

CS 246 Advanced Verification Techniques in Software Engineering

Fall 2008

Instructor	Gianfranco Ciardo	email: ciardo@cs.ucr.edu http://www.cs.ucr.edu/~ciardo/
Prerequisites	CS 111 or MATH 111, CS 141, and CS 150 (or equivalent courses)	
Units	4	
Office hours	Tue & Thu 3:40pm – 5:00pm	Room: EBUII 442
Class meetings	Tue & Thu, 2:10pm – 3:30pm	Room: EBUII 139
Final examination	Fri, Dec 12, 8:00am – 11:00am	

Catalog description A study of advanced techniques to specify and examine the correctness of complex systems and software. Focuses on concurrent and distributed behavior, formal description languages, temporal logics, model checking and symbolic model checking, partial order reduction, and the use of verification tools.

Contents A fundamental activity in Systems and Software Engineering centers around advanced techniques and tools to specify and verify the correctness of the artifacts that are under study or are being proposed or built. Many such techniques have now matured to the point that their industrial use is becoming cost-effective. The course will allow you to achieve a good level of familiarity with these techniques and their use, and will provide you with the required background to perform research in this area. Topics, roughly in the order in which they are covered, include:

- The need for hardware and software verification
- Automata: definitions, synchronization mechanisms, reachability
- A high-level language to specify automata: Petri nets
- A software tool: SMART
- Temporal logics: CTL*, CTL, LTL
- Model checking algorithms for CTL and LTL
- More software tools: NuSMV and Spin
- Decision diagrams: BDDs, MDDs, MTMDDs, EVMDDs
- Symbolic CTL model checking
- Reachability properties
- Safety properties
- Liveness properties
- Notions of fairness
- Advanced topics (their coverage depends on students' interests and on available time):
 - Abstraction
 - Partial order reduction
 - Model checking timed models
 - Model checking probabilistic models
 - Bounded model checking

Textbook The official textbook for this course is

- B. Berard et al. *Systems and Software Verification*. 2001, Springer-Verlag.

In addition, I will occasionally make available class notes and foils. For further consultation, the following books provide excellent information covering most of the class material, and much more:

- Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. 2008, MIT Press.
- Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. 1999, MIT Press.
- Michael R. A. Huth and Mark D. Ryan. *Logic in Computer Science, Modelling and reasoning about systems*. 2001, Cambridge University Press.
- Doron A. Peled. *Software Reliability Methods*. 2001, Springer-Verlag.

Coursework The following coursework will be used to assess your final grade:

Homeworks (around 6):	30% of the grade.
In-class presentation of assigned reading material:	10% of the grade.
Project paper:	30% of the grade.
In-class final exam:	30% of the grade.

In-class presentation of assigned reading material: each student will be required to make a 10-minute presentation of a research paper from the literature, using a laptop and a projector (provided by the instructor). Since 10 minutes is a very short time, it is essential to focus on the main idea (there is no need to cover much background, as we should all be familiar with it) and to practice the presentation multiple times to be sure it flows well in the allotted time.

Project paper: this is normally the result of individual work, on a topic previously agreed upon with the instructor. However, under some circumstances, a multiple-student project might be approved by the instructor, provided each student participating in the project has a clearly defined portion for which he or she is responsible. A project proposal is required at least one month before the end of classes. Various types of projects can be proposed, for example: (1) conducting a case study of a substantial system using one or more verification tools, (2) comparing verification tools on a realistic set of benchmarks, (3) implementing a new algorithm in a verification tool, or (4) developing the theory for a new analysis algorithm. A mixture of the above, e.g., (1) and (2) or (3) and (4) might also be quite appropriate. If you foresee having to turn in a homework after its due date for justifiable reasons (such as an illness), you *must* let me know by email or in person as soon as you can, and certainly before the deadline.

I *require* the use of LaTeX to typeset your homeworks and the project. If you are not familiar with LaTeX, you can find examples to get you started on my homepage, or on the web. Don't wait until your first homework is due!

Guidelines Unless otherwise stated, all coursework is to be performed by you alone. If you use external information for an assignment (for example, on the web or in a book), you can incorporate it in your homework as long as you include a proper reference. In addition, verbatim citations should be clearly indicated as such. Finally, any programming work you turn in for this course should be your own. Anything else is considered cheating and will result in a zero score for the entire homework assignment, or worse.

You are required to attend, follow, and actively participate in all lectures. All lectures and exams start at the stated time. Avoid being late coming to class, as this is very disruptive.

You are requested to refrain from using electronic devices (laptops, cell phones, music players, etc.) during lectures and exams. If, for some compelling reason, you need to be on call, put your cell phone in silent mode, and excuse yourself from the class if you need to take a call. Of course, when you come see me in my office, turn your cell phone off.

Grading policy Unless you successfully petition for a Satisfactory/No Credit (S/NC) grade, you will receive a letter grade, which will be determined as follows. Assuming your overall numerical grade is $x\%$, your letter grade is:

$-\infty < x < 57$: F	$57 \leq x < 60$: D-	$60 \leq x < 63$: D	$63 \leq x < 67$: D+
		$67 \leq x < 70$: C-	$70 \leq x < 73$: C	$73 \leq x < 77$: C+
		$77 \leq x < 80$: B-	$80 \leq x < 83$: B	$83 \leq x < 87$: B+
		$87 \leq x < 90$: A-	$90 \leq x < 93$: A	$93 \leq x < +\infty$: A+