CS415 Compilers

Final Review
Final Exam Coverage

- Classes
- Lecture Notes
- Homework Sets
- Reading in EaC
  → Indicated at the end of every lecture’s slides

A focus on materials covered in the latter half of the semester, starting from LR parsing.
Scanning and Parsing

• Regular Expression
  → NFA and DFA definitions and conversions

• LL(k) parsing
  → How to construct first set, first+ set?
  → How to build a LL(1) parse table?
  → Is a grammar LL(1)?

• LR(k) parsing
  → LR(1) item, closure, state
  → How to build LR(1) canonical collection of sets of LR(1) items?
  → How to build a LR(1) parse table?
  → Is a grammar LR(1)?
Semantic Analysis

- **Attribute Grammar**
  - Definitions and different types of attribute grammar
  - Rules of evaluating attribute values
- **Syntax Directed Translation (SDT)**
  - Definition of SDT
  - Difference between Attribute Grammar and SDT
  - Yacc-style syntax
- **Symbol Table**
  - Lexically nested symbol tables, name mapping
  - Locate and store entries into symbol table
- **Type System**
  - Type expression, type inference, type equivalence
  - Type checker - static, dynamic
  - Type proof for an expression using inference rules
IR and Code Generation

- **Intermediate Representation**
  - Structural, linear or hybrid
  - AST, DAG, three-address code, CFG, SSA & etc

- **Code shape**
  - Simple code generation using SDT
  - Handling assignment & data layout
  - Boolean & relational implementation
  - Control flows: if-else, switch-case, for-loop
Code Optimizations

- Dependence Analysis
  - Dependence test: true, anti, output, input
  - Dependence distance
  - Parallelization v.s. vectorization
  - Safety of transformation

- Directed Acyclic Graphs (DAG)
  - Definition of DAG
  - Basic block DAG construction
Procedure Abstraction

- **Control Abstraction**
  - Dynamic activation tree & runtime stack
  - Activation record layout

- **Addressability**
  - Lexical scope
  - Locate and reference variables at correct lexical scopes
  - Different implementation approaches: access link or display

- **Standard Linkage**
  - Compile-time & run-time (what is known at compile-time)
  - Prolog, epilog, pre-call, post-return