Date, Time, and Place:  Wednesday, February 25, during your regular lecture time
Duration:  50 minutes
Type:  closed book

This exam covers topics discussed in the following chapters and sections of your textbook: Chapter 5 (sections 4-6) and Chapter 6 (sections 1-6).

Use lecture handouts for guidance. Here is the list of handouts in the chronological order:
- Nondeterministic Finite Automata
- Nondeterministic Finite Automata with \( \lambda \)-transitions
- Removing nondeterminism
- Finite Automata and Regular Sets
- Regular Grammars and Finite Automata
- Closure Properties of Regular Languages
- The Pumping Lemma For Regular Languages + accompanying handouts for proofs

In midterm 2 you can expect the following type questions:

Part1. Exercises on learned topics (similar to homework exercises):
- Construct an NFA (or NFA-\( \lambda \)) to accept the given language.
- Construct an equivalent DFA for the given NFA (or NFA-\( \lambda \)) using the Algorithm 5.6.3.
- Construct a regular expression for the given finite automaton using the Algorithm 6.2.2. You’ll need to draw the graph after each deleted node (algorithm’s each loop-iteration).
- Construct a finite automaton for the given regular grammar using the algorithm presented in Theorem 6.3.1.
- Construct a regular grammar for the given finite automaton using the algorithm presented in Theorem 6.3.2.
- Prove that the given language is regular. To prove a language is regular you need to do one of the following:
  (i) construct a regular expression to specify that language,
  (ii) construct a regular grammar to generate that language,
  (iii) construct a finite automaton to accept that language,
  (iv) show that it can be obtained from regular languages using closure properties.
  Note: you will need to prove the regularity of each participating language – you can not just say “we know \( L_1 \) is regular” – you need to prove it.
- Using the Pumping Lemma, prove that the given language is not regular.
  - There will be short-answer questions asking you to check all strings in a given list (or give an example of a string) that would be appropriate to take as the \( z \) string to correctly carry out the proof.
  - You may also be asked to provide the whole proof.
Part 2. Short answer questions on learned concepts, definitions, theorems, results (similar to quiz questions posted on the course site: www.csc.calpoly.edu/~hghariby/CSC445

You are particularly expected to know the following definitions, concepts, theorems, conclusions:

Chapter 5: Nondeterministic finite automata (NFA and NFA-λ), λ-transitions, acceptance of a string, language of a nondeterministic automaton, machine configuration, tracing of a computation, λ-closure, δ and τ functions, Lemma 5.5.2, Theorem 5.5.3, Theorem 5.6.4.

Chapter 6: Expression graphs, minimal/simplest expression graphs, regular expressions specifying the language of each minimal/simplest graph. Relationships between Regular Sets, Regular Grammars, and Finite Automata. Theorem 6.2.3 (Kleene), Theorems 6.4.1-6.4.3 (closure properties), Theorem 6.6.3 (Pumping Lemma), Theorem 6.6.4, Corollary 6.6.5, Corollary 6.6.6.