques :

INF3105 Structures de données et algorithmes

Objectifs du cours

Pendant ce cours, l'étudiant apprendra à :

- Comprendre le fonctionnement des compilateurs et des interpréteurs.
- Maîtriser l'utilisation d'outils modernes d'analyse lexicale et syntaxique.
- Comprendre plus spécifiquement les techniques d'analyse lexicale (automates, expressions régulières) et d'analyse syntaxique (LR).
- Maîtriser les techniques de vérification sémantique et de génération de code.
- Maîtriser l'utilisation de techniques de programmation objet pour travailler sur des arbres syntaxiques hétérogènes.

À la fin de la session, l'étudiant devrait être en mesure de :

- Réaliser des interpréteurs et des compilateurs pour de petits langages.
- Réaliser de petits analyseurs lexicaux et syntaxiques pour lire divers types de fichiers tels que les fichiers de configuration.
- Utiliser les expressions régulières et les grammaires dans les divers outils et langages de programmation qu'il rencontrera.

Contenu du cours

Ce cours avancé couvre les problématiques théoriques et techniques du fonctionnement et du développement de compilateurs et d'interpréteurs.

Une importance particulière est mise sur le développement et la programmation ; à ce titre, il est attendu que les étudiants aient un très bon niveau en programmation.

La première partie du cours (avant l'examen intra) est davantage théorique, et couvre les thèmes suivants :

- révision des langages réguliers, expressions régulières, automates finis
- analyse lexicale
- grammaires, arbres syntaxiques, et ambiguïté
- Interprétation simple
- vérifications et analyses sémantiques

La seconde partie du cours (après l'examen intra) est davantage pratique, et couvre les thèmes suivants :

- compilation
- automates à piles et analyses syntaxique
- Optimisation de base

Les séances de laboratoire font partie intégrante du cours.

Modalités d'évaluation

- Examen intra (25%)
- Examen final (25%)
- Devoir : (15%)
- Projet de session (35%) : Livrable intermédiaire (10%) et livrable final (25%)

L'examen intra couvre la matière des semaines 1 à 7 inclusivement. L'examen final couvre toute la matière, avec une plus grande emphase sur la matière couverte après l'examen intra. L'utilisation de documents n'est pas permise aux examens. Une moyenne d'au moins 50% aux examens est nécessaire pour réussir le cours. Cette moyenne n'est pas suffisante en soi pour réussir le cours.

Une note de 0% est attribuée aux travaux pratiques remis en retard, à moins d'une entente préalable avec le professeur. Aucun travail pratique n'est accepté à partir du moment qu'une solution (même partielle) du travail est présentée par le professeur.

À moins d'avis contraire, les travaux pratiques doivent être réalisés en équipes de deux étudiants et la composition de l'équipe doit rester la même tout le long du travail. Un étudiant peut choisir de travailler individuellement, mais il ne bénéficiera d'aucune considération particulière pour son effort additionnel.

Politique d'absence aux examens

L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen – par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : PK-3150 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs. Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

Calendrier détaillé du cours

Le calendrier suivant n'est qu'un guide approximatif des sujets abordés.

1. Introduction
2. Expressions régulières, Langages réguliers.
3. Automates. Transformation d'expression régulière en automate fini non déterministe. Transformation d'automate fini non déterministe en automate fini déterministe.
4. Réalisation d'un analyseur lexical avec un automate fini déterministe. Priorité de jetons. Erreurs lexicales.
5. Analyse syntaxique. Grammaires non contextuelles et arbres syntaxiques. Ambiguïté. Priorité d'opérateur.
6. Écriture de grammaires. Arbre syntaxique concret et hiérarchie de classes. Visiteurs. Évaluateur simple.
7. Mini interpréteur.
8. Vérifications sémantiques. Tables de symboles.
9. Pile d'exécution. Interpréteur avancé.
10. Génération de code. Compilateur vers Java.
11. Automates à pile. Analyseur syntaxique LR. Automate LR(0). Conflits LR(0).
12. Automate LR(1). Conflits LR(1). Résolution de conflits.
13. Optimisations et machines virtuelles.

Intégrité académique

PLAGIAT Règlement no 18 sur les infractions de nature académique. (extraits)

Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une infraction au sens de ce règlement.

La liste non limitative des infractions est définie comme suit :

- la substitution de personnes;
- l'utilisation totale ou partielle du texte d'autrui en la faisant passer pour sien ou sans indication de référence;
- la transmission d'un travail pour fins d'évaluation alors qu'il constitue essentiellement un travail qui a déjà été transmis pour fins d'évaluation académique à l'Université ou dans une autre institution d'enseignement, sauf avec l'accord préalable de l'enseignante, l'enseignant;
- l'obtention par vol, manoeuvre ou corruption de questions ou de réponses d'examen ou de tout autre document ou matériel non autorisés, ou encore d'une évaluation non méritée;
- la possession ou l'utilisation, avant ou pendant un examen, de tout document non autorisé;
- l'utilisation pendant un examen de la copie d'examen d'une autre personne;
- l'obtention de toute aide non autorisée, qu'elle soit collective ou individuelle;
- la falsification d'un document, notamment d'un document transmis par l'Université ou d'un document de l'Université transmis ou non à une tierce personne, quelles que soient les circonstances;
- la falsification de données de recherche dans un travail, notamment une thèse, un mémoire, un mémoire-crédation, un rapport de stage ou un rapport de recherche;
- Les sanctions reliées à ces infractions sont précisées à l'article 3 du Règlement no 18.

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html> et <http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

Médiagraphie

VO Appel and Palsberg -- *Modern Compiler Implementation in Java, second edition* -- **Cambridge University Press, 2002. ISBN 0-521-82060-X.**

VC Aho et al. -- *Compilers, Principles, Techniques, and Tools, second edition* -- **Pearson Education Inc., 2007. ISBN 0-321-48681-1.**

A : article - C : comptes rendus - L : logiciel
S: Standard - U : uri - V : volume

C : complémentaire - O : Obligatoire - R : recommandé