

CS556/CS656 Formal Methods Syllabi

CS556 Introduction to Formal Methods, Models & Languages

Applications of logic and mathematics in documenting problems, requirements, specifications, designs, and software. Formal modeling languages. Diagrammatic, algebraic, and tabular models. Model checking. Students prepare, check, and present models using technique in the literature. **Prerequisites: CSCI320 and CSCI330.**

CS656 Formal Methods, Models & Languages

Applications of logic and mathematics to documenting problems, requirements, specifications, designs, and software. Formal modeling languages. Diagrammatic, algebraic, and tabular models. Model checking. Students prepare, check, and present models using techniques in the literature for a research paper. **Prerequisites: Classified status**

Notes

- This course was taught for the first time in the Winter of 2002.
- Three hours lecture and two hours activity laboratory each week.
- Students may not receive credit for both CSCI556 and CSCI656.

Syllabi and Schedule

See <http://www.csci.csusb.edu/dick/syllabus.html> for the general rules and grading for all my classes. Syllabi and other updated information will be on the WWW at <http://www.csci.csusb.edu/dick/cs556/> and <http://www.csci.csusb.edu/dick/cs656/>

Text Book

Logic in Computer Science: Modeling and Reasoning about Systems, Michael Huth and Mark Ryan, Cambridge University Press, 2000. See <http://www.cis.ksu.edu/~huth/lics/>

Work

You will need to read the assigned reading before the start of each class. The schedule lists the reading assignments. You will learn more if you try out the exercises. The reading drives the class work. Typically exercises from the reading will be assigned at the end of each class to be completed before the start of the next class. Each class is

followed by a laboratory.

Students taking CSCI 656 will need to spend time in the library. They must choose a published piece of work to review and will and present the review at the end of the quarter.

Grading Scheme

Total 500 points= 100%.

- Home work ≤ 40 points max. There will be at least 14 exercises assigned. Each is worth a max of 3 points. The assigned work is due at the start of the next class -- or it is worth nothing.
- Lab work 200 points. There will be 20 labs each graded A=10, B=9, C=8, D=7, ... Missing=0. Graded at the end of lab period. Students will show what they discover in their lab work to others in the lab to earn credit.
- Midterm on logic: 60 points. See Schedule for date.
- Comprehensive Final 200 points. Answer 4 out of given 5 questions in two hours. Each question is worth 50 points.

Students of CSCI656 are also required to prepare and present a paper. This is graded A/B/C/D/Fail. For CSCI656 students, the grade for the whole class will be the maximum of the grade on the paper and on the rest of the class: Failing the presentation automatically fails the course.

What are Formal Methods good for?

Formal methods are good when software must work. The book shows the Ariane rocket exploding on the cover because of a software error. It was about to hit a village. When errors can kill people, you should use formal methods plus rigorous quality control to eliminate them. Another example is software running inside a person's body. But, even if nobody can be hurt by errors, formal methods can be valuable. For example, suppose the software is running in a probe a long way from Earth you need to work harder on correctness because patches are expensive over a million-mile distance. A formal method: "model checking" eliminated bugs in part of the software for one NASA probe that were identical to bugs that shut down the probe in an unchecked piece of code.

Closer to home, replacing a chip in a million microwave ovens will be expensive once the ovens have been sold. Rather than recalling and replacing them, Phillips prefers to get the code right.

Formal methods are important to secure a system from terrorism, fraud, robbery, and identity theft. They are the only known way of avoiding subtle loopholes in protocols. The IEEE 802.11 (Wi-Fi) standard for example has been attacked because of the loopholes discovered after it was deployed. Formal methods can help protect information and money. When people can be hurt, or their money, career, or reputation destroyed, formal methods can help you sleep at night.

In some projects there is a trade off between budget, development time, and quality. Mass-marketed software companies employ armies of lawyers and customer service representatives to defend itself from the effects of errors on its customers rather than reducing the number of errors. Typically there is no real warranty and early customers continue testing the product for the producer. Here formal methods are not seen as valuable.

For most projects you need to be agile -- being prepared to adjust to what happens. This is how I use formal methods. I drop into a suitable method to solve a particular problem in the project and then leave the method. I use them to document a specific situation and to record, share and/or test a particular thought. This implies that I have a collection of "ready-to-go" methods. The key is

becoming familiar and skilled with many simple mathematical techniques. The skill and familiarity come from practice.

Notice that many industrial pundits and consultants confuse "formal method" with "formal process". All formal methods depend on mathematics. A **formal process** merely has a written set of rules defining how software is developed in an organization. It may or may not be mathematical. Similarly, mathematics can be applied creatively or as part of a rigid process.

What are Formal Methods?

A **Formal Method** is *any method that starts by modeling a situation using formulae and then uses predefined and precise rules to manipulate them*. For example you can express a problem and solve it using algebra. Or you can express a possible solution as a logic, state requirements as formulae in the logic, and then check that the solution meets the requirements. You can construct a model of the ideas in the users' minds and see what they mean if implemented.

In a formal method, then, the key step is translating informal statements into symbolic statements in a formal language. This method is as old as Rene Descartes. Modern tools can often do a lot of heavy lifting once the situation is "formalized". However, you need to know more than just the languages and tools: You need to know the limits of the languages and tools.

Formal methods can be used at any point in a project: they can be used to clarify, document, and analyze requirements, to check designs and prove algorithms. They can be used to express ideas and check them.

Semiformal methods include using the Unified Modeling Language. A semiformal method has a rigorous notation but may omit the ability to reason correctly.

Lite Formal Methods are tuned for particular kinds of problems, minimize the notation, and provide powerful tools.

Why Mathematics?

