PLAN DE COURS
Été 2018

UQAC

8INF958 • Spécification, test et vérification

Enseignant:	Sylvain Hallé
Bureau:	P4-5240
Courriel:	shalle@uqac.ca
Page web du cours:	http://moodle.uqac.ca
Disponibilité:	Sur rendez-vous

Contenu général

On rapporte que chez Microsoft, près d'un programmeur sur deux est chargé de tester les produits, et non de les développer. Cette donnée surprenante est une conséquence du fait que la croissance des capacités des ordinateurs s'est accompagnée d'une croissance encore plus grande de la complexité de leurs programmes —et de la difficulté à prévoir toutes leurs exécutions possibles et à retracer toutes les erreurs imaginables. Le développement d'applications informatiques s'appuie donc largement sur la notion de *test* : on imagine des séquences d'actions ou d'entrées à soumettre au programme visant à révéler des potentielles erreurs de programmation. Cette technique, qui ne fournit aucune garantie formelle de qualité, commence à montrer ses limites à mesure que les systèmes informatiques gagnent en complexité.

Ce cours se veut une introduction aux concepts fondamentaux de la spécification, du test et de la vérification automatiques de systèmes informatiques. On y verra comment spécifier le fonctionnement attendu d'un système au moyen de langages formels comme les automates, la logique propositionnelle et la logique temporelle. On étudiera ensuite les méthodes permettant de vérifier, par différents moyens, qu'un système donné satisfait bien la spécification fournie. Finalement, les concepts théoriques seront accompagnés d'une variété d'exercices pratiques, qui seront l'occasion d'utiliser plusieurs logiciels de test et de vérification automatique.

Objectifs du cours

Au terme de ce cours, l'étudiant aura acquis les compétences suivantes :

- Modéliser des systèmes au moyen de différents langages de spécification
- Comprendre et appliquer des techniques de génération et d'automatisation des tests
- Identifier les limites de l'approche par tests dans le développement de logiciels
- Comprendre et appliquer des algorithmes de vérification automatique (satisfaisabilité, runtime monitoring, model checking)
- Utiliser une variété d'outils automatiques pour vérifier qu'un système satisfait une spécification

Sujets abordés

Un aperçu du contenu de chaque leçon est donné ci-dessous. L'ordre de présentation peut être modifié selon les besoins.

1. Introduction Bugs célèbres. Coût des bugs ; relation entre bugs et taille du programme. Bug tracker ; issue tracker sur GitHub. Classification des bugs. Aperçu des travaux du LIF. *(2 mai)*

2. « Design by Contract » Première notion de spécification : typage et annotations. Types, JML, programmation par contrat. Langages de contraintes : logique propositionnelle et solveurs SAT. *(3 mai)*

3. Spécifications en tant que contraintes Logique du premier ordre, Object Constraint Language, le solveur Mace. *(16 mai)*

4. Spécifications dynamiques Spécification de séquences d'opérations. UML statecharts, automates, expressions régulières, logique temporelle LTL. *(17 mai)*

5. Introduction aux tests Terminologie de base. Principes de testing. Catégories de tests. Métriques et notion de couverture. *(23 mai)*

6. Génération de tests, partie 1 Partition en classes d'équivalence. Tests combinatoires. Réduction à un problème de coloriage. *(24 mai)*

7. Génération de tests, partie 2 Tests concoliques. Tests basés sur les automates. Tests de mutation. La librairie SealTest. *(6 juin)*

8. Automatisation des tests Intégration continue ; outils d'automatisation ; test-driven development (TDD). *(7 juin)*

9. Runtime verification Instrumenter un système avec la programmation orientée-aspects. Le moniteur JavaMOP. Au-delà du monitoring : le Complex Event Processing. Préssentation du logiciel BeepBeep 3. *(13 juin)*

10. Model checking Structure de Kripke. Modélisation. Bounded Model Checking de LTL par réduction au problème SAT. Le logiciel NuSMV. *(14 juin)*

À noter qu'il n'y aura pas de cours les 30 et 31 mai (enseignant à l'étranger). La séance du 20 juin sera consacrée à la réalisation des exercices pratiques et la réponse aux questions des étudiants.

Évaluation

L'évaluation consistera en les éléments suivants :

- Six exercices pratiques, à effectuer en équipes d'au plus deux personnes. Les exercices seront donnés au fil des séances, et du temps en classe sera réservé pour leur réalisation. Cependant, la plupart d'entre eux nécessitent du travail supplémentaire en dehors des périodes de cours pour être complétés. Les trois premiers exercices seront à remettre au plus tard le **25 mai**, et les trois autres au plus tard le **25 juin**. Chaque exercice comptera entre 5 et 15 points, pour un total de **60 points**.
- Un travail s'échelonnant sur l'ensemble de la session, portant sur l'étude d'un outil existant ou d'un article scientifique relatif au contenu du cours. L'évaluation du travail sera effectuée par une présentation orale d'au moins une demi-heure, à présenter devant l'enseignant, sur rendez-vous, la semaine du **18 juin**. Ce travail comptera pour **20 points**.
- Un examen final à la dernière séance, le 21 juin. Cet examen compte pour **20 points**.

