EECS Graduate Program: Transfer Course Request

For Completion by Student:

Name: ____ ID #: ____________________

Requested Course:

Institution:	
Course Subject/Number:	
Course Title:	
Number of Credits:	
Grade:	
Date of Completion:	

WSU Equivalency:

Course Subject/Number:	
Course Title:	
Number of Credits:	
Current Instructor:	

For Completion by Advisor:

Rationale for Transfer:

Suggested Faculty Reviewer: ____

For Completion by GSC:

Assigned Faculty Member for Review: ____

For Completion by Faculty Reviewer:

Please select one:

□ I approve the transfer of this course as requested.
□ I approve the transfer of this course for a decreased number of credits: Credits.
□ I approve the transfer of this course as a different graduate-level course than requested.I
□ deny the request to transfer this course.

Comments:

Reviewer Signature: ____

ALR Spring 2021

Welcome to the home page for Automated Logical Reasoning!

Logistical Information:

Instructor:	Işıl Dillig
Time:	Tuesday, Thursday 2-3 pm
Place:	Zoom
Instructor e-mail:	isil@cs.utexas.edu
Instructor office hours:	Thursday 3-4 pm
TA:	Shankara Pailoor
TA e-mail:	spailoor@cs.utexas.edu
TA office hours:	Monday 4-5 pm
Reference books (optional):	The Calculus of Computation by Aaron Bradley and Zohar Manna; Decision Procedures: An Algorithmic Point of View by Daniel Kroening and Ofer Strichman
Course Webpage:	http://www.cs.utexas.edu/~isil/cs389L/

Course Description:

Automated logical reasoning has enabled substantial progress in many fields of computer science, including software and hardware verification, theorem proving, program analysis, and artifical intelligence. In this course, we will study widely-used logical theories and decision procedures for answering whether formulas in these theories are satisfiable. In particular, we will consider automated reasoning techniques for propositional logic, first-order logic, linear arithmetic over reals and integers, theory of uninterpreted functions, and combinations of these theories. We will also look at applications of logic in program analysis and verification.

Requirements:

- Regular class attendance is required. Please email instructor if you will not be able to attend class on a particular day.
- This course will have a combination of programming assignments and problem sets. Collaboration is **not** allowed on either.

- This offering of the course will not have exams.

Announcements:

- Our first class will meet on Tuesday, Jan 19.
- All problem sets and programming assignments will be posted on Piazza .

Syllabus:

In the Reading section of the syllabus below, COC refers to the Bradley & Manna Calculus of Computation book, while DP refers to the "Decision Procedures: An Algorithmic Point of Book" by Kroening & Strichman.

Date	Lecture topics	Notes	Reference
01/19	Introduction and basics	Lecture 1	COC 1.1-1.5
01/21	Normal forms and DPLL	Lecture 2	COC 1.6-1.7
01/26	CDCL-based SAT Solvers	Lecture 3	DP 2.2 CDCL SAT solvers
01/28	Free time for programming assignment	N/A	N/A
02/02	Practical applications of boolean satisfiability	Lecture 4	N/A
02/04	Binary decision diagrams	Lecture 5	Notes on BDDs DP 2.3
02/09	Semantics of First Order Logic	Lecture 6	COC 2.1-2.4, COC 2.7
02/11	Proof rules and properties of FOL	Lecture 7	COC 2.6
02/16	Unification	Lecture 8	
02/18	First-order theorem proving	Lecture 9	
02/23	Overview of First-Order Theories	Lecture 10	COC Chapter 3
02/25	Theory of Equality	Lecture 11	COC Chapter 9
03/02	Free time for programming assignment	N/A	N/A
03/04	Linear Arithmetic over Rationals	Lecture 12	CLRS Chapter 29
03/09	Linear Arithmetic over Integers	Lecture 13	
03/11	Nelson-Oppen	Lecture 14	COC Chapter 10
03/16	Spring break		
03/18	Spring break		
03/23	DPLL(T) Framework	Lecture 15	
03/25	Hoare Logic	Lecture 16	
03/30	Verification conditions	Lecture 17	
04/01	VCs with functions and pointers	Lecture 18	
04/06	Intro to Dafny	N/A	
04/08	Time for programming assignment		
04/13	Abstract interpretation	Lecture 19	

04/15	Guess-and-check methods	Lecture 20	Houdini Abduction
04/20	Predicate abstraction; CEGAR	Lecture 21	
04/22	Proving (non-)termination	Ledture 22	
04/27	Reasoning about concurrency	Lecture 23	
04/29	Program Synthesis I	Lecture 24	
05/04	Program Synthesis II	Lecture 25	
05/06	Wrap-up		