Course Overview

- What does the term “*compute”* mean?
- What does the term *“computer”* mean?
- Is there anything that a computer cannot do?
- These are the kinds of questions that will be addressed in this course.
- The answers to these questions are very precise and, in particular, not simply a matter of opinion.

General Topics

- Abstract machine models of computation (e.g., DFA, NFA, LBA, TM).
- Grammars (e.g., regular grammars, context-free grammars, context sensitive grammars, unrestricted grammars).
- Classes of formal languages (e.g., regular, context-free, context-sensitive, recursive, recursively enumerable).
- Relative computational power of the abstract machine models (not just speed, but ultimate computability).
- Relationships between the languages, grammars and machines.

Practical Application

- A variety of properties concerning the models, grammars, and languages will be proven.

- The existence or non-existence of algorithms for processing languages and language processors will be proven.

- These algorithms form the basis of tools for processing languages, e.g., parsers, compilers, assemblers, etc.

- Other algorithms we will study form the basis of tools that automatically construct language processors, e.g., yacc, lex, etc.
  - Note that our perspective will be similar to, yet different from a compiler class.

- Additionally, some things will be proven to be non-computable, e.g., a compiler that detects infinite loops.