1 Catalog Description
This course focuses on the formal specification and verification of reactive systems, most notably concurrent and distributed systems. Topics relevant to these systems, such as nondeterminism, safety and liveness properties, asynchronous communication or compositional reasoning, are discussed. We rely on a notation (TLA+, the Temporal Logic of Actions) and a support tool (TLC, the TLA+ Model Checker).

2 Attributes
- This course is one of the CS electives designated as theory.
- This course can be combined with a CS696W module to satisfy a Writing Intensive requirement.
- This course can be taken as CS745H by honors students.

3 Outcomes
- methodologies of software development: formal specification, program correctness, safety and liveness, concurrent programming issues (nondeterminism, race conditions, etc.).
- fundamental computer science algorithms: concurrent algorithms, synchronization.
- principles and techniques of calculus, probability and statistics, and mathematical proof techniques: symbolic logic and formal proof, induction, variants and invariants, temporal logic.
- think abstractly and reason logically about computer science problems: modeling algorithms, state machines and transition systems.
- principles and techniques of a range of advanced topics: model-checking as a verification and debugging tool.

4 Topics
- Formal specification and verification:
  - implementation vs specification
  - functional correctness, typical properties (precondition, postcondition, invariants, termination)
- Reactive systems as state transition systems:
  - reactive systems vs transformational systems
  - system states, initial states, state transitions, behaviors
  - linear-time temporal logic (LTL), safety and liveness
  - formal definition of correctness
• Modeling of reactive systems:
  – state predicates
  – state modeling using sets and functions
  – sequentiality, parallelism, nondeterminism, atomicity
  – weak and strong fairness

• Reactive systems in TLA⁺:
  – state transitions as binary predicates
  – stuttering and termination
  – “next-state” predicates as disjunctive formulas

• TLA⁺ syntax and semantics:
  – sets and functions in TLA⁺
  – tuples, sequences and records as functions
  – quantifiers and set builder notation
  – finite sets and cardinality
  – ∧ and ∨ in bulleted lists form
  – IF-THEN-ELSE, EXCEPT, UNION and CHOOSE operators

• System properties in TLA⁺:
  – □, ◊ and ~ properties
  – state-based and action-based properties
  – correctness as logical implication

• Model checking with TLC:
  – Explicit-state model checking
  – using TLC, configuration files
  – limitations of TLC

• Proving properties in TLA⁺:
  – inductive invariants, INV1 rule
  – proving action properties
  – well-founded sets (variants), WF1 rule, Lattice Rule

5 Evaluation
Seven homework assignments (20%), one project (30%) and two exams (50%).

6 Textbooks
Required:


Additional (for a discrete math refresher):