

<i>Titre:</i>	<i>Title:</i> SOFTWARE ANALYSIS AND VERIFICATION
<i>Enseignant:</i> Viktor Kuncak	

OBJECTIFS

GOALS

The class will introduce foundations, algorithms, and tools for automated analysis and verification of complex properties of software systems.

CONTENU

CONTENTS

Motivation:

Tools for automated analysis and verification of software can improve reliability of software that we use every day. The underlying techniques are also used for compiler optimizations and program understanding. In recent years, new algorithms and combinations of existing techniques have made such tools more effective than in the past. In the near future we can expect such tools to be used more and more as a part of the everyday software development practice. This course will give an overview of the basic techniques, as well as the recent advances that made this progress possible.

The topics covered include:

- A review of program semantics and logic
- Verification condition generation
- Introduction to theorem proving and decision procedures
- Assume/guarantee reasoning and data abstraction
- Syntactic loop invariant inference
- Abstract interpretation and data flow analysis
- Predicate abstraction; shape analysis
- Interprocedural analysis
- Analysis of higher-order, object-oriented, and concurrent programs
- Dynamic analysis; bug finding; loop unrolling

FORME DE L'ENSEIGNEMENT:	Lectures (2+2), Exercises (2). The course will include lectures, exercises, paper discussions, mini project presentations, and possibly an invited lecture.	FORME DU CONTRÔLE:	Grading will be based on one mini project, weekly homeworks, class participation and taking lecture notes. Students will participate in homework grading.
BIBLIOGRAPHIE:	<p>Lectures: Powerpoint slides will be available for each lecture. Lecture notes will produced jointly by students and the instructor after the class.</p> <p>Bibliography will include:</p> <ul style="list-style-type: none"> ● A selection of recent research papers in program analysis and verification. ● Steven S. Muchnick and Neil D. Jones: Program Flow Analysis: Theory and Applications. Prentice Hall, 1981. ● Flemming Nielson, Hanne Riis Nielson, and Chris Hankin: Principles of Program Analysis. Springer, 1999. <p>Software: Mini projects may include the use of infrastructure such as Jahob as well as others.</p>		
URL DU COURS :	Google 'Viktor Kuncak' for more information.		
LIAISON AVEC D'AUTRES COURS:			
<i>Préalable requis:</i>	Familiarity with basic discrete mathematics and logic, Theory of Computation III, and Compiler Construction is helpful but not required. If you are not familiar with these topics, please see the instructor.		
<i>Préparation pour:</i>	Research and development in the area of program analysis, verification, software reliability, and compilers.		