(1) Mathematical notation acts as shorthand. It lets you say a lot in very few symbols. Recording your thinking about complex systems and software in natural language takes a lot of space. As a simple

Example, compare the number 2002 with the words needed to describe the same number. (2)

Mathematical notation makes you clarify what is going on. Most requirements and specifications are ambiguous when written in a natural language.

Reexpressing them in a formal language forces you to be precise. (3) Mathematical manipulation helps you uncover what hidden values and properties: is this computation safe? Do the requirements contradict themselves? Does the design fit the specification? Does the code fit the design?

What Mathematics?

In computing, the mathematics is discrete math. So formal and symbolic logic and Boolean algebra are fundamental. Some methods use sets and directed graphs.

The mathematical formulae can be in conventional written format, or use tables, data structures, or diagrams. They can be written/drawn by hand, using an equation editor, or using a specialized ASCII code, etc. Unlike pure mathematics using natural language to explain what you are doing is a useful procedure.

Where do Formal Methods come from?

They are typically invented in universities and research laboratories. The typical formal method goes through these stages: invented by a researcher, published in journals, criticized in journals, magazines, and online, improved in response to criticism, tested outside academe/research labs, hyped in magazines as the best thing ever, put into text books, proven in industrial settings, and quietly adopted by industry. This life cycle can be aborted at any point: a good idea may never get published or a text book method may never get into mainstream practice.

Which Methods are in CS556/656?

We will cover basic logic focusing on proofs using Natural Deduction: The Propositional Calculus (PC) plus a fundamental representation technique for encoding logical expressions in a program: OBDDs. Then we move onto the lower predicate calculus with equality (LPC) and a little Prolog (PROgramming LOGic). We will look briefly at using the ideas of sets and relations to model complex domains using the UML. We will study model checking with the Computational Tree Logic (CTL) and the SMV tool. We will look at methods for finding bugs in code and proving algorithms correct. There will be presentations about a selection of other formal methods and notations

such as: Z, VDM, automata, petri nets, CSP, the pi-calculus, etc.

Exercise 1

Do you know any ways to express the above grading rules for this class other than natural language and source code?

Exercise 2

Do you know any ways to express the rules that govern the degree you are taking other than hand books and diagrams?

Exercise 3

Do you know any really complicated sets of rules in some part of real life or the computer world?

Exercise 4

Which of the following products are most suitable for the use of formal methods: A word processor, controlling a railroad crossing and signals, The chip that controls the cruise control in an auto, a payroll program, the Palm OS, The Pentagon Phone system, a missile controller?

CSci656 Paper/Presentation

Students of CSCI656 will prepare and present a review describing a published piece of research on formal methods found in journals and/or books in the library.

You should start working in the library on this paper soon. ~~Start by looking for the Computer Reviews. Read some reviews.~~ This gives you an idea of the style and purpose. You should then search for topics in Formal Methods that interest you. You should show a list of 2 or 3 possible publications to me when possible.

A one page progress report should be submitted for review at the start of class 12. This should be the draft header sheet for your review.

The final version is handed to me at the start of your presentation.

The presentation will be in the **last two classes and labs** of the quarter. Presentation will take 15 to 20 minutes plus questions. Other faculty and students may attend these. The audience may ask questions that demand more details.

Let me know **before class 16** what equipment is needed. You would be wise to show me your draft review and visuals before Class 17.

Review Contents/Layout

Header Sheet

-- identify the work you are reviewing

Answer at least these questions:

What is it called? -- title

Who wrote it? -- authors

Where & when was it published?

Journal, volume, issue, date, pages.

Is it on the web, if so -- where?

Full URL.

Is it in the library?

If so what is the Call Number.

Is it a: book, paper, thesis, web site, tool, ...?

The Publication's Content

Subject area: What is it about?

Purpose: What is the message?

What are they trying to prove/advertise?

How do they do it?

claim, demo, experiment, survey, ...

How does it link to other publications?

How many references does it have?

Your Comments

What do you think about the paper?
readable? valuable? memorable? convincing?
How long? Too long? Too short?
Who else should read it?
Best and worst features?
How does it fit/compare to 656/556?

References/Citations

List any other works you use as evidence.
Give authors, titles, and where to find it.

Attachments:

For non-book literature, attach a printout /copy.
If it is a library book then
return it to the library. (I need to look at it:-)

Key Points

1. Quotations

Avoid copying and/or quoting text. Summarize instead!

If you need to show precisely what the authors wrote, you must do one of the following:

(1) Place short quotes in double quotation marks and add ref number and page numbers:

"..."[1 page 165]

(2) Indent larger quotations
and add a citation

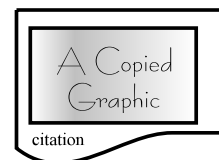
".....

.....

...."[5 pages165-166]

If you copy any pictures or graphics in your written review or in your presentation you must clearly state where they come from:

Failing to follow this set of rules will cause the paper and the student to get an F.



2 Length

Your description of the content and your comments should be between 200 and 500 words. **Writing too little or too much can reduce your grade.** You have less than 200 minutes divided by the number of graduate students in CSci656. **Making the presentation last too long will also decrease your grade.**

3 Grading

The grade will be based on holistic grading of the presentation + written paper.

Good papers will be written in good academic english. They will be clear and easy to read. They will accurately describe at least one paper published in a research journal or a library book. They will clearly indicate any quotations and give citations and references for them. They will refer to other publications and to WWW sites. They will connect the reviewed paper to topics in this class.

The report should be submitted for final grading in hard copy format at the time of presentations. The page layout, fonts, section numbering, reference lists, etc. of the final paper should be close to a thesis or project report. Drafts can be submitted as WWW pages, ASCII text files via EMail, or on paper.

The presentation should be simpler than the paper. It should present the most important points in the paper.