La note finale, sur 100, est la somme de toutes ces évaluations. La note de passage est fixée à **50%** ou C. (L'attribution des notes C-, D+ et D s'applique uniquement aux cours de premier cycle.)

Date limite d'abandon sans mention d'échec

20% de l'évaluation aura été transmise à l'étudiant avant la date limite d'abandon sans mention d'échec, soit le jeudi 14 juin 2018.

Utilisation des TIC

L'usage des TIC est permis en salle de cours pourvu qu'il ne dérange pas l'enseignant et l'apprentissage des étudiants.

Qualité du français écrit

Tout travail remis doit être conforme aux exigences de la politique institutionnelle en matière de maitrise du français écrit. Comme il en est fait mention dans le Manuel de gestion (3.1.1-012), [1] tout travail dont la qualité du français serait jugée non conforme par l'enseignant pourra être pénalisé jusqu'à concurrence de 10% du résultat maximal prévu.

Évaluation de la qualité de l'enseignement

Ce cours sera évalué en fonction de la Procédure relative à l'évaluation des activités aux programmes d'études de cycles supérieurs (Manuel de gestion, 3.1.2-008).

Formule pédagogique

Le cours sera dispensé en classe par un professeur. Certaines séances se tiendront en laboratoire et comporteront une période d'exercices et de travaux pratiques.

Situation du cours dans le programme

Le cours est optionnel pour les étudiants inscrits aux programmes de cycles supérieurs. Aucun cours ne lui est préalable.

Références

Aucun livre ou recueil de notes n'est obligatoire pour ce cours. Les diapositives utilisées durant les séances seront rendues disponibles en format électronique sur le site (Moodle) du cours et seront accompagnées de lectures complémentaires.

Les références suivantes sont suggérées pour des compléments à la matière vue en classe. Des articles de journaux et de conférences scientifiques seront également au programme ; les références

1. http://www.uqac.ca/direction_services/secretariat_general/manuel/index.pdf

seront publiées sur le site du cours. Les monographies suivantes constituent de bons points de départ pour la matière vue dans ce cours.

- P. Ammann, J. Offutt. (2016). *Introduction to Software Testing*, Cambridge University Press. ISBN 978-1107172012.
- R. Black, E. van Veenendaal, D. Graham. (2012). *Foundations of Software Testing — ISTQB Certification*. Cengage Learning EMEA. ISBN 978-1408044056.
- D.R. Kuhn, R.N. Kacker, Y. Lei. (2013). *Introduction to Combinatorial Testing*. Chapman and Hall/CRC. ISBN 978-1466552296.
- M. Huth, M. Ryan. (2004). *Logic in Computer Science : Modelling and Reasoning about Systems*. Cambridge University Press, ISBN 978-0521543101.
- R. Miles. (2004). *AspectJ Cookbook*. O'Reilly, 978-0596006549.
- A. Page, K. Johnston, Bj Rollinson. (2008). *How We Test Software at Microsoft*. Microsoft Press, ISBN 978-0735624252.
- E.M. Clarke, O. Grumberg, D.A. Peled. (2000). *Model Checking*. MIT Press, ISBN 0-262-03270-8.

On donne également les références vers certains des outils mentionnées ci-dessus. Ils sont tous disponibles gratuitement.

- MiniSAT : `http://minisat.se`
- Prover9/Mace : `http://www.cs.unm.edu/~mccune/mace4`
- Alloy : `http://www.alloy.mit.edu`
- NuSMV : `http://nusmv.fbk.eu`
- Checkers : `http://types.cs.washington.edu/jsr308/current/`
- BeepBeep : `http://liflab.github.io/beepbeep-3`
- Maude : `http://maude.cs.uiuc.edu/`
- Java-MOP : `http://fsl.cs.uiuc.edu/index.php/MOP`
- Java PathFinder : `http://javapathfinder.sourceforge.net/`

Site du cours

Toutes les ressources en ligne relatives au cours se trouvent sur le site Moodle de l'université, à l'adresse `http://moodle.uqac.ca`.

Identifiant : _____

Mot de passe : _____

Si un compte Moodle n'a pas déjà été créé, il est de la responsabilité de l'étudiant de se faire ajouter au groupe « 8INF958 (Sylvain Hallé) », groupe 01, en contactant le support informatique à l'adresse `supportsti@uqac.ca` ou par téléphone au 418-545-5011, poste 6000.